<table>
<thead>
<tr>
<th>Project-Team</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIO Project-Team</td>
<td>105</td>
</tr>
<tr>
<td>TYPICAL Project-Team</td>
<td>108</td>
</tr>
<tr>
<td>VASY Project-Team</td>
<td>109</td>
</tr>
<tr>
<td>VEGAS Project-Team</td>
<td>112</td>
</tr>
<tr>
<td>VERIDIS Team</td>
<td>114</td>
</tr>
<tr>
<td>VERTECS Project-Team</td>
<td>116</td>
</tr>
<tr>
<td>ALEA Project-Team (section vide)</td>
<td>118</td>
</tr>
<tr>
<td>APICS Project-Team</td>
<td>119</td>
</tr>
<tr>
<td>ASPI Project-Team (section vide)</td>
<td>123</td>
</tr>
<tr>
<td>BACCHUS Team</td>
<td>124</td>
</tr>
<tr>
<td>BIPOP Project-Team</td>
<td>129</td>
</tr>
<tr>
<td>CAD Team</td>
<td>132</td>
</tr>
<tr>
<td>CAGIRE Team</td>
<td>133</td>
</tr>
<tr>
<td>CALVI Project-Team</td>
<td>134</td>
</tr>
<tr>
<td>CLASSIC Project-Team (section vide)</td>
<td>135</td>
</tr>
<tr>
<td>COMMANDS Project-Team</td>
<td>136</td>
</tr>
<tr>
<td>CONCHA Project-Team</td>
<td>137</td>
</tr>
<tr>
<td>CORIDA Project-Team</td>
<td>142</td>
</tr>
<tr>
<td>CQFD Project-Team</td>
<td>143</td>
</tr>
<tr>
<td>DEFI Project-Team</td>
<td>144</td>
</tr>
<tr>
<td>DISCO Team</td>
<td>147</td>
</tr>
<tr>
<td>DOLPHIN Project-Team</td>
<td>150</td>
</tr>
<tr>
<td>GAMMA3 Project-Team (section vide)</td>
<td>154</td>
</tr>
<tr>
<td>GECO Team (section vide)</td>
<td>155</td>
</tr>
<tr>
<td>GEOSTAT Project-Team</td>
<td>156</td>
</tr>
<tr>
<td>I4S Team</td>
<td>157</td>
</tr>
<tr>
<td>IPSO Project-Team (section vide)</td>
<td>158</td>
</tr>
<tr>
<td>MATHFI Project-Team</td>
<td>159</td>
</tr>
<tr>
<td>MAXPLUS Project-Team</td>
<td>164</td>
</tr>
<tr>
<td>MC2 Project-Team</td>
<td>166</td>
</tr>
<tr>
<td>MICMAC Project-Team (section vide)</td>
<td>168</td>
</tr>
<tr>
<td>MISTIS Project-Team</td>
<td>169</td>
</tr>
<tr>
<td>MODAL Team</td>
<td>172</td>
</tr>
<tr>
<td>NACHOS Project-Team</td>
<td>175</td>
</tr>
<tr>
<td>NANO-D Team</td>
<td>178</td>
</tr>
<tr>
<td>NECS Project-Team</td>
<td>179</td>
</tr>
<tr>
<td>NON-A Team (section vide)</td>
<td>181</td>
</tr>
<tr>
<td>OPALE Project-Team</td>
<td>182</td>
</tr>
<tr>
<td>POEMS Project-Team</td>
<td>185</td>
</tr>
<tr>
<td>PUMAS Team</td>
<td>187</td>
</tr>
</tbody>
</table>
81. REALOPT Project-Team ................................................................. 188
82. REGULARITY Team .................................................................... 189
83. SELECT Project-Team ............................................................... 190
84. SEQUEL Project-Team ............................................................... 191
85. SIERRA Project-Team ............................................................... 192
86. SIMPAF Project-Team ............................................................... 195
87. SMASH Project-Team (section vide) ........................................ 197
88. TAO Project-Team ................................................................. 198
89. TOSCA Project-Team ............................................................. 200
90. TROPICS Project-Team ........................................................... 201

COMPUTATIONAL SCIENCES FOR BIOLOGY, MEDICINE AND THE ENVIRONMENT
91. ABS Project-Team .................................................................... 203
92. AMIB Project-Team ............................................................... 205
93. ANUBIS Project-Team (section vide) ....................................... 207
94. ASCLEPIOS Project-Team ...................................................... 208
95. ATHENA Project-Team .......................................................... 209
96. BAMBOO Team ...................................................................... 210
97. BANG Project-Team ............................................................... 213
98. Beagle Team ........................................................................... 214
99. BIGS Project-Team ............................................................... 215
100. BIOCORE Project-Team ......................................................... 216
101. BONSAI Project-Team .......................................................... 217
102. CARMEN Team (section vide) .............................................. 220
103. CLIME Project-Team ........................................................... 221
104. CORTEX Project-Team .......................................................... 223
105. DEMAR Project-Team ............................................................ 226
106. DIGIPLANTE Team ............................................................... 228
107. DRACULA Project-Team .......................................................... 229
108. ESTIME Project-Team ............................................................ 230
109. FLUMINANCE Project-Team .................................................. 232
110. GALEN Team ...................................................................... 234
111. IBIS Project-Team ................................................................. 235
112. MACS Project-Team ............................................................... 236
113. MAGIQUE-3D Project-Team .................................................. 238
114. MAGNOME Project-Team ...................................................... 240
115. MASAIE Project-Team (section vide) ..................................... 242
116. MODEMIC Team ................................................................. 243
117. MOISE Project-Team ............................................................. 244
118. NEUROMATHCOMP Project-Team ........................................ 246
119. NUMED Project-Team .......................................................... 249
120. PARIETAL Project-Team .......................................................... 250
<table>
<thead>
<tr>
<th></th>
<th>Project-Team</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>121.</td>
<td>REO Project-Team</td>
<td>252</td>
</tr>
<tr>
<td>122.</td>
<td>S.H.A.M.A.N Team</td>
<td>253</td>
</tr>
<tr>
<td>123.</td>
<td>SAGE Project-Team</td>
<td>256</td>
</tr>
<tr>
<td>124.</td>
<td>SERPICO Team</td>
<td>260</td>
</tr>
<tr>
<td>125.</td>
<td>SISYPHE Project-Team</td>
<td>263</td>
</tr>
<tr>
<td>126.</td>
<td>STEEP Exploratory Action</td>
<td>265</td>
</tr>
<tr>
<td>127.</td>
<td>SYMBIOSE Project-Team</td>
<td>266</td>
</tr>
<tr>
<td>128.</td>
<td>VIRTUAL PLANTS Project-Team</td>
<td>269</td>
</tr>
<tr>
<td>129.</td>
<td>VISAGES Project-Team</td>
<td>273</td>
</tr>
<tr>
<td>130.</td>
<td>ACES Project-Team (section vide)</td>
<td>278</td>
</tr>
<tr>
<td>131.</td>
<td>ADAM Project-Team</td>
<td>279</td>
</tr>
<tr>
<td>132.</td>
<td>ALGORILLE Project-Team</td>
<td>282</td>
</tr>
<tr>
<td>133.</td>
<td>AMAZONES Team</td>
<td>285</td>
</tr>
<tr>
<td>134.</td>
<td>ARLES Project-Team</td>
<td>288</td>
</tr>
<tr>
<td>135.</td>
<td>ASAP Project-Team</td>
<td>293</td>
</tr>
<tr>
<td>136.</td>
<td>ASCOLA Project-Team</td>
<td>295</td>
</tr>
<tr>
<td>137.</td>
<td>ATLANMOD Team</td>
<td>298</td>
</tr>
<tr>
<td>138.</td>
<td>CEPAGE Project-Team</td>
<td>304</td>
</tr>
<tr>
<td>139.</td>
<td>CIDRE Project-Team</td>
<td>306</td>
</tr>
<tr>
<td>140.</td>
<td>DIONYSOS Project-Team</td>
<td>307</td>
</tr>
<tr>
<td>141.</td>
<td>DISTRIBUTCOM Project-Team</td>
<td>308</td>
</tr>
<tr>
<td>142.</td>
<td>DNET Team</td>
<td>309</td>
</tr>
<tr>
<td>143.</td>
<td>FOCUS Project-Team</td>
<td>310</td>
</tr>
<tr>
<td>144.</td>
<td>GANG Project-Team (section vide)</td>
<td>311</td>
</tr>
<tr>
<td>145.</td>
<td>GRAAL Project-Team</td>
<td>312</td>
</tr>
<tr>
<td>146.</td>
<td>GRAND-LARGE Project-Team</td>
<td>315</td>
</tr>
<tr>
<td>147.</td>
<td>HIEPACS Project-Team</td>
<td>320</td>
</tr>
<tr>
<td>148.</td>
<td>HIPERCOM Project-Team</td>
<td>324</td>
</tr>
<tr>
<td>149.</td>
<td>INDES Project-Team</td>
<td>325</td>
</tr>
<tr>
<td>150.</td>
<td>KERDATA Team</td>
<td>328</td>
</tr>
<tr>
<td>151.</td>
<td>MADYNES Project-Team</td>
<td>330</td>
</tr>
<tr>
<td>152.</td>
<td>MAESTRO Project-Team (section vide)</td>
<td>332</td>
</tr>
<tr>
<td>153.</td>
<td>MASCOTTE Project-Team</td>
<td>333</td>
</tr>
<tr>
<td>154.</td>
<td>MESCAL Project-Team</td>
<td>338</td>
</tr>
<tr>
<td>155.</td>
<td>MOAIS Project-Team</td>
<td>340</td>
</tr>
<tr>
<td>156.</td>
<td>MYRIADS Team</td>
<td>342</td>
</tr>
<tr>
<td>157.</td>
<td>OASIS Project-Team</td>
<td>346</td>
</tr>
<tr>
<td>158.</td>
<td>PHOENIX Project-Team</td>
<td>348</td>
</tr>
<tr>
<td>159.</td>
<td>PLANETE Project-Team</td>
<td>353</td>
</tr>
<tr>
<td>160.</td>
<td>POPS Project-Team</td>
<td>359</td>
</tr>
<tr>
<td>Project-Team</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>161. RAP Project-Team (section vide)</td>
<td>362</td>
<td></td>
</tr>
<tr>
<td>162. REGAL Project-Team</td>
<td>363</td>
<td></td>
</tr>
<tr>
<td>163. RESO Project-Team</td>
<td>365</td>
<td></td>
</tr>
<tr>
<td>164. RMOD Project-Team</td>
<td>366</td>
<td></td>
</tr>
<tr>
<td>165. RUNTIME Project-Team</td>
<td>368</td>
<td></td>
</tr>
<tr>
<td>166. SARDES Project-Team</td>
<td>373</td>
<td></td>
</tr>
<tr>
<td>167. SCORE Team</td>
<td>376</td>
<td></td>
</tr>
<tr>
<td>168. SWING Team</td>
<td>377</td>
<td></td>
</tr>
<tr>
<td>169. TREC Project-Team</td>
<td>378</td>
<td></td>
</tr>
<tr>
<td>170. TRISKELL Project-Team</td>
<td>379</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PERCEPTION, COGNITION, INTERACTION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>171. ALICE Project-Team</td>
<td>381</td>
<td></td>
</tr>
<tr>
<td>172. ALPAGE Project-Team</td>
<td>383</td>
<td></td>
</tr>
<tr>
<td>173. ARIANA Project-Team</td>
<td>388</td>
<td></td>
</tr>
<tr>
<td>174. AROBAS Project-Team</td>
<td>389</td>
<td></td>
</tr>
<tr>
<td>175. ARTIS Project-Team</td>
<td>391</td>
<td></td>
</tr>
<tr>
<td>176. AVIZ Project-Team</td>
<td>395</td>
<td></td>
</tr>
<tr>
<td>177. AXIS Project-Team</td>
<td>399</td>
<td></td>
</tr>
<tr>
<td>178. COPRIN Project-Team</td>
<td>403</td>
<td></td>
</tr>
<tr>
<td>179. DAHU Project-Team (section vide)</td>
<td>405</td>
<td></td>
</tr>
<tr>
<td>180. DREAM Project-Team</td>
<td>406</td>
<td></td>
</tr>
<tr>
<td>181. E-MOTION Project-Team</td>
<td>408</td>
<td></td>
</tr>
<tr>
<td>182. EDELWEISS Project-Team</td>
<td>413</td>
<td></td>
</tr>
<tr>
<td>183. EVASION Project-Team</td>
<td>417</td>
<td></td>
</tr>
<tr>
<td>184. EXMO Project-Team</td>
<td>420</td>
<td></td>
</tr>
<tr>
<td>185. FLOWERS Project-Team</td>
<td>422</td>
<td></td>
</tr>
<tr>
<td>186. GRAPHIK Project-Team</td>
<td>439</td>
<td></td>
</tr>
<tr>
<td>187. IMARA Project-Team</td>
<td>440</td>
<td></td>
</tr>
<tr>
<td>188. IMEDIA Project-Team</td>
<td>442</td>
<td></td>
</tr>
<tr>
<td>189. IN-SITU Project-Team</td>
<td>443</td>
<td></td>
</tr>
<tr>
<td>190. IPARLA Project-Team</td>
<td>452</td>
<td></td>
</tr>
<tr>
<td>191. LAGADIC Project-Team</td>
<td>454</td>
<td></td>
</tr>
<tr>
<td>192. LEAR Project-Team</td>
<td>458</td>
<td></td>
</tr>
<tr>
<td>193. LEO Team</td>
<td>459</td>
<td></td>
</tr>
<tr>
<td>194. MAGRIT Project-Team</td>
<td>462</td>
<td></td>
</tr>
<tr>
<td>195. MAIA Project-Team</td>
<td>463</td>
<td></td>
</tr>
<tr>
<td>196. METISS Project-Team</td>
<td>464</td>
<td></td>
</tr>
<tr>
<td>197. MIMETIC Team</td>
<td>466</td>
<td></td>
</tr>
<tr>
<td>198. MINT Team</td>
<td>468</td>
<td></td>
</tr>
<tr>
<td>199. Morpheo Team</td>
<td>470</td>
<td></td>
</tr>
<tr>
<td>200. MOSTRARE Project-Team</td>
<td>472</td>
<td></td>
</tr>
<tr>
<td>Project-Team</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>201. ORPAILLEUR</td>
<td>473</td>
<td></td>
</tr>
<tr>
<td>202. PAROLE</td>
<td>477</td>
<td></td>
</tr>
<tr>
<td>203. PERCEPTION</td>
<td>481</td>
<td></td>
</tr>
<tr>
<td>204. PRIMA</td>
<td>482</td>
<td></td>
</tr>
<tr>
<td>205. PULSAR</td>
<td>487</td>
<td></td>
</tr>
<tr>
<td>206. REVES</td>
<td>492</td>
<td></td>
</tr>
<tr>
<td>207. Sémagramme</td>
<td>494</td>
<td></td>
</tr>
<tr>
<td>208. SMIS</td>
<td>497</td>
<td></td>
</tr>
<tr>
<td>209. TALARIS</td>
<td>498</td>
<td></td>
</tr>
<tr>
<td>210. TEMICS</td>
<td>501</td>
<td></td>
</tr>
<tr>
<td>211. TEXMEX</td>
<td>504</td>
<td></td>
</tr>
<tr>
<td>212. VR4I Team</td>
<td>509</td>
<td></td>
</tr>
<tr>
<td>213. WAM</td>
<td>511</td>
<td></td>
</tr>
<tr>
<td>214. WILLOW</td>
<td>514</td>
<td></td>
</tr>
<tr>
<td>215. ZENITH</td>
<td>515</td>
<td></td>
</tr>
</tbody>
</table>
BYMOORE Exploratory Action

3. Software

3.1. Software

- **IODC**: Framework for implementing transparent iterative optimization in data centers, see Result 4.1.
- **P & R for neuromorphic accelerator**: A place and route software which maps a neural network graph on an analog neural network hardware, see Result 4.3.
- **HPT**: A performance comparison tool based on non-parametric tests, see Result 4.2.
- **Visual cortex model**: This model is initially based on Poggio’s HMAX model, and has been reimplemented with the prospect of progressively moving it into hardware as efficiently as possible. This work is loosely connected to Result 4.4, and it is still work in progress.
5. Software

5.1. The Apron Numerical Abstract Domain Library

Participants: Antoine Miné [correspondent], Bertrand Jeannet [team PopArt, INRIA-RA].

The APRON library is dedicated to the static analysis of the numerical variables of a program by abstract interpretation. Its goal is threefold: provide ready-to-use numerical abstractions under a common API for analysis implementers, encourage the research in numerical abstract domains by providing a platform for integration and comparison of domains, and provide a teaching and demonstration tool to disseminate knowledge on abstract interpretation.

The APRON library is not tied to a particular numerical abstraction but instead provides several domains with various precision versus cost trade-offs (including intervals, octagons, linear equalities and polyhedra). A specific C API was designed for domain developers to minimize the effort when incorporating a new abstract domain: only few domain-specific functions need to be implemented while the library provides various generic services and fallback methods (such as scalar and interval operations for most numerical data-types, parametric reduced products, and generic transfer functions for non-linear expressions). For the analysis designer, the APRON library exposes a higher-level API with C, C++, OCaml, and Java bindings. This API is domain-neutral and supports a rich set of semantic operations, including parallel assignments (useful to analyze automata), substitutions (useful for backward analysis), non-linear numerical expressions, and IEEE floating-point arithmetic.

The APRON library is freely available on the web at http://apron.cri.ensmp.fr/library; it is distributed under the LGPL license and is hosted at INRIAGForge. Packages exist for the Debian and Fedora Linux distributions. In order to help disseminate the knowledge on abstract interpretation, a simple inter-procedural static analyzer for a toy language is included. An on-line version is deployed at http://pop-art.inrialpes.fr/interproc/interprocweb.cgi.

The APRON library is developed since 2006 and currently consists of 130,000 lines of C, C++, OCaml, and Java.

Current and past external library users include the Constraint team (LINA, Nantes, France), the Proval/Démon team (LRI Orsay, France), the Analysis of Computer Systems Group (New-York University, USA), the Sierum software analysis platform (Kansas State University, USA), NEC Labs (Princeton, USA), EADS CCR (Paris, France), IRIT (Toulouse, France), ONERA (Toulouse, France), CEA LIST (Saclay, France), VERIMAG (Grenoble, France), ENSMP CRI (Fontainebleau, France), the IBM T.J. Watson Research Center (USA), the University of Edinburgh (UK).

5.2. The Astrée Static Analyzer of Synchronous Software

Participants: Patrick Cousot [project scientifique leader, correspondent], Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, Xavier Rival.

ASTRÉE is a static analyzer for sequential programs based on abstract interpretation [37], [28], [38], [30].

The ASTRÉE static analyzer [27], [43] aims at proving the absence of runtime errors in programs written in the C programming language.

ASTRÉE analyzes structured C programs, with complex memory usages, but without dynamic memory allocation nor recursion. This encompasses many embedded programs as found in earth transportation, nuclear energy, medical instrumentation, and aerospace applications, in particular synchronous control/command. The whole analysis process is entirely automatic.
**ASTRÉE** discovers all runtime errors including:

- undefined behaviors in the terms of the ANSI C99 norm of the C language (such as division by 0 or out of bounds array indexing);
- any violation of the implementation-specific behavior as defined in the relevant Application Binary Interface (such as the size of integers and arithmetic overflows);
- any potentially harmful or incorrect use of C violating optional user-defined programming guidelines (such as no modular arithmetic for integers, even though this might be the hardware choice);
- failure of user-defined assertions.

The analyzer performs an abstract interpretation of the programs being analyzed, using a parametric domain (**ASTRÉE** is able to choose the right instantiation of the domain for wide families of software). This analysis produces abstract invariants, which over-approximate the reachable states of the program, so that it is possible to derive an over-approximation of the dangerous states (defined as states where any runtime error mentioned above may occur) that the program may reach, and produces alarms for each such possible runtime error. Thus the analysis is sound (it correctly discovers all runtime errors), yet incomplete, that is it may report false alarms (i.e., alarms that correspond to no real program execution). However, the design of the analyzer ensures a high level of precision on domain-specific families of software, which means that the analyzer produces few or no false alarms on such programs.

In order to achieve this high level of precision, **ASTRÉE** uses a large number of expressive abstract domains, which allow expressing and inferring complex properties about the programs being analyzed, such as numerical properties (digital filters, floating-point computations), Boolean control properties, and properties based on the history of program executions.

**ASTRÉE** has achieved the following two unprecedented results:

- **A340–300.** In Nov. 2003, **ASTRÉE** was able to prove completely automatically the absence of any RTE in the primary flight control software of the Airbus A340 fly-by-wire system, a program of 132,000 lines of C analyzed in 1h20 on a 2.8 GHz 32-bit PC using 300 MB of memory (and 50mn on a 64-bit AMD Athlon 64 using 580 MB of memory).
- **A380.** From Jan. 2004 on, **ASTRÉE** was extended to analyze the electric flight control codes then in development and test for the A380 series. The operational application by Airbus France at the end of 2004 was just in time before the A380 maiden flight on Wednesday, 27 April, 2005.

These research and development successes have led to consider the inclusion of **ASTRÉE** in the production of the critical software for the A350. **ASTRÉE** is currently industrialized by AbsInt Angewandte Informatik GmbH and is commercially available.

### 5.3. The AstréeA Static Analyzer of Asynchronous Software

**Participants:** Patrick Cousot [project scientifique leader, correspondant], Radhia Cousot, Jérôme Feret, Antoine Miné, Xavier Rival.

**ASTRÉEA** is a static analyzer prototype for parallel software based on abstract interpretation [39], [40], [32]. It started with support from THÉSÉE ANR project (2006–2010) and is continuing within the ASTRÉEA project (2012–2015).

The **ASTRÉEA** prototype www.astreea.ens.fr is a fork of the **ASTRÉE** static analyzer (see 5.2) that adds support for analyzing parallel embedded C software.
**ASTRÉE** analyzes C programs composed of a fixed set of threads that communicate through a shared memory and synchronization primitives (mutexes, FIFOs, blackboards, etc.), but without recursion nor dynamic creation of memory, threads nor synchronization objects. **ASTRÉE** assumes a real-time scheduler, where thread scheduling strictly obeys the fixed priority of threads. Our model follows the ARINC 653 OS specification used in embedded industrial aeronautic software. Additionally, **ASTRÉE** employs a weakly-consistent memory semantics to model memory accesses not protected by a mutex, in order to take into account soundly hardware and compiler-level program transformations (such as optimizations). **ASTRÉE** checks for the same run-time errors as **ASTRÉE**', with the addition of data-races.

Compared to **ASTRÉE**, **ASTRÉE** features: a new iterator to compute thread interactions, a refined memory abstraction that takes into account the effect of interfering threads, and a new scheduler partitioning domain. This last domain allows discovering and exploiting mutual exclusion properties (enforced either explicitly through synchronization primitives, or implicitly by thread priorities) to achieve a precise analysis.

**ASTRÉE** is currently being applied to analyze a large industrial avionic software: 1.6 MLines of C and 15 threads, completed with a 2,500-line model of the ARINC 653 OS developed for the analysis. The analysis currently takes 29h on a 2.66 GHz 64-bit intel server using one core and generates around 1,800 alarms. The low computation time (only a few times larger than the analysis time by **ASTRÉE** of synchronous programs of a similar size and structure) shows the scalability of the approach (in particular, we avoid the usual combinatorial explosion associated to thread interleavings). Precision-wise, the result, while not as impressive as that of **ASTRÉE**', is quite encouraging. Improvements were made this year concerning the precision of **ASTRÉE** (from 7,600 alarms in 2010 to 1,800 now) and will continue within the scope of the **ASTRÉE** ANR project (Section 8.1.1.2).

The details of the analysis are described in [22].

### 5.4. OpenKappa

**Participants:** Monte Brown [Harvard Medical School], Vincent Danos [University of Edinburgh], Jérôme Feret [Correspondent], Walter Fontana [Harvard Medical School], Russ Harmer [Harvard Medical School], Jean Krivine [Paris VII].

**OpenKappa** is a collection of tools to build, debug and run models of biological pathways. It contains a compiler for the Kappa Language [49], a static analyzer [48] (for debugging models), a simulator [47], a compression tool for causal traces [46], and a model reduction tool [4, 45, 50].

**OpenKappa** is developed since 2007 and, the OCaml version currently consists of 46,000 lines of OCaml. Software are available in OCaml and in Java. Moreover, an Eclipse plugin is available. **OpenKappa** is freely available on the web at [http://kappalanguage.org](http://kappalanguage.org) under the LGPL license. Discussion groups are also available on line.

Current external users include the Ecole Polytechnique Federale de Lausanne, the UNAM-Genomics Mexico team. It is used as pedagogical material in graduate lessons at Harvard Medical School, and at the Interdisciplinary Approaches to Life science (AIV) Master Program (Université de Médecine Paris-Descartes).

### 5.5. Translation Validation

**Participant:** Xavier Rival [correspondent].

The main goal of this software project is to make it possible to certify automatically the compilation of large safety critical software, by proving that the compiled code is correct with respect to the source code: When the proof succeeds, this guarantees that no compiler bug did cause incorrect code be generated. Furthermore, this approach should allow to meet some domain specific software qualification criteria (such as those in DO-178 regulations for avionics software), since it allows proving that successive development levels are correct with respect to each other *i.e.*, that they implement the same specification. Last, this technique also justifies the use of source level static analyses, even when an assembly level certification would be required, since it establishes separately that the source and the compiled code are equivalent.
The compilation certification process is performed automatically, thanks to a prover designed specifically. The automatic proof is done at a level of abstraction which has been defined so that the result of the proof of equivalence is strong enough for the goals mentioned above and so that the proof obligations can be solved by efficient algorithms.

The current software features both a C to Power-PC compilation certifier and an interface for an alternate source language frontend, which can be provided by an end-user.

5.6. Zarith

Participants: Antoine Miné [Correspondent], Xavier Leroy [INRIA Paris-Rocquencourt], Pascal Cuoq [CEA LIST].

ZARITH is a small (10K lines) OCaml library that implements arithmetic and logical operations over arbitrary-precision integers. It is based on the GNU MP library to efficiently implement arithmetic over big integers. Special care has been taken to ensure the efficiency of the library also for small integers: small integers are represented as Caml unboxed integers and use a specific C code path. Moreover, optimized assembly versions of small integer operations are provided for a few common architectures.

ZARITH is an open-source project hosted at OCamlForge (http://forge.ocamlcore.org/projects/zarith) and distributed under a modified LGPL license.

ZARITH is currently used in the ASTRÉE analyzer to enable the sound analysis of programs featuring 64-bit (or larger) integers. It is also used in the Frama-C analyzer platform developed at CEA LIST and INRIA Saclay.
ALF Project-Team

5. Software

5.1. Panorama

The ALF team is developing several software prototypes for research purposes: compilers, architectural simulators, programming environments, ....

Among the many prototypes developed in the project, we describe here ATMI, a microarchitecture temperature model for processor simulation, STiMuL, a temperature model for steady state studies, ATC, an address trace compressor, HAVEGE, an unpredictable random number generator and tiptop, a user-level Linux utility that collects data from hardware performance counters for running tasks, software developed by the team.

5.2. ATMI

Participant: Pierre Michaud.

Contact: Pierre Michaud

Status: Registered with APP Number IDDN.FR.001.250021.000.S.P.2006.000.10600, Available under GNU General Public License

Research on temperature-aware computer architecture requires a chip temperature model. General purpose models based on classical numerical methods like finite differences or finite elements are not appropriate for such research, because they are generally too slow for modeling the time-varying thermal behavior of a processing chip.

We have developed an ad hoc temperature model, ATMI (Analytical model of Temperature in Microprocessors), for studying thermal behaviors over a time scale ranging from microseconds to several minutes. ATMI is based on an explicit solution to the heat equation and on the principle of superposition. ATMI can model any power density map that can be described as a superposition of rectangle sources, which is appropriate for modeling the microarchitectural units of a microprocessor.

Visit http://www.irisa.fr/alf/ATMI or contact Pierre Michaud.

5.3. STiMuL

Participant: Pierre Michaud.

Status: Registered with APP Number IDDN.FR.001.220013.000.S.P.2010.000.31235, Available under GNU General Public License

Some recent research has started investigating the microarchitectural implications of 3D circuits, for which the thermal constraint is stronger than for conventional 2D circuits.

STiMuL can be used to model steady-state temperature in 3D circuits consisting of several layers of different materials. STiMuL is based on a rigorous solution to the Laplace equation \[ 9 \]. The number and characteristics of layers can be defined by the user. The boundary conditions can also be defined by the user. In particular, STiMuL can be used along with thermal imaging to obtain the power density inside an integrated circuit. This power density could be used for instance in a dynamic simulation oriented temperature modeling such as ATMI.

STiMuL is written in C and uses the FFTW library for discrete Fourier transforms computations.

Visit http://www.irisa.fr/alf/stimul or contact Pierre Michaud.
5.4. ATC

**Participant:** Pierre Michaud.

**Contact:** Pierre Michaud

**Status:** registered with APP number IDDN.FR.001.160031.000.S.P.2009.000.10800, available under GNU LGPL License.

Trace-driven simulation is an important tool in the computer architect’s toolbox. However, one drawback of trace-driven simulation is the large amount of storage that may be necessary to store traces. Trace compression techniques are useful for decreasing the storage space requirement. But general-purpose compression techniques are generally not optimal for compressing traces because they do not take advantage of certain characteristics of traces. By specializing the compression method and taking advantages of known trace characteristics, it is possible to obtain a better tradeoff between the compression ratio, the memory consumption and the compression and decompression speed.

ATC is a utility and a C library for compressing/decompressing address traces. It implements a new lossless transformation, Bytesort, that exploits spatial locality in address traces. ATC leverages existing general-purpose compressors such as gzip and bzip2. ATC also provides a lossy compression mode that yields higher compression ratios while preserving certain important characteristics of the original trace.

Visit [http://www.irisa.fr/alf/atc](http://www.irisa.fr/alf/atc) or contact Pierre Michaud.

5.5. HAVEGE

**Participant:** André Seznec.

**Contact:** André Seznec

**Status:** Registered with APP Number IDDN.FR.001.500017.001.S.P.2001.000.10000. Available under the LGPL license.

An unpredictable random number generator is a practical approximation of a truly random number generator. Such unpredictable random number generators are needed for cryptography. HAVEGE (HArdware Volatile Entropy Gathering and Expansion) is a user-level software unpredictable random number generator for general-purpose computers that exploits the continuous modifications of the internal volatile hardware states in the processor as a source of uncertainty [16]. HAVEGE combines on-the-fly hardware volatile entropy gathering with pseudo-random number generation.

The internal state of HAVEGE includes thousands of internal volatile hardware states and is merely unmonitorable. HAVEGE can reach an unprecedented throughput for a software unpredictable random number generator: several hundreds of megabits per second on current workstations and PCs.

The throughput of HAVEGE favorably competes with usual pseudo-random number generators such as `rand()` or `random()`. While HAVEGE was initially designed for cryptology-like applications, this high throughput makes HAVEGE usable for all application domains demanding high performance and high quality random number generators, e.g., Monte Carlo simulations.

Visit [http://www.irisa.fr/alf/HAVEGE](http://www.irisa.fr/alf/HAVEGE) or contact André Seznec.

5.6. Tiptop

**Participant:** Erven Rohou.

**Status:** Registered with APP (Agence de Protection des Programmes). Available under GNU General Public License v2.
Tiptop is a new simple and flexible user-level tool that collects hardware counter data on Linux platforms (version 2.6.31+). The goal is to make the collection of performance and bottleneck data as simple as possible, including simple installation and usage. In particular, we stress the following points.

- Installation is only a matter of compiling the source code. No patching of the Linux kernel is needed, and no special-purpose module needs to be loaded.
- No privilege is required, any user can run tiptop — non-privileged users can only watch processes they own, ability to monitor anybody’s process opens the door to side-channel attacks.
- The usage is similar to top. There is no need for the source code of the applications of interest, making it possible to monitor proprietary applications or libraries. And since there is no probe to insert in the application, understanding of the structure and implementation of complex algorithms and code bases is not required.
- Applications do not need to be restarted, and monitoring can start at any time (obviously, only events that occur after the start of tiptop are observed).
- Events can be counted per thread, or per process.

Tiptop is written in C. It can take advantage of libncurses when available for pseudo-graphic display. For more information, please contact Erven Rohou.
4. Software

4.1. Algolib

The Algolib library is a set of Maple routines that have been developed in the project for more than 15 years. Several parts of it have been incorporated into the standard library of Maple, but the most up-to-date version is always available for free from our web pages http://algo.inria.fr/libraries/. This library provides: tools for combinatorial structures (the combstruct package), including enumeration, random or exhaustive generation, generating functions for a large class of attribute grammars; tools for linear difference and differential equations (the gfun package), which have received a very positive review in Computing Reviews and have been incorporated in N. Sloane's superseeker at Bell Labs; tools for systems of multivariate linear operators, definite sums and integrals (the Mgfun package), including Gröbner bases in Ore algebras, that also treat commutative polynomials and have been the standard way to solve polynomial systems in Maple for a long period (although the user would not notice it); Mgfun has also been chosen at Risc (Linz) as the basis for their package Desing, tools for expansions in general asymptotic scales, which make it possible to handle in a transparent and automatic way the problems of finding the proper scale for an expansion and of dealing with the indefinite cancellation problem (the MultiSeries package).

4.2. Mathematics on the Web

We also provide access to our work to scientists who are not using Maple or any other computer algebra system in the form of automatically generated encyclopedias available on the web. The Encyclopedia of Combinatorial Structures at http://algo.inria.fr/ecs/ thus contains more than 1000 combinatorial structures for which generating functions, enumeration sequences, recurrences, and asymptotic approximations have been computed automatically. It gets more than 16,000 hits per month. The Dynamic Dictionary of Mathematical Functions (DDMF) at http://ddmf.msr-inria.inria.fr/ gathers several dozens of special functions for which identities, guaranteed high-precision numerical evaluations, power-series and asymptotic expansions, graphs, ...are generated automatically and on the user’s request, starting from a linear differential equation and its initial conditions. The underlying symbolic algorithms and implementations are those of gfun and Mgfun. All the production process being automated, the difficult and expensive step of checking each formula individually is suppressed. A nice specificity of this encyclopedia is its interactivity: the approximations values (numbers, series) are not bound to a statically set precision, rather, the user can fill in the precision he wants in a form, before clicking to ask for a refined identity to be generated, then displayed. This interactivity is based on a tool DynaMoW at http://ddmf.msr-inria.inria.fr/DynaMoW/ (for Dynamic Mathematics on the Web) that we develop as well. This is an Ocaml library that simultaneously controls external symbolic calculations and webpage generation at the same time. Being available on the web, the DDMF also plays the role of a showcase for part of the packages developed in our project. It is a successor of our former Encyclopedia of Special Functions at http://algo.inria.fr/esf/.
5. Software

5.1. TimeSquare

Participants: Charles André, Nicolas Chleq, Julien Deantoni, Benoît Ferrero, Frédéric Mallet [correspondant].

TimeSquare is a software environment for the modeling and analysis of timing constraints in embedded systems. It relies specifically on the Time Model of the MARTE UML profile (see section 3.2), and more accurately on the associated Clock Constraint Specification Language (CCSL) for the expression of timing constraints.

TimeSquare offers four main functionalities:

1. graphical and/or textual interactive specification of logical clocks and relative constraints between them;
2. definition and handling of user-defined clock constraint libraries;
3. automated simulation of concurrent behavior traces respecting such constraints, using a Boolean solver for consistent trace extraction;
4. call-back mechanisms for the traceability of results (animation of models, display and interaction with waveform representations, generation of sequence diagrams...).

In practice TimeSquare is a plug-in developed with Eclipse modeling tools. The software is registered by the Agence pour la Protection des Programmes, under number IDDN.FR.001.170007.000.S.P.2009.001.10600. It can be downloaded from the site http://timesquare.inria.fr/. It has been integrated in the OpenEmbeDD ANR RNTL platform, and other such actions are under way.

5.2. K-Passa

Participants: Anthony Coadou, Jean-Vivien Millo [correspondant], Robert de Simone.

This software is dedicated to the simulation, analysis, and static scheduling scheduling of Event/Marked Graphs, SDF and KRG extensions. A graphical interface allows to edit the Process Networks and their time annotations (latency, ...). Symbolic simulation and graph-theoretic analysis methods allow to compute and optimize static schedules, with best throughputs and minimal buffer sizes. In the case of KRG the (ultimately k-periodic) routing patterns can also be provided and transformed for optimal combination of switching and scheduling when channels are shared. KPASSA also allows for import/export of specific description formats such as UML-MARTE, to and from our other TimeSquare tool.

The tool was originally developed mainly as support for experimentations following our research results on the topic of Latency-Insensitive Design. This research was conducted and funded in part in the context of the CIM PACA initiative, with initial support from ST Microelectronics and Texas Instruments.

KPASSA is registered by the Agence pour la Protection des Programmes, under the number IDDN.FR.001.310003.000.S.P.2009.000.20700. it can be downloaded from the site http://www-sop.inria.fr/aoste/index.php?page=software/kpassa.

5.3. SynDEx

Participants: Maxence Guesdon, Yves Sorel [correspondant], Cécile Stentzel, Meriem Zidouni.

SynDEx is a system level CAD software implementing the AAA methodology for rapid prototyping and for optimizing distributed real-time embedded applications. Developed in OCaML it can be downloaded free of charge, under INRIA copyright, from the general SynDEx site http://www.syndex.org.
The AAA methodology is described in section 3.3. Accordingly, SYNDEx explores the space of possible allocations (spatial distribution and temporal scheduling), from application elements to architecture resources and services, in order to match real-time requirements; it does so by using schedulability analyses and heuristic techniques. Ultimately it generates automatically distributed real-time code running on real embedded platforms. The last major release of SYNDEx (V7) allows the specification of multi-periodic applications.

Application algorithms can be edited graphically as directed acyclic task graphs (DAG) where each edge represent a data dependence between tasks, or they may be obtained by translations from several formalisms such as Scicos (http://www.scicos.org), Signal/Polychrony (http://www.irisa.fr/espresso/Polychrony), or UML2/MARTE models (http://www.omg.org/technology/documents/profile_catalog.htm). Architectures are represented as graphical block diagrams composed of programmable (processors) and non-programmable (ASIC, FPGA) computing components, interconnected by communication media (shared memories, links and busses for message passing). In order to deal with heterogeneous architectures it may feature several components of the same kind but with different characteristics.

Two types of non-functional properties can be specified for each task of the algorithm graph. First, a period that does not depend on the hardware architecture. Second, real-time features that depend on the different types of hardware components, ranging amongst execution and data transfer time, memory, etc. Requirements are generally constraints on deadline equal to period, latency between any pair of tasks in the algorithm graph, dependence between tasks, etc.

Exploration of alternative allocations of the algorithm onto the architecture may be performed manually and/or automatically. The latter is achieved by performing real-time multiprocessor schedulability analyses and optimization heuristics based on the minimization of temporal or resource criteria. For example while satisfying deadlines and latencies constraints they can minimize the total execution time (makespan) of the application onto the given architecture, as well as the amount of memory. The results of each exploration is visualized as timing diagrams simulating the distributed real-time implementation.

Finally, real-time distributed embedded code can be automatically generated for dedicated distributed real-time executives, possibly calling services of resident real-time operating systems such as Linux/RTAI or Osek for instance. These executives are deadlock-free, based on off-line scheduling policies. Dedicated executives induce minimal overhead, and are built from processor-dependent executive kernels. To this date, executives kernels are provided for: TMS320C40, PIC18F2680, i80386, MC68332, MPC555, i80C196 and Unix/Linux workstations. Executive kernels for other processors can be achieved at reasonable cost following these examples as patterns.

5.4. SAS

Participants: Daniel de Rauglaudre [correspondant], Yves Sorel.

The SAS (Simulation and Analysis of Scheduling) software allows the user to perform the schedulability analysis of periodic task systems in the monoprocessor case.

The main contribution of SAS, when compared to other commercial and academic softwares of the same kind, is that it takes into account the exact preemption cost between tasks during the schedulability analysis. Beside usual real-time constraints (precedence, strict periodicity, latency, etc.) and fixed-priority scheduling policies (Rate Monotonic, Deadline Monotonic, Audsley++, User priorities), SAS additionally allows to select dynamic scheduling policy algorithms such as Earliest Deadline First (EDF). The resulting schedule is displayed as a typical Gantt chart with a transient and a permanent phase, or as a disk shape called "dameid", which clearly highlights the idle slots of the processor in the permanent phase.

For a schedulable task system under EDF, when the exact preemption cost is considered, the period of the permanent phase may be much longer than the least common multiple (LCM) of the periods of all tasks, as often found in traditional scheduling theory. Specific effort has been made to improve display in this case. The classical utilization factor, the permanent exact utilization factor, the preemption cost in the permanent phase, and the worst response time for each task are all displayed when the system is schedulable. Response times of each task relative time can also be displayed (separately).
SAS is written in OCaML, using CAMLP5 (syntactic preprocessor) and OLIBRT (a graphic toolkit under X). Both are written by Daniel de Rauglaudre.
5. Software

5.1. Introduction

Arénaire proposes various software and hardware realizations that are accessible from the web page http://www.ens-lyon.fr/LIP/Arenaire/Ware/. We describe below only those which progressed in 2011.

5.2. FloPoCo

Participants: Florent Dinechin [correspondant], Bogdan Pasca, Laurent-Stéphane Didier.

The purpose of the FloPoCo project is to explore the many ways in which the flexibility of the FPGA target can be exploited in the arithmetic realm. FloPoCo is a generator of operators written in C++ and outputting synthesizable VHDL automatically pipelined to an arbitrary frequency.

In 2011, FloPoCo was turned into a library which can be used as a back-end to high-level synthesis tools. An expression parser that generates a complete pipeline was also added for this context. The integer multiplier and floating-point adder were rewritten, and several new operators were added, including a floating-point power operator, and novel operators for integer and floating-point division by a constant.

Versions 2.2.0, 2.2.1, and 2.3.0 were released in 2011.

5.3. GNU MPFR

Participants: Vincent Lefèvre [correspondant], Paul Zimmermann.

GNU MPFR is an efficient multiple-precision floating-point library with well-defined semantics (copying the good ideas from the IEEE-754 standard), in particular correct rounding in 5 rounding modes. GNU MPFR provides about 80 mathematical functions, in addition to utility functions (assignments, conversions...). Special data (Not a Number, infinities, signed zeros) are handled like in the IEEE-754 standard.

MPFR was one of the main pieces of software developed by the old SPACES team at Loria. Since late 2006, with the departure of Vincent Lefèvre to Lyon, it has become a joint project between the Caramel (formerly SPACES then CACAO) and the Arénaire project-teams. MPFR has been a GNU package since 26 January 2009. GNU MPFR 3.0.1 was released on 4 April 2011 and GNU MPFR 3.1.0 was released on 3 October 2011.

The main improvements are the generic tests in a reduced exponent range, the possibility to include the mpfr.h header file several times while still supporting optional functions, and, for the developers, the choice of the native type for the exponent (and various corrections related to these features).

5.4. Exhaustive Tests for the Correct Rounding of Mathematical Functions

Participant: Vincent Lefèvre.

The search for the worst cases for the correct rounding (hardest-to-round cases) of mathematical functions (exp, log, sin, cos, etc.) in a fixed precision (mainly double precision) using Lefèvre’s algorithm is implemented by a set of utilities written in Perl, with calls to Maple/intpakX for computations on intervals and with C code generation for fast computations. It also includes a client-server system for the distribution of intervals to be tested and for tracking the status of intervals (fully tested, being tested, aborted).
These programs are run on the LIP network via Grid Engine (SGE). In June 2011, the SGE configuration was changed by the system administrator so that SIGSTOP/SIGCONT signals are sent to the jobs, allowing several users to use SGE at the same time. These signals make Maple crash (segmentation fault), and the Perl scripts needed to be improved to handle these crashes gracefully (by restarting the computations when need be, etc.). This SGE change made other problems appear, such as when the client is first stopped by SGE and is then killed by SGE (without being woke up by SGE), it cannot do its usual clean-up; workarounds were tried, but without success.

The above problems also made an inconsistency in the client-server protocol appear. The validity of the results was not affected, but the protocol had to be redesigned.

5.5. CGPE: Code Generation for Polynomial Evaluation

Participants: Christophe Mouilleron, Claude-Pierre Jeannerod.

The CGPE project, developed with Guillaume Revy (DALI research team, Université de Perpignan and LIRMM laboratory), aims at generating C codes for fast and certified polynomial evaluation, given various accuracy and architectural constraints. Several improvements for this tool, based on the addition of constraints in the first step of the generation process, were proposed in the PhD thesis of Ch. Mouilleron [ 12 ]. These improvements have been implemented, thus allowing us to reduce the whole generation time by about 50% on average.

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software).
- Recommended library or software: MPFI or Gappa.
- License: CeCILL
- Type of human computer interaction: command-line interface
- OS/Middleware: Unix
- Required library or software: Xerces-C++ XML Parser library and MPFR
- Programming Language: C++
- Status: beta
- Documentation: available in html format on URL: http://cgpe.gforge.inria.fr/

5.6. FLIP: Floating-point Library for Integer Processors

Participants: Claude-Pierre Jeannerod, Jingyan Jourdan-Lu.

FLIP is a C library for the efficient software support of binary32 IEEE 754-2008 floating-point arithmetic on processors without floating-point hardware units, such as VLIW or DSP processors for embedded applications. The current target architecture is the VLIW ST200 family from STMicroelectronics (especially the ST231 cores). This year, we have mostly worked on improving the design and implementation of the following operators with correct rounding “to nearest even”: DP2 (fused dot product in dimension two) and sum of two squares. The impact of the DP2 operator has been evaluated on the UTDSP benchmark, and on some kernels speed-ups of 1.46 have been observed. On the other hand, specializing DP2 to a sum of squares brings a speed-up of 2.

URL: http://flip.gforge.inria.fr/

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software)
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- APP: IDDN.FR.001.230018.S.A.2010.000.10000
- License: CeCILL v2
- Type of human computer interaction: C library callable, from any C program.
- OS/Middleware: any, as long as a C compiler is available.
- Required library or software: none.
- Programming language: C
5.7. SIPE: Small Integer Plus Exponent

Participant: Vincent Lefèvre.

SIPE (Small Integer Plus Exponent) is a C header file providing a fast floating-point arithmetic with correct rounding to the nearest in very small precision. Implemented operations are the addition, subtraction, multiplication, FMA, and minimum/maximum/comparison functions (of the signed numbers or in magnitude). SIPE has been written for exhaustive tests of simple algorithms in small precision in order to prove results or find conjectures (which could then be proved). In 2011, a research report was written about SIPE [62], including documentation and proof of the implementation; some bugs were fixed at the same time.

- ACM: D.2.2 (Software libraries), G.4 (Mathematical software).
- AMS: 26-04 Real Numbers, Explicit machine computation and programs.
- License: LGPL version 2.1 or later.
- Type of human computer interaction: C header file.
- OS/Middleware: any OS.
- Required library or software: GCC compiler.
- Programming language: C.
- Documentation: Research report RR-7832. [62]
- URL: http://www.vinc17.net/software/sipe.h
ATEAMS Project-Team

5. Software

5.1. AmbiDexter

Participants: Bas Basten [correspondent], Jurgen Vinju.

Characterization: A-3-up4, SO-4, SM-2-up3, EM-2-up3, SDL-4-up5, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: http://homepages.cwi.nl/~basten/ambiguity/

Objective: Statically detect ambiguity of context-free grammars for programming languages, as fast and precise as possible.

Users: Authors of context-free grammars of programming languages in SDF2, Rascal, ANTLR, etc.

Impact: This is the first usable ambiguity detection tool, aiming to solve the Achilles’ heel of context-free general parsing.

Competition: AmbiDexter is the fastest and most accurate tool currently available.

Engineering: AmbiDexter was developed by one person and will be maintained by another. It is 25 LOC in Java and distributed as a component of the Rascal IDE.

Publications: [14], [9], [2], [1]

5.2. Derric

Participants: Tijs van der Storm, Jeroen van den Bos [correspondent].

Characterization: A-2-up3, SO-4, SM-2-up3, EM-3, SDL-3-up4, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: http://svn.rascal-mpl.org/derric/

Objective: Encapsulate all the variability in the construction of so-called “carving” algorithms, then generate the fastest and most accurate implementations. Carving algorithms recover information that has been deleted or otherwise scrambled on digital media such as hard-disks, usb sticks and mobile phones.

Users: Digital forensic investigation specialists

Impact: Derric has the potential of revolutionizing the carving area. It does in 1500 lines of code what other systems need tens of thousands of lines for with the same accuracy. Derric will be an enabler for faster, more specialized and more successful location of important evidence material.

Competition: Derric competes in a small market of specialized open-source and commercial carving tools.

Engineering: Derric is a Rascal program of 1.5 kloc designed by two persons.

Publications: [27], [13]

5.3. Pacioli

Participants: Tijs van der Storm, Paul Griffioen [correspondent].

Characterization: A-2-up3, SO-4, SM-2, EM-3, SDL-3-up4, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: http://svn.rascal-mpl.org/pacioli/

Objective: Encapsulate all the variability in the construction of modeling and analysis tools in computational auditing

Users: Financial auditing experts

Impact: Pacioli is an experiment with a big potential in the field of computational auditing. It operates as a vehicle now for experimenting with new ideas in this field. The goal is to tackle the enormous complexity in the (trading) of companies using high level modeling and analysis techniques.

Competition: Pacioli competes with less specialized and less formal business analysis tooling, mostly based on spreadsheets.

Engineering: Pacioli is a part Java, part Rascal project written by one person.
5.4. Rascal

Participants: Paul Klint, Jurgen Vinju [correspondent], Tijs van der Storm, Bas Basten, Jeroen van den Bos, Mark Hills, Bert Lisser, Arnold Lankamp, Atze van der Ploeg, Vadim Zaytsev, Anastasia Izmaylova, Anya Helene Bagge.

Characterization: A5, SO-4, SM-4, EM-4, SDL-4-up5, OC-DA-3-CD-3-MS-3-TPM-3.

WWW: http://www.rascal-mpl.org

Objective: Provide a completely integrated programming language parametric meta programming language for the construction of any kind of meta program for any kind of programming language: analysis, transformation, generation, visualization.

Users: Researchers in model driven engineering, programming languages, software engineering, software analysis, as well as practitioners that need specialized tools.

Impact: Rascal is making the mechanics of meta programming into a non-issue. We can now focus on the interesting details of the particular fact extraction, model, source analysis, domain analysis as opposed to being distracted by the engineering details. Simple things are easy in Rascal and complex things are manageable, due to the integration, the general type system and high-level programming features.

Competition: There is a plethora of meta programming toolboxes and frameworks available, ranging from plain parser generators to fully integrated environments. Rascal is distinguished because it is a programming language rather than a specification formalism and because it completely integrates different technical domains (syntax definition, term rewriting, relational calculus). For simple tools, Rascal competes with scripting languages and for complex tools it competes context-free general parser generators, with query engines based on relational calculus and with term rewriting and strategic programming languages.

Engineering: Rascal is about 100 kLOC of Java code, designed by a core team of three and with a team of around 8 phd students and post-docs contributing to its design, implementation and maintenance. The goal is to work towards more bootstrapping and less Java code as the project continues.

Publications: [21], [28], [29], [22][6], [7]

5.4.1. Novelties

- Re-design of embedded grammar formalism including semantic disambiguation facilities.
- Extremely fast top-down context-free general parsing algorithm in cubic time and space.
- Parse error reporting via partial parse trees (useful in incremental syntax highlighting and incremental type analysis).
- Auto-indent feature for code generation templates.
- Significant extensions and improvements of software visualization library, such as hierarchical graphs and smaller set of more powerful primitives for charts and interactive features.
- Significant improvements to online documentation and interactive tutor environment.
- “ToLaTex” mode to include Rascal code in papers.
- ShellExec library for interacting via pipes with external programs.
- Bridge to Maude and K.
- Generalized function dispatch to arbitrary pattern dispatch.
- New module composition mechanism “extend” next to “import”.
- Ambiguity diagnostics library and parse tree visualizations as a first step towards more grammarware in the IDE.
- A command-line interface to run a single Rascal program.
- Fixed a number of memory leaks in the IDE.
- IDE features for mixed Java/Rascal projects.
- Rational numbers.
- Formal concept analysis library.
- Enhanced SDF2 to Rascal translation.
- Redesigned and simplified abstract grammar format.
- Added “break”, “continue” and “fail” statements for back-tracking and continuation control.
- Radically changed internal design from Visitor to Interpreter design pattern (using an automated refactoring).

5.5. IDE Meta-tooling Platform

**Participants:** Jurgen Vinju [correspondent], Arnold Lankamp, Anya Helene Bagge.

IMP, the IDE meta tooling platform is an Eclipse plugin developed mainly by the team of Robert M. Fuhrer at IBM TJ Watson Research institute. It is both an abstract layer for Eclipse, allowing rapid development of Eclipse based IDEs for programming languages, and a collection of meta programming tools for generating source code analysis and transformation tools.

**Characterization:** A5, SO-3, SM4-up5, EM-4, SDL-5, DA-2-CD-2-MS-2-TPM-2

**WWW:** [http://www.eclipse.org/imp](http://www.eclipse.org/imp)

**Objective:** The IDE Meta Tooling Platform (IMP) provides a high-level abstraction over the Eclipse API such that programmers can extend Eclipse with new programming languages or domain specific languages in a few simple steps. IMP also provides a number of standard meta tools such as a parser generator and a domain specific language for formal specifications of configuration parameters.

**Users:** Designers and implementers of IDEs for programming languages and domain specific languages. Also, designers and implementers of meta programming tools.

**Impact:** IMP is popular among meta programmers especially for it provides the right level of abstraction.

**Competition:** IMP competes with other Eclipse plugins for meta programming (such as Model Driven Engineering tools), but its API is more general and more flexible. IMP is a programmers framework rather than a set of generators.

**Engineering:** IMP is a long-lived project of many contributors, which is managed as an Eclipse incubation project at [eclipse.org](http://eclipse.org).

**Publications:** [ 3 ]

Jurgen Vinju and Arnold Lankamp contribute significantly to the development of IMP. Their effort is focused on the maintenance and optimization of a general purpose symbolic representation library for source code artifacts, called “PDB”. PDB stands for Program DataBase. For more information, please visit [http://www.eclipse.org/imp](http://www.eclipse.org/imp).

The Rascal language itself was accepted by Eclipse as a contribution to the IMP project. This will further strengthen the collaboration between the IMP and the Rascal team as well as generate a wider audience for Rascal.
5.6. Ensō

Participant: Tijs van der Storm [correspondent].

Characterization: A5, SO-4, SM-3-up-4, EM-2-up-4, SDL-4, OC-DA-4-CD-4-MS-4-TPM-4

WWW: http://www.enso-lang.org

Objective: Together with Prof. Dr. William R. Cook of the University of Texas at Austin, Tijs van der Storm has been designing and implementing a new programming system, called Ensō. Ensō is theoretically sound and practical reformulation of model-based development. It is based on model-interpretation as opposed to model transformation and code generation. Currently, the system already supports models for schemas (data models), web applications, context-free grammars, diagram editors and security.

Users: All programmers.

Impact: Ensō has the potential to revolutionize the activity of programming. By looking at model driven engineering from a completely fresh perspective, with as key ingredients interpreters and partial evaluation, it may make higher level (domain level) program construction and maintenance as effective as normal programming.

Competition: Ensō competes as a programming paradigm with model driven engineering tools and generic programming and languages that provide syntax macros and language extensions.

Engineering: Ensō is less than 7000 lines of (bootstrapped) Ruby code.

5.7. Software Language Processing Suite

Participant: Vadim Zaytsev [correspondent].

Characterization: A3-up4, SO-4, SM-3, EM-2up3, SDL-2, OC-DA-4-CD-4-MS-4-TPM-4

WWW: http://slps.sourceforge.net

Objective: The project facilitates exposition and comparison of approaches and techniques on language processing.

Users: Computer science students, teachers, engineers and practitioners

Impact: SLPS contains the largest collection of grammars for programming languages directly recovered from documentation, as well as the largest collection of source-to-source grammar formalisms translators and other related grammarware.

Engineering: SLPS is a large collection of scripts and programs written by Ralf Lämmel and Vadim Zaytsev.

5.7.1. Novelties

- New grammars: Ada, Dart, Eiffel, Fortran, Modula, Mediawiki, ...(now a total of 41)
- Grammar Tank: a new collection of 54 small grammars for research purposes
- TestMatch: a tool for grammar-based differential testing of ANTLR grammars and for nonterminal matching based of parsing generated test data (in collaboration with Ralf Lämmel).
- Grammar Hunter: a tool for automated notation-parametric grammar recovery (will also be a Rascal library).

5.8. Demo Light for Composing Models

Participants: Jan van Eijck [correspondent], Floor Sietsma.

Characterization: A2,SO-3,SM-1,EM-2,SDL-2,OC-4

WWW: http://homepages.cwi.nl/~jve/software/demolight0/

Objective: Demonstrate epistemic modeling and reasoning

Users: Students and researchers in application of epistemic logic

Impact: Demo light makes the theory of epistemic reasoning insightful by offering a Haskell library for experimenting with it.

Engineering: Demo Light is a Haskell library.
CAIRN Project-Team

5. Software

5.1. Panorama

Besides the development of new reconfigurable architectures, the need for efficient compilation flow is stronger than ever. Challenges come from the high parallelism of these architectures and also from new constraints such as resource heterogeneity, memory hierarchy and power constraints and management. We aim at defining a highly effective software framework for the compilation of high-level specifications into optimized code executed on a reconfigurable hardware platform. Figure 2 shows the global framework that we are currently developing.

Our approach assumes that the application is specified as a hierarchical block diagram of communicating tasks expressing data-flow or control, where each task is expressed using languages like C, Signal, Scilab or Matlab, and is then transformed into an internal representation by the compiler front-end. Then, our framework is based on applying some high-level transformations onto the internal representation. Different internal representations are used depending on the targeted transformations or the targeted architectures.

- The classical Control and Data Flow Graph (CDFG) is the main internal formalism of our framework. It is the basis for transformations like code optimizations, fixed-point transformations, instruction-set extraction or scheduling. Gateways will be provided from CDFG to other supported formalisms.
- The Hierarchical Conditional Dependency Graph (HCDG) format is also used as the internal representation for pattern-based transformations.
- Other internal representations like Signal Flow Graphs (SFG) and Polyhedral Reduced Dependence Graph (PRDG) will be used respectively for application accuracy estimation and loop parallelization techniques.

Finally, back-end tools enable the generation of code like VHDL for the hardwired or reconfigurable blocks, C for embedded processor software, and SystemC for simulation purposes (e.g. fixed-point simulations). The compiler front-end, the back-end generators, the transformation toolbox as well as the different internal representations and their respective gateways are based on a single framework: the Gecos framework.

Besides CAIRN’s general design workflow, and in order to promote research undertaken by CAIRN, several hardware and software prototypes are developed. Among those, some distributed software are presented in this report: Gecos a flexible compilation platform, ID.Fix an infrastructure for the automatic transformation of software code aiming at the conversion of floating-point data types into a fixed-point representation, UPaK and Durase for the compilation and the synthesis targeting reconfigurable platforms, and Interconnect Explorer a high-level power and delay estimation tool for on-chip interconnects.

5.2. Gecos

Participants: Steven Derrien [correspondant], Daniel Menard, Kevin Martin, Maxime Naullet, Antoine Floch, Antoine Morvan, Clément Guy, Amit Kumar.

The Gecos (Generic Compiler Suite) project is an open source Eclipse-based C compiler infrastructure developed in the CAIRN group since 2004 that allows for fast prototyping of complex compiler passes. Gecos was designed so as to address part of the shortcomings of existing C/C++ infrastructures such as SUIF and LLVM.

Gecos is a 100% Java based implementation and is based on modern software engineering practices. It uses Eclipse plugin as an underlying infrastructure and thus takes benefits of its plugin mechanism to be easily extensible. Gecos follows Model Driven Software Engineering techniques and rely on Eclipse Modeling Framework. The framework is open-source and is hosted on the INRIA gforge at http://gecos.gforge.inria.fr.

The Gecos infrastructure is still under very active development, and now serves as a backbone infrastructure to many group members (Upak, Durase, ID.Fix). In 2011, the work has focused on extending the loop analysis transformation framework, which now includes an OpenMP static analysis tool (developed jointly with Colorado State University) that was presented in June at the 7th International Workshop on OpenMP [39]. The software engineering challenges posed by optimizing compiler also happen to be a novel and promising application field for the MDE community, which led to joint publication [45] with members from CSU and the Triskell EPI team at the IEEE/ACM Models conference in October 2011. This cross fertilization between MDE and Compilers is the core topic of Clément Guy’s PhD thesis supervised by members of CAIRN (S. Derrien) and Triskell (J.M. Jezequel and B. Combemale).

5.3. ID.Fix: Infrastructure for the Design of Fixed-point Systems

Participants: Daniel Menard [correspondant], Olivier Senteiys, Romuald Rocher, Nicolas Simon, Quentin Meunier.

The different techniques proposed by the team for fixed-point conversion are implemented on the ID.Fix infrastructure. The application is described with a C code using floating-point data types and different pragmas, used to specify parameters (dynamic, input/output word-length, delay operations) for the fixed-point conversion. This tool determines and optimizes the fixed-point specification and then, generates a C code using fixed-point data types (ac_fixed) from Mentor Graphics. The infrastructure is made-up of three main modules corresponding to the fixed-point conversion (Fix.Conv), the accuracy evaluation (Acc.Eval) and the dynamic range evaluation (Dyn.Eval).
The different developments carried-out in 2011 allow obtaining a fixed-point conversion tool handling functions, conditional structures and repetitive structures having a fixed number of iterations during time. For the accuracy evaluation (Acc.Eval), conditional structures and correlation between noise sources have been considered. For the dynamic range evaluation (Dyn.Eval), the method based on the Karhunen-Loève Expansion (KLE) have been implemented. It allows determining the dynamic range for a given overflow probability.

The development of this tool has been achieved thanks to an INRIA post-doc in the context of S2S4HLS project until August 2011, and a University of Rennes graduate engineer from November 2011 in the context of DEFIS ANR project and different students during their training period.

5.4. UPaK: Abstract Unified Pattern-Based Synthesis Kernel for Hardware and Software Systems

**Participants:** Christophe Wolinski [correspondant], François Charot, Antoine Floch.

We are developing (with strong collaboration of Lund University, Sweden and Queensland University, Australia) UPaK Abstract Unified Pattern Based Synthesis Kernel for Hardware and Software Systems [119]. The preliminary experimental results obtained by the UPak system show that the methods employed in the systems enable a high coverage of application graphs with small quantities of patterns. Moreover, high application execution speed-ups are ensured, both for sequential and parallel application execution with processor extensions implementing the selected patterns. UPaK is one of the basis for our research on compilation and synthesis for reconfigurable platforms. It is based on the HCDG representation of the Polychrony software designed at INRIA-Rennes in the project-team Espresso.

5.5. DURASE: Automatic Synthesis of Application-Specific Processor Extensions

**Participants:** Christophe Wolinski [correspondant], François Charot, Antoine Floch.

We are developing a framework enabling the automatic synthesis of application specific processor extensions. It uses advanced technologies, such as algorithms for graph matching and graph merging together with constraints programming methods. The framework is organized around several modules.

- **CoSaP:** Constraint Satisfaction Problem. The goal of CoSaP is to decouple the statement of a constraint satisfaction problem from the solver used to solve it. The CoSaP model is an Eclipse plugin described using EMF to take advantage of the automatic code generation and of various EMF tools.
- **HCDG:** Hierarchical Conditional Dependency Graph. HCDG is an intermediate representation mixing control and data flow in a single acyclic representation. The control flow is represented as hierarchical guards specifying the execution or the definition conditions of nodes. It can be used in the Gecos compilation framework via a specific pass which translates a CDFG representation into an HCDG.
- **Patterns:** Flexible tools for identification of computational pattern in a graph and graph covering. These tools model the concept of pattern in a graph and provide generic algorithms for the identification of pattern and the covering of a graph. The following sub-problems are addressed: (sub)-graphs isomorphism, patterns generation under constraints, covering of a graph using a library of patterns. Most of the implemented algorithms use constraints programming and rely on the CoSaP module to solve the optimization problem.

5.6. PowWow: Power Optimized Hardware and Software FrameWork for Wireless Motes (AP-L-10-01)

**Participants:** Olivier Sentieys [correspondant], Olivier Berder, Romain Fontaine, Arnaud Carer, Samuel Mouget, Steven Derrien.
PowWow is an open-source hardware and software platform designed to handle wireless sensor network (WSN) protocols and related applications. Based on an optimized preamble sampling medium access (MAC) protocol, geographical routing and protothread library, PowWow requires a lighter hardware system than Zigbee [90] to be processed (memory usage including application is less than 10kb). Therefore, network lifetime is increased and price per node is significantly decreased.

CAIRN’s hardware platform (see Figure 3) is composed of:

- The motherboard, designed to reduce power consumption of sensor nodes, embeds an MSP430 microcontroller and all needed components to process PowWow protocol except radio chip. JTAG, RS232, and I2C interfaces are available on this board.
- The radio chip daughter board is currently based on a TI CC2420.
- The coprocessing daughter board includes a low-power FPGA which allows for hardware acceleration for some PowWow features and also includes dynamic voltage scaling features to increase power efficiency. The current version of PowWow integrates an Actel IGLOO AGL250 FPGA and a programmable DC-DC converter. We have shown that gains in energy of up to 700 can be obtained by using FPGA acceleration on functions like CRC-32 or error detection with regards to a software implementation on the MSP430.

![Figure 3. CAIRN’s PowWow motherboard with radio board connected](image)

PowWow distribution also includes a generic software architecture using event-driven programming and organized into protocol layers (PHY, MAC, LINK, NET and APP). The software is based on Contiki [100], and more precisely on the Protothread library which provides a sequential control flow without complex state machines or full multi-threading.

To optimize the network regarding a particular application and to define a global strategy to reduce energy, PowWow offers the following extra tools: over-the-air reprogramming (and soon reconfiguration), analytical power estimation based on software profiling and power measurements, a dedicated network analyzer to probe and fix transmissions errors in the network. More information can be found at [http://powwow.gforge.inria.fr](http://powwow.gforge.inria.fr).

5.7. SoCLib: Open Platform for Virtual Prototyping of Multi-Processors System on Chip
Participants: François Charot [correspondant], Laurent Perraudeau, Charles Wagner.

SoCLib is an open platform for virtual prototyping of multi-processors system on chip (MP-SoC) developed in the framework of the SoCLib ANR project. The core of the platform is a library of SystemC simulation models for virtual components (IP cores), with a guaranteed path to silicon. All simulation models are written in SystemC, and can be simulated with the standard SystemC simulation environment distributed by the OSCI organization. Two types of models are available for each IP-core: CABA (Cycle Accurate / Bit Accurate), and TLM-DT (Transaction Level Modeling with Distributed Time). All simulation models are distributed as free software. We have developed the simulation model of the NIOSII processor, of the Altera Avalon interconnect, and of the TMS320C62 DSP processor from Texas Instruments. Find more information on its dedicated web page: [http://www.soclib.fr](http://www.soclib.fr).

5.8. OCHRE: On-Chip Randomness Extraction

Participants: Olivier Sentieys [correspondant], Arnaud Carer, Arnaud Tisserand.

Ochre is a set of synthesizable VHDL models for true and pseudo random number generation and hardware accelerated statistical tests. It includes IP cores of different oscillator-based TRNGs, different PRNGs (linear feedback shift registers, cellular automata, AES) and several statistical tests (FIPS 140-2, AIS31, Diehard). This set of IPs has been used to design Ochre V1 and V2 chips and were delivered under license to a company.
CAMUS Team

5. Software

5.1. PolyLib

PolyLib \(^8\) is a C library of polyhedral functions, that can manipulate unions of rational polyhedra of any
dimension, through the following operations: intersection, difference, union, convex hull, simplify, image
and preimage. It was the first to provide an implementation of the computation of parametric vertices of
a parametric polyhedron, and the computation of an Ehrhart polynomial (expressing the number of integer
points contained in a parametric polytope) based on an interpolation method.

It is used by an important community of researchers (in France and the rest of the world) in the area of
compilation and optimization using the polyhedral model. Vincent Loechner is the maintainer of this software.
It is distributed under GNU General Public License version 3 or later, and it has a Debian package maintained
by Serge Guelton (Symbiose Projet, IRISA).

5.2. ZPolyTrans

ZPolyTrans \(^9\) is a C library and a set of executables, that permits to compute the integer transformation of a
union of parametric \(\mathbb{Z}\)-polyhedra (the intersection between lattices and parametric polyhedra), as a union of
parametric \(\mathbb{Z}\)-polyhedra. The number of integer points of the result can also be computed. It is build upon
PolyLib and Barvinok library. This work is based on some theoretical results obtained by Rachid Seghir and
Vincent Loechner, that will be published in ACM TACO in 2011 \([13]\).

It allows for example to compute the number of solutions of a Presburger formula by eliminating existancial
integer variables, or to compute the number of different data accessed by some array accesses contained in an
affine parametric loop nest.

The authors of this software are Rachid Seghir (Univ. Batna, Algeria) and Vincent Loechner. It is distributed
under GNU General Public License version 3 or later.

5.3. NLR

We have developed a program implementing our loop-nest recognition algorithm, detailed in \([7]\). This
standalone, filter-like application takes as input a raw trace and builds a sequence of loop nests that, when
executed, reproduce the trace. It is also able to predict forthcoming values at an arbitrary distance in the
future. Its simple, text-based input format makes it applicable to all kinds of data. These data can take the form
of simple numeric values, or have more elaborate structure, and can include symbols. The program is written
is standard ANSI C. The code can also be used as a library.

We have used this code to evaluate the compression potential of loop nest recognition on memory address
traces, with very good results. We have also shown that the predictive power of our model is competitive with
other models on average. The software is available upon request to anybody interested in trying to apply loop
nest recognition. It has been distributed to a dozen of colleagues around the world.

We plan on using this software as the base for a new tool we currently design, for the analysis of parallel
traces.

5.4. Dynamic version selector

We are developing a toolchain to automatically select between different versions of parallel loop nests, as
described in subsection \(6.2\). It generates the profiling code and selection code from a loop nest source code
and different schedules, expressed in the CLooG format.

\(^8\) http://icps.u-strasbg.fr/PolyLib
\(^9\) http://ZPolyTrans.gforge.inria.fr
Benoît Pradelle (PhD) wrote this toolchain, based on python scripts. It is not yet distributed.

5.5. Binary files decompiler

Our research on efficient memory profiling has lead us to develop a sophisticated decompiler. This tool analyzes x86-64 binary programs and libraries, and extracts various structured representations of the code. It works on a routine per routine basis, and first builds a loop hierarchy to characterize the overall structure of the algorithm. It then puts the code into Static Single Assignment (SSA) form to highlight the fine-grain data-flow between registers and memory. Building on these, it performs the following analyzes:

- All memory addresses are expressed as symbolic expressions involving specific versions of register contents, as well as loop counters. Loop counter definitions are recovered by resolving linearly incremented registers and memory cells, i.e., registers that act as induction variables.
- Most conditional branches are also expressed symbolically (with registers, memory contents, and loop counters). This captures the control-flow of the program, but also helps in defining what amounts to loop “trip-counts”, even though our model is slightly more general, because it can represent any kind of iterative structure.

This tool embodies several passes that, as far as we know, do not exist in any existing similar tool. For instance, it is able to track data-flow through stack slots in most cases. It has been specially designed to extract a representation that can be useful in looking for parallel (or parallelizable) loops [48]. It is the basis of several of our studies.

Because binary program decompilation is especially useful to reduce the cost of memory profiling, our current implementation is based on the Pin binary instrumenter. It uses Pin’s API to analyze binary code, and directly interfaces with the upper layers we have developed (e.g., program skeletonization, or minimal profiling). However, we have been careful to clearly decouple the various layers, and to not use any specific mechanism in designing the binary analysis component. Therefore, we believe that it could be ported with minimal effort, by using a binary file format extractor and a suitable binary code parser. It is also designed to abstract away the detailed instruction set, and should be easy to port (even though we have no practical experience in doing so).

We feel that such a tool could be useful to other researchers, because it makes binary code available under abstractions that have been traditionally available for source code only. If sufficient interest emerges, e.g., from the embedded systems community, or from researchers working on WCET, or from teams working on software security, we are willing to distribute and/or to help make it available under other environments.

5.6. Dynamic dependency analyser

We have recently started developing a dynamic dependence analyzer. Such a tool consumes the trace of memory (or object) accesses, and uses the program structure to list all the data dependences appearing during execution. Data dependences in turn are central to the search for parallel sections of code, with the search for parallel loops being only a particular case of the general problem. Most current works of these questions are either specific to a particular analysis (e.g., computing dependence densities to select code portions for thread-level speculation), or restricted to particular forms of parallelism (e.g., typically to fully parallel loops). Our tool tries to generalize existing approaches, and focuses on the program structures to provide helpful feedback either to a user (as some kind of “smart profiler”), or to a compiler (for feedback-directed compilation). For example, the tool is able to produce a dependence schema for a complete loop nest (instead of just a loop). It also targets irregular parallelism, for example analyzing a loop execution to estimate the expected gain of parallelization strategies like inspector-executor.

We have developed this tool in relation to our minimal profiling research project. However, the tool itself has been kept independent of our profiling infrastructure, getting data from it via a well-defined trace format. This intentional design decision has been motivated by our work on distinct execution environments: first
on our usual x86-64 benchmark programs, and second on less regular, more often written in Java, real-world applications. The latter type of applications is likely the one that will most benefit from such tools, because their intrinsic execution environment does not offer enough structure to allow effective static analysis techniques. Parallelization efforts in this context will most likely rely on code annotations, or specific programming language constructs. Programmers will therefore need tools to help them choose between various constructs. Our tool has this ambition. We already have a working tool-chain for C/C++/Fortran programs (or any binary program). We are in the process of developing the necessary infrastructure to connect the dynamic dependence profiler to instrumented Java programs. Other managed execution environments could be targeted as well, e.g., Microsoft’s .Net architecture, but we have no time and/or workforce to devote to such time-consuming engineering efforts.

5.7. VMAD software and LLVM

For dynamic analysis and optimization of programs, we are developing a virtual machine called VMAD, and specific passes to the LLVM compiler suite, plus a modified Clang frontend. It is fully described in subsection 6.1.

We implemented for now a memory access predictor in loop nests, based on the computation of linear interpolation functions. The profiling is very fast compared to other existing tools, as it samples only the first few iterations of each loop in the nest, then it is deactivated to return to the original, faster version. Other tools like PIN or PEBIL do not support such activation/deactivation mechanism.

New annotations for the final user, taken as input by LLVM, and new VMAD modules will be developed, as these tools have been designed to be very evolving.

Alexandra Jimborean (PhD), Matthieu Herrmann (Master student) and Luis Mastrangelo (Master student) are the main contributors of this software. It is not yet distributed.

5.8. Polyhedral prover

Participants: Nicolas Magaud, Julien Narboux, Éric Violard [correspondant].

We are currently developing a formal proof of program transformations based on the polyhedral model. We use the CompCert verified compiler [54] as a framework. This tool is written in the specification language of Coq. It is connected to the activity described in section 6.6.
5. Software

5.1. Introduction

A major part of the research done in the CARAMEL team is published within software. On the one hand, this enables everyone to check that the algorithms we develop are really efficient in practice; on the other hand, this gives other researchers — and us of course — basic software components on which they — and we — can build other applications.

5.2. GNU MPFR

Participant: Paul Zimmermann [contact].

GNU MPFR is one of the main pieces of software developed by the CARAMEL team. Since end 2006, with the departure of Vincent Lefèvre to ENS Lyon, it has become a joint project between CARAMEL and the ARÉNAIRE project-team (INRIA Grenoble - Rhône-Alpes). GNU MPFR is a library for computing with arbitrary precision floating-point numbers, together with well-defined semantics, and is distributed under the GPL license. All arithmetic operations are performed according to a rounding mode provided by the user, and all results are guaranteed correct to the last bit, according to the given rounding mode.

Several software systems use GNU MPFR, for example: the GCC and GFORTRAN compilers; the SAGE computer algebra system; the KDE calculator Abakus by Michael Pyne; CGAL (Computational Geometry Algorithms Library) developed by the Geometrica project-team (INRIA Sophia Antipolis - Méditerranée); Gappa, by Guillaume Melquiond; Sollya, by Sylvain Chevillard, Mioara Joldeş and Christoph Lauter; Genius Math Tool and the GEL language, by Jiri Lebl; Giac/Xcas, a free computer algebra system, by Bernard Parisse; the iRRAM exact arithmetic implementation from Norbert Müller (University of Trier, Germany); the Magma computational algebra system; and the Wcalc calculator by Kyle Wheeler.

The main developments in 2011 are the release of version 3.0.1 in April, and the release of version 3.1.0 (the “canard à l’orange” release) in October. The main changes in GNU MPFR 3.1.0 are the following: thread local storage (TLS) support is now detected automatically, the squaring and division routines got a major speed up thanks to Mulders’ algorithm [20], and a new divide-by-zero exception was introduced. Note that the automatic TLS support did exhibit several compiler bugs (http://www.loria.fr/~zimmerma/software/compilerbugs.html). We had a developers meeting in January 13-14, and in August GNU MPFR was presented at the GNU Hackers Meeting in Paris.

5.3. MPC

Participant: Paul Zimmermann [contact].

MPC is a floating-point library for complex numbers, which is developed on top of the GNU MPFR library, and distributed under the LGPL license. It is co-written with Andreas Enge (LFANT project-team, INRIA Bordeaux - Sud-Ouest). A complex floating-point number is represented by \( x + iy \), where \( x \) and \( y \) are real floating-point numbers, represented using the GNU MPFR library. The MPC library provides correct rounding on both the real part \( x \) and the imaginary part \( y \) of any result. MPC is used in particular in the TRIP celestial mechanics system developed at IMCCE (Institut de Mécanique Céleste et de Calcul des Éphémérides), and by the Magma computational number theory system.

A new version, MPC 0.9 (Epilobium montanum), was released in February 2011, with new functions, some speed-ups, a few bug fixes, and a logging feature for debugging. Since version 4.5 of GCC, released in May 2010, GCC requires MPC to compute constant complex expressions at compile-time (constant folding), like it requires GNU MPFR since GCC 4.3.
5.4. GMP-ECM

Participants: Cyril Bouvier, Paul Zimmermann [contact].

GMP-ECM is a program to factor integers using the Elliptic Curve Method. Its efficiency comes both from the use of the GMP library, and from the implementation of state-of-the-art algorithms. GMP-ECM contains a library (LIBECM) in addition to the binary program (ECM). The binary program is distributed under GPL, while the library is distributed under LGPL, to allow its integration into other non-GPL software. The Magma computational number theory software and the SAGE computer algebra system both use LIBECM.

During his internship of 4 months in 2011, Cyril Bouvier developed a version of ECM for GPUs. The code was written for NVIDIA GPUs using CUDA. First, the code was written for all NVIDIA cards, and later, it was optimized for the newer Fermi cards. As there is no modular arithmetic library (like GMP) available for GPU, it was necessary to implement a modular arithmetic using array of unsigned integers from scratch, while taking into account constraints of GPU programming. The code was optimized for factoring 1024 bits integers. For now, the code has a throughput roughly four times bigger than GMP-ECM on one core. This code is not yet fully integrated in GMP-ECM but is available in the GMP-ECM svn repository.

The implementation of ECM on GPU uses a different algorithm for scalar multiplication (the binary ladder instead of PRAC) and a different parametrization. This new approach was implemented for CPU in GMP-ECM. It results in a speedup in the execution time of GMP-ECM for finding big factors (more than 20 digits). It will be integrated in the next release of GMP-ECM.

5.5. Finite fields

Participants: Pierrick Gaudry, Emmanuel Thomé [contact].

mpFq is (yet another) library for computing in finite fields. The purpose of mpFq is not to provide a software layer for accessing finite fields determined at runtime within a computer algebra system like Magma, but rather to give a very efficient, optimized code for computing in finite fields precisely known at compile time. mpFq is not restricted to a finite field in particular, and can adapt to finite fields of any characteristic and any extension degree. However, one of the targets being the use in cryptology, mpFq somehow focuses on prime fields and on fields of characteristic two.

mpFq’s ability to generate specialized code for desired finite fields differentiates this library from its competitors. The performance achieved is far superior. For example, mpFq can be readily used to assess the throughput of an efficient software implementation of a given cryptosystem. Such an evaluation is the purpose of the “EBats” benchmarking tool. mpFq entered this trend in 2007, establishing reference marks for fast elliptic curve cryptography: the authors improved over the fastest examples of key-sharing software in genus 1 and 2, both over binary fields and prime fields. These timings are now comparison references for other implementations [27].

The library’s purpose being the generation of code rather than its execution, the working core of mpFq consists of roughly 18,000 lines of Perl code, which generate most of the C code. mpFq is distributed at http://mpfq.gforge.inria.fr/.

The mpFq library has undergone no change in 2011.

5.6. gf2x

Participants: Pierrick Gaudry, Emmanuel Thomé [contact], Paul Zimmermann.

GF2X is a software library for polynomial multiplication over the binary field, developed together with Richard Brent (Australian National University, Canberra, Australia). It holds state-of-the-art implementation of fast algorithms for this task, employing different algorithms in order to achieve efficiency from small to large operand sizes (Karatsuba and Toom-Cook variants, and eventually Schönhage’s or Cantor’s FFT-like algorithms). GF2X takes advantage of specific processors instruction (SSE, PCLMULQDQ).
The current version of GF2X is 1.0, released in 2010 under the GNU GPL. Since 2009, GF2X can be use as an auxiliary package for the widespread software library NTL, as of version 5.5. There has been no update of GF2X in 2011, but the software is still maintained. An LGPL-licensed portion of GF2X is also part of the CADO-NFS software package.

5.7. CADO-NFS

Participants: Jérémie Detrey, Pierrick Gaudry, Lionel Muller, Emmanuel Thomé [contact], Paul Zimmermann.

CADO-NFS is a program to factor integers using the Number Field Sieve algorithm (NFS), developed in the context of the ANR-CADO project (November 2006 to January 2010).

NFS is a complex algorithm which contains a large number of sub-algorithms. The implementation of all of them is now complete, but still leaves some places to be improved. Compared to existing implementations, the CADO-NFS implementation is already a reasonable player. Several factorizations have been completed using our implementations.

Since 2009, the source repository of CADO-NFS is publicly available for download. On October 28, 2011, the 1.1 version of CADO-NFS has been released. Several improvements to the program have been obtained, in practically all areas of the program. In particular, the polynomial selection code described by Thorsten Kleinjung at the CADO workshop in 2008 is now used within CADO-NFS, together with some efficient root-sieve code written by Shi Bai (Australian National University). Overall, CADO-NFS keeps improving its competitiveness over alternative code bases. The lattice siever now supports a sieving region of $2^{31}$ ($I = 16$); its code has been deeply reorganized to allow future improvements that we have in mind but were difficult to implement (proper sieving of powers, sieve according to the parities of the coordinates of the location). The executables in the linear algebra step have been reorganized (now using shared libraries), and now use a code generation mechanism built on top of the MPFQ library for the arithmetic parts. This is in particular meant to ease future accommodation of other base fields that GF$(2)$, which is a requirement for adapting CADO-NFS to discrete logarithm computation. The MPI performance of the linear algebra code has been optimized. Some experimental scripts have been added to execute the sieve on a cluster; these scripts rely on the OAR job scheduler being used, and exploit its “besteffort” mode.

The largest factorizations performed by CADO-NFS in 2011 are a 170-digit integer from aliquot sequence 660 and a 171-digit integer from aliquot sequence 966.

5.8. AVIsogenies

Participants: Gaëtan Bisson [contact], Romain Cosset.

AVIsogenies (Abelian Varieties and Isogenies) is a Magma package for working with abelian varieties, with a particular emphasis on explicit isogeny computation; it has been publicly released under the LGPLv2+ license in 2010.

Its prominent feature is the computation of $(\ell, \ell)$-isogenies between Jacobian varieties of genus-2 hyperelliptic curves over finite fields; practical runs have involved values of $\ell$ in the hundreds. It also provides procedures for exploring and drawing isogeny graphs, and for computing various complex-multiplication-related structures, such as Shimura’s gothic C group.

In 2011, two incremental versions have been released. They provide the following new features: the characteristic 2 is now supported, and the complete addition laws of [23] have been implemented.

The package can be obtained at http://avisogenies.gforge.inria.fr/.
5. Software

5.1. Morpus/MMDEX
An anti-virus software based on morphological analysis, Dépôt APP du logiciel MMDEX, 2009, IDDN.FR.001.300033.000.R.P.2009.000.10000

5.2. PYMS
Online disassembler. http://pym86.appspot.com/

5.3. TraceSurfer

5.4. Crème Brûlée
Crème Brûlée is an experimental Javascript dynamic instrumentation engine. http://code.google.com/p/cremebrulee/
CASCADE Project-Team

5. Software

5.1. MitMTool

Participants: Charles Bouillaguet, Patrick Derbez, Pierre-Alain Fouque.

The purpose of MitMTool is to look for guess-and-determine and meet-in-the-middle attacks on AES and AES-based constructions. This tool allows us to improve known attacks on round-reduced versions of AES, on the LEX stream-cipher on the PELICAN Message Authentication Code and on fault attack on AES. Basically, it solves the problem to find all the solutions of a linear system of equations on the variables \( x \) and \( S(x) \) where \( S \) is an inert function. The tool allows to compute the complexity of some good attack as well as the C code of the attack. We verify that the complexity estimates are accurate using experiments. We also use it to find one solution of the system for chosen-key differential attacks. There are mainly two tools: the first one only looks for guess-and-determine attack and tries to propagate some knowledge and guesses value when it cannot find automatically the value of some variable. The second tool uses the technique of the first tool and more advanced technique to take into account attacks with memory that use the meet-in-the-middle attack.

5.2. ProVerif

Participants: Bruno Blanchet, Vincent Cheval.

ProVerif (www.proverif.ens.fr) is an automatic security protocol verifier, in the formal model (so called Dolev–Yao model). In this model, cryptographic primitives are considered as black boxes. This protocol verifier is based on an abstract representation of the protocol by Horn clauses. Its main features are:

- It can handle many different cryptographic primitives, including shared- and public-key cryptography (encryption and signatures), hash functions, and Diffie–Hellman key agreements, specified both as rewrite rules or as equations.
- It can handle an unbounded number of sessions of the protocol (even in parallel) and an unbounded message space. This result has been obtained thanks to some well-chosen approximations. This means the verifier can give false attacks, but if it claims that the protocol satisfies some property, then the property is actually satisfied. ProVerif also provides attack reconstruction: when it cannot prove a property, it tries to reconstruct an attack, that is, an execution trace of the protocol that falsifies the desired property.

The ProVerif verifier can prove the following properties:

- secrecy (the adversary cannot obtain the secret);
- authentication and more generally correspondence properties, of the form "if an event has been executed, then other events have been executed as well";
- strong secrecy (the adversary does not see the difference when the value of the secret changes);
- equivalences between processes that differ only by terms;

ProVerif has been used by researchers for studying various kinds of protocols, including electronic voting protocols, certified email protocols, and zero-knowledge protocols. It has been used as a back-end for the tool TULAFALE implemented at Microsoft Research Cambridge, which verifies web services protocols. It has also been used as a back-end for verifying implementations of protocols in F# (a dialect of ML included in .NET), by Microsoft Research Cambridge and the joint INRIA-Microsoft research center.

ProVerif is freely available on the web, at www.proverif.ens.fr, under the GPL license.
5.3. CryptoVerif

Participants: Bruno Blanchet, David Cadé.

CryptoVerif (www.cryptoverif.ens.fr) is an automatic protocol prover sound in the computational model. In this model, messages are bitstrings and the adversary is a polynomial-time probabilistic Turing machine. CryptoVerif can prove:

- secrecy;
- correspondences, which include in particular authentication.

CryptoVerif provides a generic mechanism for specifying the security assumptions on cryptographic primitives, which can handle in particular symmetric encryption, message authentication codes, public-key encryption, signatures, hash functions, Diffie-Hellman key agreement.

The generated proofs are proofs by sequences of games, as used by cryptographers. These proofs are valid for a number of sessions polynomial in the security parameter, in the presence of an active adversary. CryptoVerif can also evaluate the probability of success of an attack against the protocol as a function of the probability of breaking each cryptographic primitive and of the number of sessions (exact security).

CryptoVerif is still at a rather early stage of development, but it has already been used for a study of Kerberos in the computational model. It is also used as a back-end for verifying implementations of protocols in F# at Microsoft Research Cambridge and at the joint INRIA-Microsoft research center.

CryptoVerif is freely available on the web, at www.cryptoverif.ens.fr, under the CeCILL-B license.
5. Software

5.1. Protocol Verification Tools

Participants: Pierre-Cyrille Héam, Olga Kouchnarenko, Michaël Rusinowitch, Mathieu Turuani, Laurent Vigneron.

5.1.1. AVISPA

Cassis has been one of the 4 partners involved in the European project AVISPA, which has resulted in the distribution of a tool for automated verification of security protocols, named AVISPA Tool. It is freely available on the web\(^2\) and it is well supported. The AVISPA Tool compares favourably to related systems in scope, effectiveness, and performance, by (i) providing a modular and expressive formal language for specifying security protocols and properties, and (ii) integrating 4 back-ends that implement automatic analysis techniques ranging from protocol falsification (by finding an attack on the input protocol) to abstraction-based verification methods for both finite and infinite numbers of sessions.

5.1.2. CL-AtSe

We develop, as a first back-end of AVISPA, CL-AtSe, a Constraint Logic based Attack Searcher for cryptographic protocols. The CL-AtSe approach to verification consists in a symbolic state exploration of the protocol execution, for a bounded number of sessions. This necessary restriction (for decidability, see \cite{85}) allows CL-AtSe to be correct and complete, i.e., any attack found by CL-AtSe is a valid attack, and if no attack is found, then the protocol is secure for the given number of sessions. Each protocol step is represented by a constraint on the protocol state. These constraints are checked lazily for satisfiability, where satisfiability means reachability of the protocol state. CL-AtSe includes a proper handling of sets (operations and tests), choice points, specification of any attack states through a language for expressing secrecy, authentication, fairness, non-abuse freeness, advanced protocol simplifications and optimizations to reduce the problem complexity, and protocol analysis modulo the algebraic properties of cryptographic operators such as XOR (exclusive or) and Exp (modular exponentiation). The handling of XOR and Exp has required to implement an optimized version of the combination algorithm of Baader & Schulz \cite{76} for solving unification problems in disjoint unions of arbitrary theories.

CL-AtSe has been successfully used \cite{75} to analyse France Telecom R&D, Siemens AG, IETF, or Gemalto protocols in funded projects. It is also employed by external users, e.g., from the AVISPA's community. Moreover, CL-AtSe achieves very good analysis times, comparable and sometimes better than state-of-the-art tools in the domain (see \cite{90} for tool details and precise benchmarks).

5.1.3. TA4SP

We have developed, as a second back-end of AVISPA, TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols), an automata based tool dedicated to the validation of security protocols for an unbounded number of sessions. This tool provides automatic computations of over- and under-approximations of the knowledge accessible by an intruder. This knowledge is encoded as a regular tree language and protocol steps and intruder abilities are encoded as a term rewriting system. When given a reachability problem such as secrecy, TA4SP reports that (1) the protocol is safe if it manages to compute an over-approximation of intruder's knowledge that does not contain a secret term or (2) the protocol is unsafe in the rewrite model if it manages to compute an underapproximation of intruder's knowledge containing a secret term or (3) I don’t know otherwise. TA4SP has verified 28 industrial protocols and case (3) occurred only once, for Kaochow protocol version 2.

\(^2\) http://www.avispa-project.org
TA4SP handles protocols using operators with algebraic properties. Thanks to a recent quadratic completion algorithm new experimental results have been obtained, for example for the Encrypted Key Exchange protocol (EKE2) using the exponential operator.

Recently, TA4SP was used in [89] to analyse a hierarchy of authentication properties.

5.2. Testing Tools

Participants: Fabrice Bouquet, Frédéric Dadeau, Philippe Paquelier.

In December 2008, we have started the redevelopment of our original testing tools environment, with two objectives: first, refactoring the existing developments, and, second, providing an open platform aiming at gathering together the various developments, increasing the reusability of components. The resulting platform, named Hydra, is a Eclipse-like platform, based on Plug-ins architecture. Plug-ins can be of five kinds: parser is used to analyze source files and build an intermediate format representation of the source; translator is used to translate from a format to another or to a specific file; service denotes the application itself, i.e. the interface with the user; library denotes an internal service that can be used by a service, or by other libraries; tool encapsulates an external tool. The following services have been developed so far:

- **BZPAnimater:** performs the animation of a BZP model (a B-like intermediate format);
- **Angluin:** makes it possible to perform a machine learning algorithm (à la Angluin) in order to extract an abstraction of a system behavior;
- **UML2SMT:** aims at extracting first order logic formulas from the UML Diagrams and OCL code of a UML/OCL model to check them with a SMT solver.

These services involve various libraries (sometimes reusing each other), and rely on several tool plug-ins that are: SMTProver (encapsulating Z3 solver), PrologTools (encapsulating CLPS-B solver), Grappa (encapsulating a graph library). We are currently working on transferring the existing work on test generation from B abstract machines, JML, and statecharts using constraint solving techniques.

5.3. Collaborative Tools

Participants: Abdessamad Imine, Asma Cherif.

The collaborative tools allow us to manage collaborative works on shared documents using flexible access control models. These tools have been developed in order to validate and evaluate our approach on combining collaborative edition with optimistic access control.

- **P2PEdit.** This prototype is implemented in Java and supports the collaborative editing of HTML pages and it is deployed on P2P JXTA platform\(^3\). In our prototype, a user can create a HTML page from scratch by opening a new collaboration group. Other users (peers) may join the group to participate in HTML page editing, as they may leave this group at any time. Each user can dynamically add and remove different authorizations for accessing to the shared document according the contribution and the competence of users participating in the group. Using JXTA platform, users exchange their operations in real-time in order to support WYSIWIS (What You See Is What I See) principle. Furthermore, the shared HTML document and its authorization policy are replicated at the local memory of each user. To deal with latency and dynamic access changes, an optimistic access control technique is used where enforcement of authorizations is retroactive.

- **P2PCalendar.** To extend our collaboration and access control models to mobile devices, we implemented a shared calendar on iPhone OS which is decentralized and scalable (i.e. it can be used over both P2P and ad-hoc networks) [58]. This application aims to make a collaborative calendar where users can simultaneously modify events (or appointments) and control access on events. The access rights are determined by the owner of an event. The owner decides who is allowed to access the event and what privileges they have. Likewise to our previous tool, the calendar and its authorization policy are replicated at every mobile device.

\(^3\) [http://www.sun.com/software/jxta/](http://www.sun.com/software/jxta/)
5.4. Other Tools

Several software tools described in previous sections are using tools that we have developed in the past. For instance BZ-TT uses the set constraints solver CLPS. Note that the development of the SMT prover haRVey has been stopped. The successor of haRVey is called veriT and is developed by David Déharbe (UFRN Natal, Brasil) and Pascal Fontaine (Veridis team).
CELTIQUE Project-Team

4. Software

4.1. Javalib

Participants: Frédéric Besson [correspondant], David Pichardie, Vincent Monfort.

Javalib is an efficient library to parse Java .class files into OCaml data structures, thus enabling the OCaml programmer to extract information from class files, to manipulate and to generate valid .class files. See also the web page http://sawja.inria.fr/.

- Version: 2.2
- Programming language: Ocaml

4.2. SAWJA

Participants: Frédéric Besson [correspondant], David Pichardie, Vincent Monfort.

Sawja is a library written in OCaml, relying on Javalib to provide a high level representation of Java bytecode programs. It name comes from Static Analysis Workshop for JAva. Whereas Javalib is dedicated to isolated classes, Sawja handles bytecode programs with their class hierarchy and with control flow algorithms. Moreover, Sawja provides some stackless intermediate representations of code, called JBir and A3Bir. The transformation algorithm, common to these representations, has been formalized and proved to be semantics-preserving. See also the web page http://sawja.inria.fr/.

- Version: 1.2
- Programming language: Ocaml

4.3. Timbuk

Participant: Thomas Genet [correspondant].

Timbuk is a library of OCAML functions for manipulating tree automata. More precisely, Timbuk deals with finite bottom-up tree automata (deterministic or not). This library provides the classical operations over tree automata (intersection, union, complement, emptiness decision) as well as exact or approximated sets of terms reachable by a given term rewriting system. This last operation can be certified using a checker extracted from a Coq specification.

- Version: 3.1
- Programming language: Ocaml
COMETE Project-Team

5. Software

5.1. A model checker for the probabilistic asynchronous π-calculus

**Participants:** Miguel Andrés [correspondant], Catuscia Palamidessi.

In collaborations with Dave Parker and Marta Kwiatkowska, we are developing a model checker for the probabilistic asynchronous π-calculus. Case studies with Fair Exchange and MUTE, an anonymous peer-to-peer file sharing system, are in progress.

Technically we use MMC as a compiler to encode the probabilistic π-calculus into certain PRISM representation, which will then be verified against PCTL using PRISM. The transitional semantics defined in MMC can be reused to derive the symbolic transition graphs of a probabilistic process. The code for derivation will work as an add-on to MMC under XSB and invoke a graph traversal to enumerate all reachable nodes and transitions of the probabilistic process.

In the meanwhile we are also attempting a direct and more flexible approach to the development of a model checker for the probabilistic π-calculus, using OCaml. This should allow to extend the language more easily, to include cryptographic primitives and other features useful for the specification of security protocols. As the result of our preliminary steps in this direction we have developed a rudimentary model checker, available at the following URL: [http://vamp.gforge.inria.fr/](http://vamp.gforge.inria.fr/).

5.2. PRISM model generator

**Participants:** Konstantinos Chatzikokolakis [correspondant], Catuscia Palamidessi.

This software generates PRISM models for the Dining Cryptographers and Crowds protocols. It can also use PRISM to calculate the capacity of the corresponding channels. More information can be found in [33] and in the file README file width instructions at the URL [http://www.lix.polytechnique.fr/comete/software/README-anonmodels.html](http://www.lix.polytechnique.fr/comete/software/README-anonmodels.html).

The software can be downloaded at [http://www.lix.polytechnique.fr/comete/software/anonmodels.tar.gz](http://www.lix.polytechnique.fr/comete/software/anonmodels.tar.gz). These scripts require Perl to run and have been tested in Linux. The GUI of the corners tool also requires the Perl/Tk library. Finally some parts of the model generator tool require PRISM and gnuplot to be installed.

5.3. Calculating the set of corner points of a channel

**Participants:** Konstantinos Chatzikokolakis [correspondant], Catuscia Palamidessi.

The corner points can be used to compute the maximum probability of error and to improve the Hellman-Raviv and Santhi-Vardy bounds. More information can be found in [34] and in the file README file width instructions at the URL [http://www.lix.polytechnique.fr/comete/software/README-corners.html](http://www.lix.polytechnique.fr/comete/software/README-corners.html).

The software can be downloaded at [http://www.lix.polytechnique.fr/comete/software/corners.tar.gz](http://www.lix.polytechnique.fr/comete/software/corners.tar.gz). These scripts require Perl to run and have been tested in Linux. The GUI of the corners tool also requires the Perl/Tk library. Finally some parts of the model generator tool require PRISM and gnuplot to be installed.

5.4. MMCsp, a compiler for the π-calculus

**Participants:** Peng Wu [correspondant], Catuscia Palamidessi.

MMCsp is a compiler from a simple probabilistic π-calculus to PRISM (http://www.prismmodelchecker.org/manual/Main/Introduction), models. It is built on XSB (http://xsb.sourceforge.net/), a tabled logic programming system, and generates the symbolic semantic representation of a probabilistic pi-calculus term in text. A separate Java program then translates this semantic representation into a probabilistic model for PRISM.
The tool was developed by Peng Wu during his postdoc period in Comète in the context of the collaboration between the teams Comète and PRISM under the INRIA/ARC Project ProNoBib (http://www.lsv.ens-cachan.fr/~goubault/ProNobis/index.html). It is based on the papers [41] and [38].

The source code is free and can be downloaded from http://www.cs.ucl.ac.uk/staff/p.wu/mmc_sp_manual.html.
5. Software

5.1. Introduction

This section lists and briefly describes the software developments conducted within Compsys. Most are tools that we extend and maintain over the years. They now concern two activities only: a) the development of tools linked to polyhedra and loop/array transformations, b) the development of algorithms within the back-end compiler of STMicroelectronics.

Many tools based on the polyhedral representation of codes with nested loops are now available. They have been developed and maintained over the years by different teams, after the introduction of Paul Feautrier’s Pip, a tool for parametric integer linear programming. This “polytope model” view of codes is now widely accepted: it used by Inria projects-teams Cairn and Alchemy/Parkas, PIPS at École des Mines de Paris, Surf from Stanford University, Compaan at Berkeley and Leiden, PiCo from the HP Labs (continued as PicoExpress by Synfora and now Synopsis), the DTSE methodology at Imec, Sadayappan’s group at Ohio State University, Rajopadhye’s group at Colorado State’s University, etc. More recently, several compiler groups have shown their interest in polyhedral methods, e.g., the GCC group, IBM, and Reservoir Labs, a company that develops a compiler fully-based on the polytope model and on the techniques that we (the french community) introduced for loop and array transformations. Polyhedra are also used in test and certification projects (Verimag, Lande, Veritecs). Now that these techniques are well-established and disseminated in and by other groups, we prefer to focus on the development of new techniques and tools, which are described here.

The other activity concerns the developments within the compiler of STMicroelectronics. These are not stand-alone tools, which could be used externally, but algorithms and data structures implemented inside the LAO back-end compiler, year after year, with the help of STMicroelectronics colleagues. As these are also important developments, it is worth mentioning them in this section. They are also completed by important efforts for integration and evaluation within the complete STMicroelectronics toolchain. They concern exact methods (ILP-based), algorithms for aggressive optimizations, techniques for just-in-time compilation, and for improving the design of the compiler.

5.2. Pip

Participants: Cédric Bastoul [MCF, IUT d’Orsay], Paul Feautrier.

Paul Feautrier is the main developer of Pip (Parametric Integer Programming) since its inception in 1988. Basically, Pip is an “all integer” implementation of the Simplex, augmented for solving integer programming problems (the Gomory cuts method), which also accepts parameters in the non-homogeneous term. Pip is freely available under the GPL at http://www.piplib.org. Pip is widely used in the automatic parallelization community for testing dependences, scheduling, several kind of optimizations, code generation, and others. Beside being used in several parallelizing compilers, Pip has found applications in some unconnected domains, as for instance in the search for optimal polynomial approximations of elementary functions (see the Inria project Arénaire).

5.3. Syntol

Participants: Hadda Cherroun [Former PhD student in Compsys], Paul Feautrier.
Syntol is a modular process network scheduler. The source language is C augmented with specific constructs for representing communicating regular process (CRP) systems. The present version features a syntax analyzer, a semantic analyzer to identify DO loops in C code, a dependence computer, a modular scheduler, and interfaces for CLooG (loop generator developed by C. Bastoul) and Cl@k (see Sections 5.4 and 5.6). The dependence computer now handles casts, records (structures), and the modulo operator in subscripts and conditional expressions. The latest developments are, firstly, a new code generator, and secondly, several experimental tools for the construction of bounded parallelism programs.

- The new code generator, based on the ideas of Boulet and Feautrier [31], generates a counter automaton that can be presented as a C program, as a rudimentary VHDL program at the RTL level, as an automaton in the Aspic input format, or as a drawing specification for the DOT tool.
- Hardware synthesis can only be applied to bounded parallelism programs. Our present aim is to construct threads with the objective of minimizing communications and simplifying synchronization. The distribution of operations among threads is specified using a placement function, which is found using techniques of linear algebra and combinatorial optimization.

5.4. Cl@k

Participants: Christophe Alias, Fabrice Baray [Mentor, Former post-doc in Compsys], Alain Darte.

Cl@k (Critical LAttice Kernel) is a stand-alone optimization tool useful for the automatic derivation of array mappings that enable memory reuse, based on the notions of admissible lattice and of modular allocation (linear mapping plus modulo operations). It has been developed in 2005-2006 by Fabrice Baray, former post-doc Inria under Alain Darte’s supervision. It computes or approximates the critical lattice for a given 0-symmetric polytope. (An admissible lattice is a lattice whose intersection with the polytope is reduced to 0; a critical lattice is an admissible lattice with minimal determinant.)

Its application to array contraction has been implemented by Christophe Alias in a tool called Bee (see Section 5.6). Bee uses Rose as a parser, analyzes the lifetimes of the elements of the arrays to be compressed, and builds the necessary input for Cl@k, i.e., the 0-symmetric polytope of conflicting differences. Then, Bee computes the array contraction mapping from the lattice provided by Cl@k and generates the final program with contracted arrays. See previous reports for more details on the underlying theory. Cl@k can be viewed as a complement to the Polylib suite, enabling yet another kind of optimizations on polyhedra. Initially, Bee was the complement of Cl@k in terms of its application to memory reuse. Now, Bee is a stand-alone tool that contains more and more features for program analysis and loop transformations.

5.5. PoCo

Participant: Christophe Alias.

PoCo is a polyhedral compilation framework providing many features to quickly prototype program analysis and optimizations in the polyhedral model. Essentially, PoCo provides:

- C front-end extracting the polyhedral representation of the input program. The parser itself is based on EDG (via ROSE), an industrial C/C++ parser from Edison group used in Intel compilers.
- Extended language of pragmas to feed the source code with compilation directives (a schedule, for example).
- Symbolic layer on polyhedral libraries POLYLIB (set operations on polyhedra) and PIPLIB (parameterized ILP). This feature simplifies drastically the developer task.
- Dependence analysis (polyhedral dependence graph, array dataflow analysis), array region analysis, array liveness analysis.
- C and VHDL code generation based on the ideas of P. Boulet and P. Feautrier [31].
The array dataflow analysis (ADA) of PoCo has been extended to a FADA (Fuzzy ADA) by M. Belaoucha, former PhD student at Université de Versailles. FADALib is available at http://www.prism.uvsq.fr/~bem/fadalib/.

PoCo has been developed by Christophe Alias. It represents more than 19000 lines of C++ code. The tools Bee, Chuba, and RanK presented thereafter make an extensive use of PoCo abstractions.

5.6. Bee

Participants: Christophe Alias, Alain Darte.

Bee is a source-to-source optimizer that contracts the temporary arrays of a program under scheduling constraints. Bee bridges the gap between the mathematical optimization framework described in [32] and implemented in Cl@k (Section 5.4), and effective source-to-source array contraction. Bee applies a precise lifetime analysis for arrays to build the mathematical input of Cl@k. Then, Bee derives the array allocations from the basis found by Cl@k and generates the C code accordingly. Bee is – to our knowledge – the only complete array contraction tool.

Bee is sensitive to the program schedule. This latter feature enlarges the application field of array contraction to parallel programs. For instance, it is possible to mark a loop to be software-pipelined (with an affine schedule) and to let Bee find an optimized array contraction. But the most important application is the ability to optimize communicating regular processes (CRP). Given a schedule for every process, Bee can compute an optimized size for the channels, together with their access functions (the corresponding allocations). We currently use this feature in source-to-source transformations for high-level synthesis (see Section 3.3).

- Bee was made available to STMICROELECTRONICS as a binary.
- Bee will be transferred to the (incubated) start-up Zettice, initiated by Alexandru Plesco.
- Bee is used as an external tool by the compiler GECOS developed in the Cairn team at IRISA.

Bee has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 2400 lines of C++ code.

5.7. Chuba

Participants: Christophe Alias, Alain Darte, Alexandru Plesco.

Chuba is a source-level optimizer that improves a C program in the context of the high-level synthesis (HLS) of hardware. Chuba is an implementation of the work described in the PhD thesis of Alexandru Plesco. The optimized program specifies a system of multiple communicating accelerators, which optimize the data transfers with the external DDR memory. The program is divided into blocks of computations obtained thanks to tiling techniques, and, in each block, data are fetched by block to reduce the penalty due to line changes in the DDR accesses. Four accelerators achieve data transfers in a macro-pipeline fashion so that data transfers and computations (performed by a fifth accelerator) are overlapped.

So far, the back-end of Chuba is specific to the HLS tool C2H but the analysis is quite general and adapting Chuba to other HLS tools should be possible. Besides, it is interesting to mention that the program analysis and optimizations implemented in Chuba address a problem that is also very relevant in the context of GPGPUs.

Chuba has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 900 lines of C++. The reduced size of Chuba is mainly due to the high-level abstractions provided by PoCo.

5.8. C2fsm

Participant: Paul Feautrier.
C2fsm is a general tool that converts an arbitrary C program into a counter automaton. This tool reuses the parser and pre-processor of Syntol, which has been greatly extended to handle while and do while loops, goto, break, and continue statements. C2fsm reuses also part of the code generator of Syntol and has several output formats, including FAST (the input format of Aspic), a rudimentary VHDL generator, and a DOT generator which draws the output automaton. C2fsm is also able to do elementary transformations on the automaton, such as eliminating useless states, transitions and variables, simplifying guards, or selecting cut-points, i.e., program points on loops that can be used by RanK to prove program termination.

5.9. RanK

**Participants:** Christophe Alias, Alain Darte, Paul Feautrier, Laure Gonnord.

RanK is a software tool that can prove the termination of a program (in some cases) by computing a ranking function, i.e., a mapping from the operations of the program to a well-founded set that decreases as the computation advances. In case of success, RanK can also provide an upper bound of the worst-case time complexity of the program as a symbolic affine expression involving the input variables of the program (parameters), when it exists. In case of failure, RanK tries to prove the non-termination of the program and then to exhibit a counter-example input. This last feature is of great help for program understanding and debugging, and has already been experimented.

The input of RanK is an integer automaton, computed by C2fsm (see Section 5.8), representing the control structure of the program to check. RanK uses the Aspic tool, developed by Laure Gonnord during her PhD thesis, to compute automaton invariants. RanK has been used to discover successfully the worst-case time complexity of many benchmarks programs of the community. It uses the libraries Piplib and Polylib.

RanK has been implemented by Christophe Alias, using the compiler infrastructure PoCo. It represents more than 3000 lines of C++.

5.10. Simplifiers

**Participant:** Paul Feautrier.

The aim of the simple library is to simplify boolean formulas on affine inequalities. It works by detecting redundant inequalities in the representation of the subject formula as an ordered binary decision diagram. It uses PIP for testing the feasibility – or unfeasibility – of a conjunction of affine inequality.

The library is written in Java and is presented as a collection of class files. For experimentation, several front-ends have been written They differ mainly in their input syntax, among which are a C like syntax, the Mathematica and SMTLib syntaxes, and an ad hoc Quast syntax.

5.11. LAO Developments in Aggressive Compilation

**Participants:** Benoît Boissinot, Florent Bouchez, Florian Brandner, Quentin Colombet, Alain Darte, Benoît Dupont de Dinechin [Kalray], Christophe Guillon [STMicroelectronics], Sebastian Hack [Former post-doc in Compsys], Fabrice Rastello, Cédric Vincent [Former student in Compsys].

Our aggressive optimization techniques are all implemented in stand-alone experimental tools (as for example for register coalescing algorithms) or within LAO, the back-end compiler of STMicroelectronics, or both. They concern SSA construction and destruction, instruction-cache optimizations, register allocation. Here, we report only our more recent activities, which concern register allocation.
Our developments on register allocation within the STMicroelectronics compiler started when Cédric Vincent (bachelor degree, under Alain Darte supervision) developed a complete register allocator in LAO, the assembly-code optimizer of STMicroelectronics. This was the first time a complete implementation was done with success, outside the MCDT (now CEC) team, in their optimizer. Since then, new developments are constantly done, in particular by Florent Bouchet, advised by Alain Darte and Fabrice Rastello, as part of his master internship and PhD thesis. In 2009, Quentin Colombet started to develop and integrate into the main trunk of LAO a full implementation of a two-phases register allocation. This implementation now includes two different decoupled spilling phases, the first one as described in Sebastian Hack’s PhD thesis and a new ILP-based solution (see Section 6.2). It also includes an up-to-date graph-based register coalescing. Finally, since all these optimizations take place under SSA form, it includes also a mechanism for going out of colored-SSA (register-allocated SSA) form that can handle critical edges and does further optimizations (see for example Section 6.3).

5.12. LAO Developments in JIT Compilation

Participants: Benoit Boissinot, Florian Brandner, Alain Darte, Benoît Dupont de Dinechin [Kalray], Christophe Guillon [STMicroelectronics], Fabrice Rastello.

The other side of our work in the STMicroelectronics compiler LAO has been to adapt the compiler to make it more suitable for JIT compilation. This means lowering the time and space complexity of several algorithms. In particular we implemented our fast out-of-SSA translation method, and we programmed and tested various ways to compute the liveness information as described in Section 6.6. Recent efforts (see Section 6.4) also focused on developing a tree-scan register allocator for the JIT part of the compiler, in particular a JIT conservative coalescing. The technique is to bias the tree-scan coalescing, taking into account register constraints, with the result of a JIT aggressive coalescing.

5.13. Low-Level Exchange Format (TireX) and Minimalist Intermediate Representation (MinIR)

Participants: Christophe Guillon [STMicroelectronics], Fabrice Rastello, Benoît Dupont de Dinechin [Kalray].

Most compilers define their own intermediate representation (IR) to be able to work on a program. Sometimes, they even use a different representation for each representation level, from source code parsing to the final object code generation. MinIR (Minimalist Intermediate Representation) is a new intermediate representation, designed to ease the interconnection of compilers, static analyzers, code generators, and other tools. In addition to the specification of MinIR, generic core tools have been developed to offer a basic toolkit and to help the connection of client tools. MinIR generators exist for several compilers, and different analyzers are developed as a testbed to rapidly prototype different static analyses over SSA code. This new common format enables the comparison of the code generator of several production compilers, and simplifies the connection of external tools to existing compilers.

MinIR has been extended into TireX, a Textual Intermediate Representation for EXchanging target-level information between compiler optimizers and whole or parts of code generators (aka compiler backend). The first motivation for this intermediate representation is to factor target-specific compiler optimizations into a single component, in case several compilers need to be maintained for a particular target (e.g., operating system compiler and application code compiler). Another motivation is to reduce the runtime cost of JIT compilation and of mixed mode execution, since the program to compile is already in a representation lowered to the level of the target processor. Besides the lowering at the target level, the extensions of MinIR include the program data stream and loop scoped information. TireX is currently produced by the Open64/Path64 and the LLVM compilers, with a GCC producer under work. It is consumed by the LAO code generator. Detailed information, generic core tools, and LLVM IR based generator for MinIR are available at http://www.assembla.com/spaces/minir-dev/wiki. Open64/Path64 emitter for TireX and its LAO back-end are available at https://compilation.ens-lyon.fr/. MinIR was presented at WIR’11 [17].
5. Software

5.1. BIOCHAM

Participants: François Fages, Steven Gay, Dragana Jovanovska, Aurélien Rizk, Sylvain Soliman.

The Biochemical Abstract Machine BIOCHAM [29] is a modeling environment for systems biology distributed as open-source since 2003. Current version is v3.3. BIOCHAM uses a compositional rule-based language for modeling biochemical systems, allowing patterns for expressing set of rules in a compact form. This rule-based language is compatible with the Systems Biology Markup Language (SBML) and is interpreted with three semantics corresponding to three abstraction levels:

1. the boolean semantics (presence or absence of molecules),
2. the differential semantics (concentrations of molecules),
3. the stochastic semantics (discrete numbers of molecules).

Based on this formal framework, BIOCHAM features:

- Boolean and numerical simulators (Rosenbrock’s method for the differential semantics, Gillespie’s algorithm with tau leaping for the stochastic semantics);
- a temporal logic language (CTL for qualitative models and QFTL(R) with numerical constraints for quantitative models) for formalizing biological properties such as reachability, checkpoints, oscillations or stability, and checking them automatically with model-checking techniques;
- automatic search procedures to infer parameter values, initial conditions and even reaction rules from temporal logic properties;
- automatic detection of invariants, through constraint-based analysis of the underlying Petri net;
- an SBGN-compatible reaction graph editor;
- an event handler allowing the encoding of hybrid models and formalisms [30].

BIOCHAM is implemented in GNU-Prolog and interfaced to the symbolic model checker NuSMV and to the continuous optimization tool CMAES developed by the EPI TAO.

5.2. Nicotine

Participant: Sylvain Soliman.

Nicotine is a GNU Prolog framework dedicated to the analysis of Petri nets. It was originally built for the computation of invariants using GNU Prolog’s CLP(FD) solver but has been further extended to allow import/export of various Petri nets formats. It provides as independent modules different features that can sometimes also be integrated in BIOCHAM, like SEPI computation, or left aside, like unambiguous ODE to Petri net conversion, since a more general heuristic conversion is developed for BIOCHAM.

5.3. Spatio-temporal simulation environment (STSE)

Participant: Szymon Stoma.

The overall goal of this project is to provide a software platform gathering a set of open-source tools and workflows facilitating spatio-temporal simulations (preferably of biological systems) based on microscopy data. The framework currently contains modules to digitize, represent, analyze, and model spatial distributions of molecules in static and dynamic structures (e.g. growing). A strong accent is put on the experimental verification of biological models by actual, spatio-temporal data acquired using microscopy techniques. Project was initially started at Humboldt University Berlin and moved to INRIA with its founder. Project webpage is: http://stse-software.org.
5.4. YeastTracker

Participant: Jannis Uhlendorf.

YeastTracker is a software to follow single cells in movies and to quantify fluorescent images based on this tracking. It has been developed for yeast cells, but is also applicable to other cells that have a defined round shape. The software is written in Matlab and uses a circular Hough transform and binary integer programming to detect and follow cells. It allows to quantify the mean fluorescence of each cell as well as the co-localization of two different fluorescent markers. The software is available on request (jannis.uhlendorf@inria.fr).

5.5. Rules2CP

Participants: François Fages, Raphaël Martin.

Rules2CP is a rule-based modeling language for constraint programming. It is distributed since 2009 as open-source. Unlike other modeling languages for constraint programming, Rules2CP adopts a single knowledge representation paradigm based on rules without recursion, and a restricted set of data structures based on records and enumerated lists given with iterators. This allows us to model complex constraint satisfaction problems together with search strategies, where search trees are expressed by logical formulae and heuristic choice criteria are defined with preference orderings by pattern-matching on the rules’ left-hand sides.

The expressiveness of Rules2CP has been illustrated in the FP6 Strep project Net-WMS by a complete library for packing problems, called PKML (Packing Knowledge Modeling Library), which, in addition to pure bin packing and bin design problems, can deal with common sense rules about weights, stability, as well as specific packing business rules.

5.6. SiLCC

Participant: Thierry Martinez.

SiLCC is an extensible modular concurrent constraint programming language relying upon linear logic. It is a complete implementation of the Linear logic Concurrent Constraint programming paradigm of Saraswat and Lincoln using the formal semantics of Fages, Ruet and Soliman. It is a single-paradigm logical language, enjoying concurrency, imperative traits, and a clean module system allowing to develop hierarchies of constraint systems within the language.

This software prototype is used to study the design of hierarchies of extensible libraries of constraint solvers. SiLCC is also considered as a possible implementation language for restructuring the code of BIOCHAM.

5.7. EMoP

Participant: Thierry Martinez.

EMoP is an extension of Prolog with first-class modules. These modules have the formal semantics of the LCC modules and provide Prolog with notions of namespaces, closures and objects within a simple programming model. Modules are also the support for user-definition of macros and modular syntax extensions. EMoP is bootstrapped and uses the GNU Prolog compilation chain as back-end.

5.8. CHRat

Participant: Thierry Martinez.

CHRat is a modular version of the well known Constraint Handling Rules language CHR, called for CHRat for CHR with ask and tell. Inspired by the LCC framework, this extension of CHR makes it possible to reuse CHRat components both in rules and guards in other CHRat components, and define hierarchies of constraint solvers. CHRat is a bootstrapped preprocessor for CHR which generates code for SWI/Prolog.
5.9. CLPGUI

Participant: François Fages.

CLPGUI is a generic graphical user interface written in Java for constraint logic programming. It is available for GNU-Prolog and SICStus Prolog. CLPGUI has been developed both for teaching purposes and for debugging complex programs. The graphical user interface is composed of several windows: one main console and several dynamic 2D and 3D viewers of the search tree and of finite domain variables. With CLPGUI it is possible to execute incrementally any goal, backtrack or recompute any state represented as a node in the search tree. The level of granularity for displaying the search tree is defined by annotations in the CLP program.

CLPGUI has been mainly developed in 2001 and is distributed as third-party software on GNU-Prolog and SICStus Prolog web sites. In 2009, CLPGUI has been interfaced to Rules2CP/PKML and used in the FP6 Strep Net-WMS with a non-released version.
4. Software

4.1. Gaspard 2

Participants: Jean-Luc Dekeyser [correspondant], All DaRT team.

Gaspard2 is an Integrated Development Environment (IDE) for SoC visual co-modeling. It allows or will allow modeling, simulation, testing and code generation of SoC applications and hardware architectures. Its purpose is to provide a single environment for all the SoC development processes:

- High level modeling of applications and hardware architectures
- Application and hardware architecture association (mapping and scheduling)
- Application refactoring
- Deployment specification
- Model to model transformation (to automatically produce models for several target platforms)
- Code generation
- Simulation
- Reification of any stages of the development

The Gaspard2 tool is based on the Eclipse IDE. A set of plugins provides the different functionalities. Gaspard2 provides an internal engine to execute transformation chains. This engine is able to run either QVT (OMG standard) or Java transformations. It is also able to run model-to-text transformations based on Acceleo. The Gaspard2 engine is defined to execute models conform to an internal transformation chains metamodel. A GUI has been developed to specify transformation chain models by drawing them. For the final user, application, hardware architecture, association, deployment and technology models are specified and manipulated by the developer through UML diagrams, and saved by the UML tool in an XMI file format. Gaspard2 manipulates these models through repositories (Java interfaces and implementations) automatically generated thanks to the Ecore specification. Several transformation chains are provided with Gaspard2 to target, from UML models, several execution or simulation platforms (OpenMP, OpenCL, Pthread, SystemC, VHDL, ...). This input language is based on the MARTE UML profile. A tool to generate SIMD configurations derived from the mppSoC model was developed. It allows to automatically generate the VHDL code from a high specification modeled at a high abstraction level (UML model using MARTE profile) based on the IP mppSoC library. The developed tool facilitates to the user to choose a SIMD configuration adapted to his application needs. It has been integrated in the Gaspard environment. **Gaspard2 as an educational resource.**

The Gaspard2 platform was one of the topics taught in the context of the courses on embedded systems in Telecom Lille and in a Master 2 (TNSI) lecture "Design tools for embedded systems" at the University of Valenciennes. These lectures focused on the potentiality to generate several targets from a subset of the Marte profile and the ability to target system on chip architectures at the TLM level respectively. Furthermore, the model driven engineering characteristics of Gaspard2 are largely detailed in the lecture of Software engineering at Polytech Lille and in the Master of research at university of Lille too.

- See also the web page [http://www.gaspard2.org/](http://www.gaspard2.org/)
- Inria software evaluation: A-2, SO-4, SM-2, EM-1, SDL-2, DA-4, CD-4, MS-4, TPM4
- Version: 2.1.0

4.2. Papyrus

Participants: Cédric Dumoulin [correspondant], Amine El Kouhen, Rahma Yangui.
The Papyrus tool is an UML Development Environment fully compliant with the UML standard and providing all UML diagrams. It is now an Eclipse project (in the incubator state). Papyrus Eclipse can easily be installed in Eclipse from the Eclipse update site. The Papyrus Tool is developed under an Open source license in collaboration with CEA, Atos, Airbus, LIFL.

- See also the web page [http://www.eclipse.org/papyrus/](http://www.eclipse.org/papyrus/)
- Software data: plugins number > 150, lines number > 1 million
- Inria software evaluation: A-5, SO-4, SM-4, EM-4, SDL-5, DA-4, CD-4, MS-4, TPM3
- Version: 0.9.0

4.3. Model Driven Factory

**Participants:** Alexis Muller, Anne Etien [correspondant], Thomas Legrand.

MDFactory is a Model Driven Engineering environment to design, develop and run software production chains. This tool supports our approach based on localized transformation and our Extend operator [96]. It provides a graphical editor to build such production chains with drag and drop from a reusable transformation library. MDFactory is based on the Eclipse platform and the Eclipse Modeling Framework (EMF). It is used to build Gaspard2 integrated transformation chains. This software will be transferred to the start up company Axellience.

- Software data: plugins number around 75
- Evaluation of the software: A 4; SO 4; SM 2; EM 3; SDL 3; DA 4; CD 3; MS 2; TPM 2
- Version: 1.0

4.4. OMEGSI

**Participant:** Amen Souissi [correspondant].

OMEGSI is an integrated development environment (IDE) for collaborative portals. It allows business process-centered modeling, process simulation, process optimization and full code generation for collaborative portals. The OMEGSI tool is based on the Eclipse IDE. A set of plugins provides the different functionalities. OMEGSI provides an internal engine to execute interactives transformation strategies. This engine (TranS) is written in QVT transformation and able to run any transformation type (QVT, Java, Acceleo...). Currently one transformation strategy is provided with OMEGSI to target, from an UML model, the Dolmen execution platform. This input language is based on the MACoP (Modeling and Analysis of Collaborative Portal) UML profile. The fully functional OMEGSI Beginning version still available on Ecreall website.

- See also the web page [http://omegsi.ecreall.com/](http://omegsi.ecreall.com/)
- Inria software evaluation: A-3, SO-3, SM-1, EM-2, SDL-4, DA-4, CD-4, MS-4, TPM4
- Version:
4. Software

4.1. The Polychrony toolset

Participants: Loïc Besnard, Thierry Gautier, Paul Le Guernic.

The Polychrony toolset is an Open Source development environment for critical/embedded systems. It is based on Signal, a real-time polychronous data-flow language. It provides a unified model-driven environment to perform design exploration by using top-down and bottom-up design methodologies formally supported by design model transformations from specification to implementation and from synchrony to asynchrony. It can be included in heterogeneous design systems with various input formalisms and output languages.

The Polychrony toolset provides a formal framework:

- to validate a design at different levels, by the way of formal verification and/or simulation,
- to refine descriptions in a top-down approach,
- to abstract properties needed for black-box composition,
- to assemble heterogeneous predefined components (bottom-up with COTS),
- to generate executable code for various architectures.

The Polychrony toolset contains three main components and an experimental interface to GNU Compiler Collection (GCC):

- The Signal toolbox, a batch compiler for the Signal language, and a structured API that provides a set of program transformations. The Signal toolbox can be installed without the other components. The Signal toolbox is distributed under GPL V2 license.
- The Signal GUI, a Graphical User Interface to the Signal toolbox (editor + interactive access to compiling functionalities). The Signal GUI is distributed under GPL V2 license.
- The SME platform, a front-end to the Signal toolbox in the Eclipse environment. The SME platform is distributed under EPL license.
- GCCst, a back-end to GCC that generates Signal programs (not yet available for download).

The Polychrony toolset also provides:

- libraries of Signal programs,
- a set of Signal program examples,
- user oriented and implementation documentations,
- facilities to generate new versions.

The Polychrony toolset can be freely downloaded on the following web sites:

- The Polychrony toolset public web site: [http://www.irisa.fr/espresso/Polychrony](http://www.irisa.fr/espresso/Polychrony). This site, intended for users and for developers, contains downloadable executable and source versions of the software for different platforms, user documentation, examples, libraries, scientific publications and implementation documentation. In particular, this is the site for the new open-source distribution of Polychrony.
- The INRIAGForge: [https://gforge.inria.fr](https://gforge.inria.fr). This site, intended for internal developers, contains the whole sources of the environment and their documentation.
- The TOPCASED distribution site: [http://www.topcased.org](http://www.topcased.org). This site provides the current reference version of the SME platform, including the executable of the Signal toolbox.
The Polychrony toolset currently runs on Linux, MacOS and Windows systems. The Geensoft company, now part of Dassault Systèmes, supplies a commercial implementation of Polychrony, called RT-Builder, used for industrial scale projects (see www.geensoft.com).

4.2. The Eclipse interface

Participants: Loïc Besnard, Yann Glouché, Huafeng Yu, François Fabre, Yue Ma.

We have developed a meta-model and interactive editor of Polychrony in Eclipse. Signal-Meta is the meta-model of the Signal language implemented with Eclipse/eCore. It describes all syntactic elements specified in [35]: all Signal operators (e.g. arithmetic, clock synchronization), model (e.g. process frame, module), and construction (e.g. iteration, type declaration).

The meta-model primarily aims at making the language and services of the Polychrony environment available to inter-operation and composition with other components (e.g. AADL, Simulink, GeneAuto) within an Eclipse-based development tool-chain. Polychrony now comprises the capability to directly import and export eCore models instead of textual Signal programs, in order to facilitate interaction between components within such a tool-chain.

![Eclipse SME Environment](image)

Figure 7. Eclipse SME Environment.

It also provides a graphical modeling framework allowing to design applications using a component-based approach. Application architectures can be easily described by just selecting components via drag and drop, creating some connections between them and specifying their parameters as component attributes. Using the modeling facilities provided with the Topcased framework, we have created a graphical environment for Polychrony (see figure 7) called SME (Signal-Meta under Eclipse). To highlight the different parts of the modeling in Signal, we split the modeling of a Signal process in three diagrams: one to model the interface of the process, one to model the computation (or dataflow) part, and one to model all explicit clock relations and dependences. The SME environment is available through the Espresso update site [27], in the current OpenEmbeDD distribution [26], or in the TopCased distribution [28].

4.3. Integrated Modular Avionics design using Polychrony

Participants: Thierry Gautier, Paul Le Guernic, Jean-Pierre Talpin.
The Apex interface, defined in the ARINC standard \[ 29 \], provides an avionics application software with the set of basic services to access the operating-system and other system-specific resources. Its definition relies on the Integrated Modular Avionics approach (IMA \[ 30 \]). A main feature in an IMA architecture is that several avionics applications (possibly with different critical levels) can be hosted on a single, shared computer system. Of course, a critical issue is to ensure safe allocation of shared computer resources in order to prevent fault propagations from one hosted application to another. This is addressed through a functional partitioning of the applications with respect to available time and memory resources. The allocation unit that results from this decomposition is the partition.

A partition is composed of processes which represent the executive units (an ARINC partition/process is akin to a Unix process/task). When a partition is activated, its owned processes run concurrently to perform the functions associated with the partition. The process scheduling policy is priority preemptive. Each partition is allocated to a processor for a fixed time window within a major time frame maintained by the operating system. Suitable mechanisms and devices are provided for communication and synchronization between processes (e.g. buffer, event, semaphore) and partitions (e.g. ports and channels). The specification of the ARINC 651-653 services in Signal \[ 7 \] is now part of the Polychrony distribution and offers a complete implementation of the Apex communication, synchronization, process management and partitioning services. Its Signal implementation consists of a library of generic, parameterizable Signal modules.

### 4.4. Multi-clocked mode automata

**Participants:** Jean-Pierre Talpin, Thierry Gautier, Christian Brunette.

Gathering advantages of declarative and imperative approaches, mode automata were originally proposed by Maraninchi et al. to extend the functionality-oriented data-flow paradigm with the capability to model transition systems easily and provide an additional imperative flavor. Similar variants and extensions of the same approach to mix multiple programming paradigms or heterogeneous models of computation \[ 36 \] have been proposed until recently, the latest advance being the combination of stream functions with automata in \[ 38 \]. Nowadays, commercial toolsets such as the Esterel Studio’s Scade or Matlab/Simulink’s Stateflow are largely inspired from similar concepts.

While the introduction of preemption mechanism in the multi-clocked data-flow formalism Signal was previously studied by Rutten et al. in \[ 51 \], no attempt has been made to extend mode automata with the capability to model multi-clocked systems and multi-rate systems. In \[ 53 \], we extend Signal-Meta with an inherited meta-model of multi-clocked mode automata. A salient feature is the simplicity incurred by the separation of concerns between data-flow (that expresses structure) and control-flow (that expresses a timing model) that is characteristic to the design methodology of Signal.

While the specification of mode automata in related works requires a primary address on the semantics and on compilation of control, the use of Signal as a foundation allows to waive this specific issue to its analysis and code generation engine Polychrony and clearly exposes the semantics and transformation of mode automata in a much simpler way by making use of clearly separated concerns expressed by guarded commands (data-flow relations) and by clock equations (control-flow relations).

### 4.5. Hyper-text source documentation of Polychrony

**Participants:** Loïc Besnard, Thierry Gautier, Paul Le Guernic.

As part of its open-source release, the Polychrony toolset not only comprises source code libraries but also an important corpus of structured documentation, whose aim is not only to document each functionality and service, but also to help a potential developer to package a subset of these functionalities and services and adapt them to developing a new application-specific tool: a new language front-end, a new back-end compiler. This multi-scale, multi-purpose documentation aims to provide different views of the software, from a high-level structural view to low-level descriptions of basic modules. It supports a distribution of the software “by apartment” (a functionality or a set of functionalities) intended for developers who would only be interested by part of the services of the toolset.

A high-level architectural view of the Polychrony toolset is given in Figure 8.
Figure 8. The Polychrony toolset high-level architecture
5. Software

5.1. aCiNO

Participants: Fei He [correspondant], Min Zhou.

aCiNO is an SMT (Satisfiability Modulo Theory) solver based on a Nelson-Oppen [65] architecture, and written in C++. Currently, two popular theories are considered: linear real arithmetic (LRA) and uninterpreted functions (UF). A lazy approach is used for solving SMT problem. For efficiency consideration, the solver is implemented in an incremental way. It also invokes an online SAT solver, which is now a modified MiniSAT, so that recovery from conflict is possible.

5.2. CoLoR and Rainbow

Participants: Frédéric Blanqui [correspondant], Kim-Quyen Ly, Sidi Ould Biha.

CoLoR is a Coq [44] library on rewriting theory and termination of nearly 70,000 lines of code [11]. It provides definitions and theorems for:

- Mathematical structures: relations, (ordered) semi-rings.
- Data structures: lists, vectors, polynomials with multiple variables, finite multisets, matrices.
- Term structures: strings, algebraic terms with symbols of fixed arity, algebraic terms with varyadic symbols, simply typed lambda-terms.
- Transformation techniques: conversion from strings to algebraic terms, conversion from algebraic to varyadic terms, arguments filtering, rule elimination, dependency pairs, dependency graph decomposition, semantic labelling.
- Termination criteria: polynomial interpretations, multiset ordering, lexicographic ordering, first and higher order recursive path ordering, matrix interpretations.

Rainbow is a tool for automatically certifying termination certificates expressed in the CPF XML format [29] used in the termination competition on termination [32]. Termination certificates are translated and checked in Coq by using the CoLoR library.

CoLoR and Rainbow are distributed under the CeCILL license on http://color.inria.fr/ . Various people participated to its development (see the website for more information).

5.3. EDOLA

Participants: Hehua Zhang [correspondant], Ming Gu, Hui Kong, Yu Jiang.

Joint work with Jiaguang Sun (Tsinghua University, China).

EDOLA [26] is an integrated tool for domain-specific modeling and verification of PLC applications [74]. It is based on a domain-specific modeling language to describe system models. It supports both model checking and automatic theorem proving techniques for verification. The goal of this tool is to possess both the usability in domain modeling, the reusability in its architecture and the capability of automatic verification.

For the moment, we have developed a prototype of the EDOLA language, which can easily describe the features of PLC applications like the scan cycle mechanism, the pattern of environment model, time constraints and five property patterns. TLA+ [59] was chosen as the intermediate language to implement the automatic verification of EDOLA models. A prototype of EDOLA has also been developed, which comes along with an editor to help writing EDOLA models. To automatically verify properties on EDOLA models, it provides the interface for both a model checker TLC [59] and a first-order theorem prover SPASS [75].
5.4. Moca

**Participant:** Frédéric Blanqui [correspondant].

Joint work with Pierre Weis (INRIA Rocquencourt) and Richard Bonichon (CEA).

Moca is a construction functions generator for OCaml data types with invariants. It allows the high-level definition and automatic management of complex invariants for data types. In addition, it provides the automatic generation of maximally shared values, independently or in conjunction with the declared invariants.

A relational data type is a concrete data type that declares invariants or relations that are verified by its constructors. For each relational data type definition, Moca compiles a set of construction functions that implements the declared relations.

Moca supports two kinds of relations:

- predefined algebraic relations (such as associativity or commutativity of a binary constructor),
- user-defined rewrite rules that map some pattern of constructors and variables to some arbitrary user's define expression.

The properties that user-defined rules should satisfy (completeness, termination, and confluence of the resulting term rewriting system) must be verified by a programmer's proof before compilation. For the predefined relations, Moca generates construction functions that allow each equivalence class to be uniquely represented by their canonical value.

Moca is distributed under QPL on http://moca.inria.fr/.

5.5. SimSoC

**Participant:** Vania Joloboff [correspondant].

SimSoC is an infrastructure to run simulation models which comes along with a library of simulation models. SimSoC allows its users to experiment various system architectures, study hardware/software partition, and develop embedded software in a co-design environment before the hardware is ready to be used. SimSoC aims at providing high performance, yet accurate simulation, and provide tools to evaluate performance and functional or non functional properties of the simulated system.

SimSoC is based on SystemC standard and uses Transaction Level Modeling for interactions between the simulation models. The current version of SimSoC is based on the open source libraries from the OSCI Consortium: SystemC version 2.2 and TLM 2.0.1 [54], [33]. Hardware components are modeled as TLM models, and since TLM is itself based on SystemC, the simulation is driven by the SystemC kernel. We use standard, unmodified, SystemC (version 2.2), hence the simulator has a single simulation loop.

The second open source version of SimSoC, SimSoC v0.7.1, has been released in November 2010. It contains a full simulator for ARM V5 and PowerPC both running at an average speed of about 80 Millions instructions per second, and a simulator for the MIPS architecture with an average speed of 20 Mips in mode DT1. It represents about 70,000 lines of source code and includes:

- Instruction Set Simulators. The ARM Version 5 architecture has been implemented with DT0, DT1, DT2 mode. The ARM and PowerPC 600 architecture with DT0 and DT1 mode. For both architectures, complete simulation models of the processor and MMU are provided, making it possible to run operating systems of the simulated platform. MIPS architecture in DT0 mode is under development.

- A dynamic translator from binary programs to an internal representation. For the ARM architecture a compiler has been developed that generates the C++ translated code (for DT2), using parametrized specialization options.
• Peripheral models including a serial line controller, a flash memory controller, an interrupt controller.
• A utility to generate permanent storage for flash memory simulation; a compiler tool to generate instruction binary decoder.
• Examples illustrating the use of the library and infrastructure.

SimSoC is distributed under LGPL on https://gforge.inria.fr/projects/simsoc.

5.6. SimSoC-Cert

Participants: Frédéric Blanqui, Vania Joloboff, Jean-François Monin [correspondant], Xiaomu Shi.

SimSoC-Cert is a set of tools that can automatically generate in various target languages (Coq and C) the decoding functions and the state transition functions of each instruction and addressing mode of the ARMv6 architecture manual [28] (implemented by the ARM11 processor family) but the Thumb and coprocessor instructions. The input of SimSoC-Cert is the ARMv6 architecture manual itself.

Based on this, we first developed simlight (8000 generated lines of C, plus 1500 hand-written lines of C), a simulator for ARMv6 programs using no peripheral and no coprocessor. Next, we developed simlight2, a fast ARMv6 simulator integrated inside a SystemC/TLM module, now part of SimSoC v0.7.

We can also generate similar programs for SH4 [31] but this is still under test.
5. Software

5.1. Mathemagix, a free computer algebra environment

Participants: Bernard Mourrain, Angelos Mantzaflaris.

http://www.mathemagix.org/

MATHEMAGIX is a free computer algebra system which consists of a general purpose interpreter, which can be used for non-mathematical tasks as well, and efficient modules on algebraic objects. It includes the development of standard libraries for basic arithmetic on dense and sparse objects (numbers, univariate and multivariate polynomials, power series, matrices, etc., based on FFT and other fast algorithms). These developments are based on C++, offer generic programming without losing effectiveness, via the parameterization of the code (template) and the control of their instantiations.

The language of the interpreter is imperative, strongly typed and high level. A compiler of this language is available. A special effort has been put on the embedding of existing libraries written in other languages like C or C++. An interesting feature is that this extension mechanism supports template types, which automatically induce generic types inside Mathemagix. Connections with GMP, MPFR for extended arithmetic, LAPACK for numerical linear algebra are currently available in this framework.

The project aims at building a bridge between symbolic computation and numerical analysis. It is structured by collaborative software developments of different groups in the domain of algebraic and symbolic-numeric computation.

In this framework, we are working more specifically on the following components:

- **REALROOT**: a set of solvers using subdivision methods to isolate the roots of polynomial equations in one or several variables; continued fraction expansion of roots of univariate polynomials; Bernstein basis representation of univariate and multivariate polynomials and related algorithms; exact computation with real algebraic numbers, sign evaluation, comparison, certified numerical approximation.
- **SHAPE**: tools to manipulate curves and surfaces of different types including parameterised, implicit with different type of coefficients; algorithms to compute their topology, intersection points or curves, self-intersection locus, singularities, ...

These packages are integrated from the former library SYNAPS (SYmbolic Numeric APplicationS) dedicated to symbolic and numerical computations. There are also used in the algebraic-geometric modeler AXEL.

Collaborators: Grégoire Lecerf, Joris van der Hoeven and Philippe Trébuchet.

5.2. Axel, a geometric modeler for algebraic objects

Participants: Angelos Mantzaflaris, Bernard Mourrain, Meriadeg Perrinel.

http://axel.inria.fr.

We are developing a software called AXEL (Algebraic Software-Components for gEometric modeLing) dedicated to algebraic methods for curves and surfaces. Many algorithms in geometric modeling require a combination of geometric and algebraic tools. Aiming at the development of reliable and efficient implementations, AXEL provides a framework for such combination of tools, involving symbolic and numeric computations.
The software contains data structures and functionalities related to algebraic models used in geometric modeling, such as polynomial parameterisation, B-Spline, implicit curves and surfaces. It provides algorithms for the treatment of such geometric objects, such as tools for computing intersection points of curves or surfaces, detecting and computing self-intersection points of parameterized surfaces, implicitization, for computing the topology of implicit curves, for meshing implicit (singular) surfaces, etc.

The developments related to isogeometric analysis in Exciting have been integrated as dedicated plugins. Optimisation techniques and solvers for partial differential equations developed by R. Duvigneau (OPALE) have been connected.

A new version of the algebraic-geometric modelers is developed by Meriadeg Perinnel to connect it to the platform Dtk in order to provide a better modularity and a better interface to existing computation facilities and geometric rendering interface.

The package is distributed as binary packages for Linux as well as for MacOSX. It is hosted at the INRIA’s gforge (http://gforge.inria.fr) and referenced by many leading software websites such as http://apple.com. The first version of the software has been downloaded more than 15000 times, since it is available. Collaboration with Gang Xu (Hangzhou Dianzi University, China), Julien Wintz (Dream).

5.3. Maple packages for differential algebra and algebraic invariants

Participant: Evelyne Hubert.

- The Maple package `diffalg` is a collection of routines to handle systems of polynomial differential equations and inequations. The functionalities include differential elimination, expansion of the solutions into formal power series and analysis of singular solutions. The underlying theory and terminology belongs to differential algebra.
  Collaborators: François Boulier and François Lemaire from University of Lille.

- The Maple `AIDA` package is a collection of routines to explore algebra of differential invariants: computation of generating sets of invariants, rewritings, syzygies, and their differential analogues. The package builds on the Maple libraries Groebner, Vessiot and diffalg. 
GALLIUM Project-Team

5. Software

5.1. OCaml

**Participants:** Xavier Leroy [correspondant], Xavier Clerc [team SED], Damien Doligez, Alain Frisch [LexiFi], Jacques Garrigue [Nagoya University], Maxence Guersdon [team SED], Luc Maranget [EPI Moscova], Michel Mauny [ENSTA], Nicolas Pouillard, Pierre Weis [EPI Estime].

OCaml, formerly known as Objective Caml, is our flagship implementation of the Caml language. From a language standpoint, it extends the core Caml language with a fully-fledged object and class layer, as well as a powerful module system, all joined together by a sound, polymorphic type system featuring type inference. The OCaml system is an industrial-strength implementation of this language, featuring a high-performance native-code compiler for several processor architectures (IA32, AMD64, PowerPC, ARM, etc) as well as a bytecode compiler and interactive loop for quick development and portability. The OCaml distribution includes a standard library and a number of programming tools: replay debugger, lexer and parser generators, documentation generator, compilation manager, and the Camlp4 source pre-processor.


5.2. CompCert C

**Participants:** Xavier Leroy [correspondant], Sandrine Blazy [EPI Celtique], Alexandre Pilkiewicz.

The CompCert C verified compiler is a compiler for a large subset of the C programming language that generates code for the PowerPC, ARM and x86 processors. The distinguishing feature of CompCert is that it has been formally verified using the Coq proof assistant: the generated assembly code is formally guaranteed to behave as prescribed by the semantics of the source C code. The subset of C supported is quite large, including all C types except `long long` and `long double`, all C operators, almost all control structures (the only exception is unstructured `switch`), and the full power of functions (including function pointers and recursive functions but not variadic functions). The generated PowerPC code runs 2–3 times faster than that generated by GCC without optimizations, and only 7% (resp. 12%) slower than GCC at optimization level 1 (resp. 2).


5.3. Zenon

**Participant:** Damien Doligez.

Zenon is an automatic theorem prover based on the tableaux method. Given a first-order statement as input, it outputs a fully formal proof in the form of a Coq proof script. It has special rules for efficient handling of equality and arbitrary transitive relations. Although still in the prototype stage, it already gives satisfying results on standard automatic-proving benchmarks.

Zenon is designed to be easy to interface with front-end tools (for example integration in an interactive proof assistant), and also to be easily retargetted to output scripts for different frameworks (for example, Isabelle).


5.4. Menhir

**Participants:** François Pottier [correspondant], Yann Régis-Gianas [U. Paris Diderot].

Menhir is a new LR(1) parser generator for Objective Caml. Menhir improves on its predecessor, `ocamlyacc`, in many ways: more expressive language of grammars, including EBNF syntax and the ability to parameterize a non-terminal by other symbols; support for full LR(1) parsing, not just LALR(1); ability to explain conflicts in terms of the grammar.

5. Software

5.1. CGAL, the Computational Geometry Algorithms Library

Participants: Pierre Alliez, Jean-Daniel Boissonnat, Olivier Devillers, Monique Teillaud, Mariette Yvinec.

With the collaboration of Hervé Brönnimann, Manuel Caroli, Pedro Machado Manhães de Castro, Frédéric Cazals, Frank Da, Christophe Delage, Andreas Fabri, Julia Flötotto, Philippe Guigue, Michael Hemmer, Samuel Hornus, Menelaos Karavelas, Sébastien Loriot, Abdelkrim Mebarki, Naceur Meskini, Andreas Meyer, Sylvain Pion, Marc Pouget, François Rebufat, Laurent Rineau, Laurent Saboret, Stéphane Tayeb, Radu Ursu, and Camille Wormser. http://www.cgal.org

CGAL is a C++ library of geometric algorithms and data structures. Its development has been initially funded and further supported by several European projects (CGAL, GALIA, ECG, ACS, AIM@SHAPE) since 1996. The long term partners of the project are research teams from the following institutes: INRIA Sophia Antipolis - Méditerranée, Max-Planck Institut Saarbrücken, ETH Zürich, Tel Aviv University, together with several others. In 2003, CGAL became an Open Source project (under the LGPL and QPL licenses), and it also became commercialized by GEOMETRY FACTORY, a company Born of INRIA founded by Andreas Fabri.

The aim of the CGAL project is to create a platform for geometric computing supporting usage in both industry and academia. The main design goals are genericity, numerical robustness, efficiency and ease of use. These goals are enforced by a review of all submissions managed by an editorial board. As the focus is on fundamental geometric algorithms and data structures, the target application domains are numerous: from geological modeling to medical images, from antenna placement to geographic information systems, etc.

The CGAL library consists of a kernel, a list of algorithmic packages, and a support library. The kernel is made of classes that represent elementary geometric objects (points, vectors, lines, segments, planes, simplices, isothetic boxes, circles, spheres, circular arcs...), as well as affine transformations and a number of predicates and geometric constructions over these objects. These classes exist in dimensions 2 and 3 (static dimension) and d (dynamic dimension). Using the template mechanism, each class can be instantiated following several representation modes: one can choose between Cartesian or homogeneous coordinates, use different types to store the coordinates, and use reference counting or not. The kernel also provides some robustness features using some specifically-devised arithmetic (interval arithmetic, multi-precision arithmetic, static filters...).

A number of packages provide geometric data structures as well as algorithms. The data structures are polygons, polyhedra, triangulations, planar maps, arrangements and various search structures (segment trees, d-dimensional trees...). Algorithms are provided to compute convex hulls, Voronoi diagrams, Boolean operations on polygons, solve certain optimization problems (linear, quadratic, generalized of linear type). Through class and function templates, these algorithms can be used either with the kernel objects or with user-defined geometric classes provided they match a documented interface.

Finally, the support library provides random generators, and interfacing code with other libraries, tools, or file formats (ASCII files, QT or LEDA Windows, OpenGL, Open Inventor, Postscript, Geomview...). Partial interfaces with Python, SCILAB and the Ipe drawing editor are now also available.

GEOMETRICA is particularly involved in general maintenance, in the arithmetic issues that arise in the treatment of robustness issues, in the kernel, in triangulation packages and their close applications such as alpha shapes, in meshes... Three researchers of GEOMETRICA are members of the CGAL Editorial Board, whose main responsibilities are the control of the quality of CGAL, making decisions about technical matters, coordinating communication and promotion of CGAL.

CGAL is about 700,000 lines of code and supports various platforms: GCC (Linux, Mac OS X, Cygwin...), Visual C++ (Windows), Intel C++... A new version of CGAL is released twice a year, and it is downloaded about 10000 times a year. Moreover, CGAL is directly available as packages for the Debian, Ubuntu and Fedora Linux distributions.
More numbers about CGAL: there are now 13 editors in the editorial board, with approximately 20 additional developers. The user discussion mailing-list has more than 1000 subscribers with a relatively high traffic of 5-10 mails a day. The announcement mailing-list has more than 3000 subscribers.
5. Software

5.1. Pari/Gp

Participants: Karim Belabas [correspondant], Bill Allombert, Henri Cohen, Andreas Enge.

http://pari.math.u-bordeaux.fr/

PARI/GP is a widely used computer algebra system designed for fast computations in number theory (factorisation, algebraic number theory, elliptic curves, ...), but it also contains a large number of other useful functions to compute with mathematical entities such as matrices, polynomials, power series, algebraic numbers, etc., and many transcendental functions.

- PARI is a C library, allowing fast computations.
- GP is an easy-to-use interactive shell giving access to the PARI functions.
- gp2c, the GP-to-C compiler, combines the best of both worlds by compiling GP scripts to the C language and transparently loading the resulting functions into GP; scripts compiled by gp2c will typically run three to four times faster.

2011 has seen the release of the next major stable version, 2.5, ending the 2.3 release series started in 2007.

- Version of PARI/GP: 2.5.0
- Version of gp2c: 0.0.7pl11
- License: GPL v2+
- Programming language: C

5.2. MPC

Participants: Andreas Enge [correspondant], Mickaël Gastineau, Philippe Théveny, Paul Zimmermann [INRIA project-team CARAMEL].

http://mpc.multiprecision.org/

MPC is a C library for the arithmetic of complex numbers with arbitrarily high precision and correct rounding of the result. It is built upon and follows the same principles as MPFR.

It is a prerequisite for the GNU compiler collection GCC since version 4.5, where it is used in the C and Fortran frontends for constant folding, the evaluation of constant mathematical expressions during the compilation of a program. Since 2011, it is an official GNU project.

- Version: 0.9 *Epilobium montanum*
- License: LGPL v2.1+
- ACM: G.1.0 (Multiple precision arithmetic)
- AMS: 30.04 Explicit machine computation and programs
- APP: Dépôt APP le 2003-02-05 sous le numéro IDDN FR 001 060029 000 R P 2003 000 10000
- Programming language: C

5.3. MPFRCX

Participant: Andreas Enge.

http://mpfrcx.multiprecision.org/
MPFRCX is a library for the arithmetic of univariate polynomials over arbitrary precision real (MPFR) or complex (MPC) numbers, without control on the rounding. For the time being, only the few functions needed to implement the floating point approach to complex multiplication are implemented. On the other hand, these comprise asymptotically fast multiplication routines such as Toom-Cook and the FFT.

- Version: 0.3.1 Banane
- License: LGPL v2.1+
- Programming language: C

5.4. CM

Participant: Andreas Enge.

http://cm.multiprecision.org/

The CM software implements the construction of ring class fields of imaginary quadratic number fields and of elliptic curves with complex multiplication via floating point approximations. It consists of libraries that can be called from within a C program and of executable command line applications. For the implemented algorithms, see [9].

- Version: 0.1 Apfelkraut
- License: LGPL v2+
- Programming language: C

5.5. AVIsogenies

Participants: Damien Robert [correspondant], Gaëtan Bisson, Romain Cosset [INRIA project-team CARAMEL].

http://avisogenies.gforge.inria.fr/.

AVISOGENIES (Abelian Varieties and Isogenies) is a MAGMA package for working with abelian varieties, with a particular emphasis on explicit isogeny computation.

Its prominent feature is the computation of $(\ell, \ell)$-isogenies between Jacobian varieties of genus-two hyper-elliptic curves over finite fields of characteristic coprime to $\ell$; practical runs have used values of $\ell$ in the hundreds.

It can also be used to compute endomorphism rings of abelian surfaces, and find complete addition laws on them.

- Version: 0.4
- License: LGPL v2.1+
- Programming language: Magma

5.6. Cubic

Participant: Karim Belabas.

http://www.math.u-bordeaux1.fr/~belabas/research/software/cubic-1.2.tgz

CUBIC is a standalone program that prints out generating equations for cubic fields of either signature and bounded discriminant. It depends on the PARI library. The algorithm has quasi-linear time complexity in the size of the output.

- Version: 1.2
- License: GPL v2+
- Programming language: C
LICIT Exploratory Action (section vide)
5. Software

5.1. Semantics

Participant: Yves Bertot [correspondant].

This is a library for the Coq system, where the description of a toy programming language is presented. The value of this library is that it can be re-used in classrooms to teach programming language semantics or the Coq system. The topics covered include introductory notions to domain theory, pre and post-conditions, abstract interpretation, and the proofs of consistency between all these point of views on the same programming language. Standalone tools for the object programming language can be derived from this development. See also the web page http://coq.inria.fr/pylons/pylons/contribs/view/Semantics/v8.3.

- ACM: F3.2 F4.1
- AMS: 68N30
- Programming language: Coq

5.2. Certicrypt

Participants: Gilles Barthe [IMDEA Software institute], Juan Manuel Crespo [IMDEA Software institute], Benjamin Grégoire [correspondant], Sylvain Heraud, César Kunz [IMDEA Software institute], Federico Olmedo [IMDEA Software institute], Santiago Zanella Béguelin [IMDEA Software institute].

CertiCrypt takes a language-based approach to cryptography: the security of a cryptographic scheme and the cryptographic assumptions upon which its security relies are expressed by means of probabilistic programs, called games; in a similar way, adversarial models are specified in terms of complexity classes, e.g. probabilistic polynomial-time programs. This code-centric view leads to statements that are amenable to formalization and tool-assisted verification. CertiCrypt instruments a rich set of verification techniques for probabilistic programs, including equational theories of observational equivalence, relational Hoare logic, data-flow analysis-based program transformations, and game-based techniques such as eager/lazy sampling and failure events. See also the web page http://easycrypt.gforge.inria.fr/.
MEXICO Project-Team

5. Software

5.1. Software

5.1.1. libalf: the Automata Learning Framework

Participant: Benedikt Bollig [correspondant].

libalf is a comprehensive, open-source library for learning finite-state automata covering various well-known learning techniques (such as, Angluin’s L*, Biermann, and RPNI, as well as a novel learning algorithm for NFA). libalf is highly flexible and allows for facilely interchanging learning algorithms and combining domain-specific features in a plug-and-play fashion. Its modular design avirtual plantsnd its implementation in C++ make it a flexible platform for adding and engineering further, efficient learning algorithms for new target models (e.g., Büchi automata).

Details on libalf can be found at http://libalf.informatik.rwth-aachen.de/

5.1.2. Mole/Cunf: unfolders for Petri Nets

Participants: Stefan Schwoon [correspondant], César Rodríguez.

Mole computes, given a safe Petri net, a finite prefix of its unfolding. It is designed to be compatible with other tools, such as PEP and the Model-Checking Kit, which are using the resulting unfolding for reachability checking and other analyses. The tool Mole arose out of earlier work on Petri nets. Details on Mole can be found at http://www.lsv.ens-cachan.fr/~schwoon/tools/mole/

In the context of MExICo, we have created a new tool called Cunf, which is able to handle contextual nets (Petri nets with read arcs). Recent work carried out within MExICo [53] has transformed a preliminary implementation into an efficient tool. While in principle every contextual net can be transformed into an “equivalent” Petri net and then unfolded using Mole, Cunf can take advantage of their special features to do the job faster. More details can be found at http://www.lsv.ens-cachan.fr/~rodrigue/tools/cunf/

5.1.3. COSMOS: a Statistical Model Checker for the Hybrid Automata Stochastic Logic

Participants: Hilal Djafri [correspondant], Benoît Barbot.

COSMOS is a statistical model checker for the Hybrid Automata Stochastic Logic (HASL). HASL employs Linear Hybrid Automata (LHA), a generalization of Deterministic Timed Automata (DTA), to describe accepting execution paths of a Discrete Event Stochastic Process (DESP), a class of stochastic models which includes, but is not limited to, Markov chains. As a result HASL verification turns out to be a unifying framework where sophisticated temporal reasoning is naturally blended with elaborate reward-based analysis. COSMOS takes as input a DESP (described in terms of a Generalized Stochastic Petri Net), an LHA and an expression Z representing the quantity to be estimated. It returns a confidence interval estimation of Z. COSMOS is written in C++ and is freely available to the research community.

Details on COSMOS can be found at http://www.lsv.ens-cachan.fr/~barbot/cosmos/
5. Software

5.1. F7: Refinement Types for F#

Participants: Karthikeyan Bhargavan [correspondant], Cédric Fournet [MSR Cambridge], Andrew D. Gordon [MSR Cambridge].

F7 is an enhanced typechecker for the F# programming language that enables static checking of properties expressed as refinement types.

A refinement type is a base type qualified with a logical formula; the formula can express invariants, preconditions, and postconditions. F7 relies on type annotations, including refinements, provided in specific interface files. While checking code, F7 generates many logical problems which it solves by submitting to Z3, an external theorem prover for first-order logic (de Moura and Bjørner 2008). Finally, F7 erases all refinements and yields ordinary F# modules and interfaces.

Our main aim is to use F7 for the verification of security-critical programs. We have used it to verify implementations of access control mechanisms, multi-party secure sessions, cryptographic protocols for web services security and federated authentication, and secure audit logs.

A first version of F7 was released in 2008. In 2011, we revised the F7 libraries and typechecker and ported it to the released version of F# for .NET 4.0. The second version of F7 was released in December 2011.

The typechecker is written in 16000 lines of F#, with an additional cryptographic library of 9000 lines, and sample code of more than 12000 lines.

5.2. JSTY: Logical Auditing of JavaScript Programs

Participants: Karthikeyan Bhargavan [correspondant], Sergio Maffeis [Imperial College], Ravinder Shankesi [U. of Illinois at Urbana Champain].

JSTY is a runtime monitoring and logical auditing framework for JavaScript web applications. It has three components: (1) a contract language for JavaScript that enables programmers to annotate their scripts with assumptions and goals written as first-order logic pre- and post-conditions; (2) a runtime monitor implemented as a browser extension in the web browser Chrome that interprets these contracts at runtime and generates proof obligations for an SMT solver; (3) a logical auditor that checks proof obligations and maps counterexamples to violations of program correctness goals.

The target applications for JSTY include browser extensions as well as website scripts. In the case of browser extensions, our goal is to help extension writers to test their code by annotating it with logical contracts and auditing the code with JSTY. For website scripts, our goal is to check whether a website script obeys a generic security policy expressed as pre-conditions on functions in the browser or DOM API. We have used JSTY to analyze a variety of security-critical browser extensions and website scripts and found several vulnerabilities. We are currently incorporating static checking into JSTY.

JSTY is written in about 1000 lines of JavaScript and we plan a public release in 2012.

5.3. OTT: Tool support for the working semanticist

Participants: Peter Sewell [U. of Cambridge], Francesco Zappa Nardelli [correspondant].
Ott is a tool for writing definitions of programming languages and calculi. It takes as input a definition of a language syntax and semantics, in a concise and readable ASCII notation that is close to what one would write in informal mathematics. It generates output:

1. a LaTeX source file that defines commands to build a typeset version of the definition;
2. a Coq version of the definition;
3. an Isabelle version of the definition; and
4. a HOL version of the definition.

Additionally, it can be run as a filter, taking a LaTeX/Coq/Isabelle/HOL source file with embedded (symbolic) terms of the defined language, parsing them and replacing them by typeset terms.

The main goal of the Ott tool is to support work on large programming language definitions, where the scale makes it hard to keep a definition internally consistent, and to keep a tight correspondence between a definition and implementations. We also wish to ease rapid prototyping work with smaller calculi, and to make it easier to exchange definitions and definition fragments between groups. The theorem-prover backends should enable a smooth transition between use of informal and formal mathematics.

In collaboration with Peter Sewell (Cambridge University).

The current version of Ott is about 30000 lines of OCaml. The tool is available from http://moscova.inria.fr/~zappa/software/ott (BSD licence).

Since its release in December 2007, the tool has been used in several projects, including a large proof of type preservation for the OCaml language (without modules) done by Scott Owens.

In 2011, apart from minor bug-fixes and features added, we implemented several performance improvements which result in a up-to 6x speed-up, and kept the Isabelle and Coq backend up-to date with the theorem prover evolution.

The currently released version is 0.21.1.

5.4. Lem, a tool for lightweight executable mathematics

Participants: Scott Owens [U. of Cambridge], Peter Sewell [U. of Cambridge], Francesco Zappa Nardelli [correspondent].

Lem is a lightweight tool for writing, managing, and publishing large scale semantic definitions. It is also intended as an intermediate language for generating definitions from domain-specific tools, and for porting definitions between interactive theorem proving systems (such as Coq, HOL4, and Isabelle). As such it is a complementary tool to Ott.

Lem resembles a pure subset of Objective Caml, supporting typical functional programming constructs, including top-level parametric polymorphism, datatypes, records, higher-order functions, and pattern matching. It also supports common logical mechanisms including list and set comprehensions, universal and existential quantifiers, and inductively defined relations. From this, Lem generates OCaml, HOL4 and Isabelle code; the OCaml backend uses a finite set library (and does not yet support inductive relations). A Coq backend is in development.

Lem is already in use at Cambridge and INRIA for research on relaxed-memory concurrency. We are currently preparing a feature-complete release with back-ends for HOL4, Isabelle/HOL, Coq, OCaml, and LaTeX. The project web-page is http://www.cl.cam.ac.uk/~so294/lem/. A paper on a Lem prototype appeared in ITP 2011, in the “rough diamond” category [ 25 ].

5.5. Memevents-Litmus-Diy-Dont

Participants: Jade Alglave, Luc Maranget [correspondent], Susmit Sarkar [U. of Cambridge, UK], Peter Sewell [U. of Cambridge, UK].
Luc Maranget is the main developer of the tools suite of project “Weak Memory Models” (cf. the relevant section).

This suite features three subtools memevents (model checker), litmus (runs tests on actual machines) and diy (generate tests from concise specifications). This year saw a new tool and one official releases (with documentation) [33] — see also http://diy.inria.fr. The releases feature all tools except memevents, which we wish to keep for ourselves.

This year main extensions are the handling of the ARM architecture and more collaboration between tools. For the latter, the test generator diy enriches tests with meta-data that are exploited by litmus, so as to perform binding of test threads to machine processors, intelligent prefetch of data etc. We plan a new release of the tool suite early next year.

A new, independent, “proof of concept” tool, offence was written by J. Alglave and L. Maranget, as a support of our publication [30].

This software is available at http://diy.inria.fr/offence.

5.6. Jocaml

Participants: Luc Maranget, Xavier Clerc [correspondant].

Jocaml is an implementation of the join-calculus integrated into Ocaml. With respect to previous join-language prototypes, the most salient feature of the new prototype is a better integration into Ocaml. We achieve binary compatibility with Ocaml, moreover Jocaml releases now follow Ocaml releases. See previous year reports for details on Jocaml. The current version is 3.12.1 (released in September [34]) is available at http://jocaml.inria.fr/.

This new release features an extended Jocaml specific library that provide programmers with an easier access to concurrency and distribution:

1. Some utilities to parse command line, organize client-server connection, etc. This code was written partly by Xavier Clerc, engineer at INRIA SED department.
2. Some new abstractions of text channels help for writing text oriented applications.

5.7. Hevea

Participant: Luc Maranget [correspondant].

Hevea is a fast translator from full LaTeX to HTML, written in Ocaml. Hevea is highly configurable with commands written in LaTeX. Mathematics are rendered with UNICODE characters for symbols and HTML tables for formatting. Hevea produces HTML 4.0, enriched by css files. Hevea comes with Hacha companion, which produces a set of HTML pages (for instance, one page per chapter). Since it is very efficient and configurable, Hevea is adequate for on-line manuals or teaching courses.

This year saw a few developments around Hevea, mostly for maintenance. Hevea is available at http://hevea.inria.fr/.
5. Software

5.1. ATerm

**Participant:** Pierre-Etienne Moreau [correspondant].

ATerm (short for Annotated Term) is an abstract data type designed for the exchange of tree-like data structures between distributed applications.

The ATerm library forms a comprehensive procedural interface which enables creation and manipulation of ATerms in C and Java. The ATerm implementation is based on maximal subterm sharing and automatic garbage collection.

A binary exchange format for the concise representation of ATerms (sharing preserved) allows the fast exchange of ATerms between applications. In a typical application—parse trees which contain considerable redundant information—less than 2 bytes are needed to represent a node in memory, and less than 2 bits are needed to represent it in binary format. The implementation of ATerms scales up to the manipulation of ATerms in the giga-byte range.

The ATerm library provides a comprehensive interface in C and Java to handle the annotated term data-type in an efficient manner.

We are involved (with the CWI) in the implementation of the Java version, as well as in the garbage collector of the C version. The Java version of the ATerm library is used in particular by Tom.

The ATerm library is documented, maintained, and available at the following address: http://www.meta-environment.org/Meta-Environment/ATerms.

5.2. Tom

**Participants:** Jean-Christophe Bach, Horatiu Cirstea, Pierre-Etienne Moreau [correspondant], Claudia Tavares.

Since 2002, we have developed a new system called Tom [49], presented in [27], [28]. This system consists of a pattern matching compiler which is particularly well-suited for programming various transformations on trees/terms and XML documents. Its design follows our experiences on the efficient compilation of rule-based systems [45]. The main originality of this system is to be language and data-structure independent. This means that the Tom technology can be used in a C, C++ or Java environment. The tool can be seen as a Yacc-like compiler translating patterns into executable pattern matching automata. Similarly to Yacc, when a match is found, the corresponding semantic action (a sequence of instructions written in the chosen underlying language) is triggered and executed. Tom supports sophisticated matching theories such as associative matching with neutral element (also known as list-matching). This kind of matching theory is particularly well-suited to perform list or XML based transformations for example.

In addition to the notion of rule, Tom offers a sophisticated way of controlling their application: a strategy language. Based on a clear semantics, this language allows to define classical traversal strategies such a innermost, outermost, etc.. Moreover, Tom provides an extension of pattern matching, called anti-pattern matching. This corresponds to a natural way to specify complements (i.e. what should not be there to fire a rule). Tom also supports the definition of cyclic graph data-structures, as well as matching algorithm and rewriting rules for term-graphs.

5.3. Cat

Participant: Yves Guiraud [correspondant].

Cat is a library for polygraphic calculus, written in Caml. It has been used, in a joint work with F. Blanqui, to produce an automatic termination prover for first-order functional programs. It translates such a rewriting system into a polygraph and tries to find a derivation proving its termination, using the results of [6], [39]. If possible, it seeks a derivation that proves that the program is polynomial [29], [3]. Cat is also at the basis of Catex.

5.4. Catex

Participant: Yves Guiraud [correspondant].

Catex is a tool for (pdf)Latex, used in the same way as Bibtex, that automatically produces string diagrams from their algebraic expression. It follows the same design as Tom, a Catex file being a Latex file enriched with formal islands corresponding to those algebraic expressions, such as:

```
deftwocell[red]{delta : 1 -> 2}
deftwocell[orange]{mu : 2 -> 1}
deftwocell[crossing]{tau : 2 -> 2}
twocell{(delta *0 delta) *1 (1 *0 tau *0 1) *1 (mu *0 mu)}
```

Catex dissolves such an island into Latex code, using the PGF/Tikz package. Executed on the result, (pdf)Latex produces the following diagram:

![Figure 4](image)

Catex is distributed through the page: http://www.loria.fr/~guiraudy/catex. We want to extend Catex in two directions. First, to produce diagrams not only for Latex but also for web publications. Then, Catex will be adapted to a tool for the automatic production, in scientific papers, of certified algebraic computations, which are a three-dimensional equivalent of string diagrams.
5. Software

5.1. Lucid Synchrone

Participant: Marc Pouzet [contact].

Lucid Synchrone is a language for the implementation of reactive systems. It is based on the synchronous model of time as provided by Lustre combined with features from ML languages. It provides powerful extensions such as type and clock inference, type-based causality and initialization analysis and allows to arbitrarily mix data-flow systems and hierarchical automata or flows and valued signals.

It is distributed under binary form, at URL http://www.di.ens.fr/~pouzet/lucid-synchrone/ .

The language was used, from 1996 to 2006 as a laboratory to experiment various extensions of the language Lustre. Several programming constructs (e.g. merge, last, mix of data-flow and control-structures like automata), type-based program analysis (e.g., typing, clock calculus) and compilation methods, originally introduced in Lucid Synchrone are now integrated in the new SCADE 6 compiler developed at Esterel Technologies and commercialized since 2008.

Three major releases of the language have been done and the current version is V3 (dev. in 2006). The language is still used for teaching and in our research but we do not develop it anymore. Nonetheless, we have integrated several features from Lucid Synchrone in new research prototypes described below.

5.2. ReactiveML

Participants: Mehdi Dogguy, Louis Mandel [contact], Cédric Pasteur.

ReactiveML is a programming language dedicated to the implementation of interactive systems as found in graphical user interfaces, video games or simulation problems. ReactiveML is based on the synchronous reactive model due to Boussinot, embedded in an ML language (Objective Caml).

The Synchronous reactive model provides synchronous parallel composition and dynamic features like the dynamic creation of processes. In ReactiveML, the reactive model is integrated at the language level (not as a library) which leads to a safer and a more natural programming paradigm.

ReactiveML is distributed at URL http://www.lri.fr/~mandel/rml . The compiler is distributed under the terms of the Q Public License and the library is distributed under the terms of the GNU Library General Public License. The development of ReactiveML started at the University Paris 6 (from 2002 to 2006).

The language was mainly used for the simulation of mobile ad hoc networks at the University Paris 6 and for the simulation of sensor networks at France Telecom and Verimag (CNRS, Grenoble).

5.3. Heptagon

Participants: Cédric Pasteur [contact], Brice Gelineau, Léonard Gérard, Adrien Guatto, Cédric Pasteur, Marc Pouzet.

Heptagon is an experimental language for the implementation of embedded real-time reactive systems. It is developed inside the Synchronics large-scale initiative, in collaboration with INRIA Rhones-Alpes. It is essentially a subset of Lucid Synchrone, without type inference, type polymorphism and higher-order. It is thus a Lustre-like language extended with hierarchical automata in a form very close to SCADE 6. The intention for making this new language and compiler is to develop new aggressive optimization techniques for sequential C code and compilation methods for generating parallel code for different platforms. This explains much of the simplifications we have made in order to ease the development of compilation techniques.
Some extensions have already been made, most notably automata. It’s currently used to experiment with linear typing for arrays and also to introduce a concept of asynchronous parallel computations. The compiler developed in our team generates C, java and VHDL code.

Heptagon is jointly developed by Gwenael Delaval and Alain Girault from the INRIA POP ART team (Grenoble).

5.4. Lucy-n

Participants: Louis Mandel [contact], Adrien Guatto, Marc Pouzet.

http://www.lri.fr/~mandel/lucy-n

Lucy-n is a language to program in the n-synchronous model. The language is similar to Lustre with a buffer construct. The Lucy-n compiler ensures that programs can be executed in bounded memory and automatically computes buffer sizes. Hence this language allows to program Kahn networks, the compiler being able to statically compute bounds for all FIFOs in the program.

5.5. ML-Sundials

Participants: Timothy Bourke, Marc Pouzet [contact].

ML-Sundials library provides an Ocaml interface to the Sundials numerical suite 10 (version 2.4.0). This library is used for solving and initial value problem and includes a zero-crossing detection mechanism. Only the CVODE solver with serial nvectors is currently supported. The structure and naming conventions largely follow the original libraries, both for ease of reading the existing documentation and for converting existing source code, but several changes have been made for programming convenience, namely:

- solver sessions are configured through algebraic data types rather than through multiple function calls,
- error conditions are signalled by exceptions rather than return codes (including in user-supplied callback routines),
- closures (partial applications of higher-order functions) are used to share user data between callback routines, and,
- explicit free commands are not necessary nor provided since Ocaml is a garbage-collected language.

The library is in use in a new synchronous hybrid language we are currently developing.

5.6. GCC

Participants: Albert Cohen [contact], Tobias Grosser, Antoniu Pop, Konrad Trifunovic, Feng Li, Riyadh Baghdadi, Cupertino Miranda.

http://gcc.gnu.org

Licence: GPLv3+ and LGPLv3+

The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgcj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100% free software, free in the sense that it respects the user’s freedom.

PARKAS contributes to the polyhedral compilation framework, also known as Graphite. We also distribute an experimental branch for a stream-programming extension of OpenMP, parallel data-flow programming, and automatic parallelization to a data-flow runtime or architecture. This experiment borrows key design elements to synchronous data-flow languages.

10 https://computation.llnl.gov/casc/sundials/main.html
Tobias Grosser is the maintainer of the Graphite optimization pass of GCC.

5.7. isl

Participants: Sven Verdoolaege [contact], Tobias Grosser, Albert Cohen.

http://freshmeat.net/projects/isl

Licence: LGPLv2.1+

isl is a library for manipulating sets and relations of integer points bounded by linear constraints. Supported operations on sets include intersection, union, set difference, emptiness check, convex hull, (integer) affine hull, integer projection, transitive closure (and over-approximation), computing the lexicographic minimum using parametric integer programming. It also includes an ILP solver based on generalized basis reduction. isl also supports affine transformations for polyhedral compilation.
5. Software

5.1. Profound

Participants: Kaustuv Chaudhuri [correspondant], Nicolas Guenot, Lutz Straßburger.

Profound is an interactive proof-development tool based on the focused calculus of structures [19]. It allows the user to build proofs using direct manipulation of the current proof state using the cursor keys and the mouse, instead of learning a formal textual proof interaction language. The tool checks all user actions dynamically with the aid of a theorem prover.

We plan to investigate adaptations of a tool such as Profound for proof development in other interactive proof development systems such as Abella or Coq. We also plan to use the high degree of proof compression that is enabled by the calculus of structures to create efficient proof certificates for exchange between different proof development systems.

The first public release of Profound is expected in December 2011. The development can be followed on INRIA GForge.

5.2. Abella

Participants: Andrew Gacek, Dale Miller.

The earliest versions of the Abella theorem prover was written while Gacek was a PhD student at the University of Minnesota. Two years ago, Gacek was a post doc in the Parsifal team and more features were added to this prover. During 2011, Chaudhuri and three interns (Andrew Cave from McGill and Salil Joshi and Chris Martens from CMU) developed some news designs and new prototypes of needed features for Abella. These features will provide this prover with better ways to manipulate specification-logic contexts in the reasoning-logic and a means for outputting proper proof object (different from the scripts that are used to find a proof).

For more information, see the Abella home page: http://abella.cs.umn.edu/.

5.3. Bedwyr

Participants: Kaustuv Chaudhuri, Quentin Heath, Dale Miller [correspondant].

During 2011, our close colleagues Alwen Tui (Australian National University) and David Baelde (INRIA team Proval) have made some improvements to the Bedwyr system. In the case of Tui, he made these changes in order to build SPEC, a model checker for the spi-calculus, on top of Bedwyr.

Starting in September, Quentin Heath has joined the team as a technical staff member. He is currently working on the Bedwyr code so that it can share files with the Abella system. These two provers work within a logic that is essentially the same: Heath is working to ensure that the concrete syntax and static semantics for the logical expressions on which they work is also the same. Thus, we expect to have our model checker (Bedwyr) and interactive theorem prover (Abella) share theories and proofs.

The work of Heath is being done in the context of the BATT ADJ project funded by INRIA. The boarder goals of the BATT project is to get four software systems (Bedwyr, Abella, Tac, and Teyjus) to inter-operate.

See also the web page http://slimmer.gforge.inria.fr/bedwyr/.
5. Software


Participants: Bruno Barras [TypiCal team, Saclay], Yves Bertot [Marelle team, Sophia], Frédéric Besson [Lande team, Rennes], Pierre Boutillier, Xavier Clerc [SED team], Pierre Corbineau [University Joseph Fourier, Grenoble], Pierre Courtieu [CNAM], Julien Forest [CNAM], Stéphane Glondu, Benjamin Grégoire [Marelle team, Sophia], Vincent Gross, Hugo Herbelin [correspondent], Stéphane Lescuyer [ProVal team, Saclay], Pierre Letouzey, Assia Mahboubi [TypiCal team, Saclay], Julien Narboux [University of Strasbourg], Jean-Marc Notin [TypiCal team, Saclay], Christine Paulin [Proval team, Saclay], Loïc Pottier [Marelle team, Sophia], Matthias Puech, Yann Régis-Gianas, Vincent Siles, Elie Soubiran, Matthieu Sozeau, Arnaud Spiwack, Pierre-Yves Strub [Formes team, Beijing], Laurent Théry [Marelle team, Sophia], Benjamin Werner [TypiCal team, Saclay].

5.1.1. Version 8.4

Version 8.4 beta was released in December 2011. It introduces a new proof engine designed and implemented by Arnaud Spiwack and a new extensive modular library of arithmetic contributed by Pierre Letouzey. It also includes an extension of the underlying logic with \( \eta \)-conversion by Hugo Herbelin and “commutative-cuts compliant guard condition” by Pierre Boutillier, an extension of the pattern-matching compilation algorithm by Hugo Herbelin, an extension of the procedure of simplication of polynomial expressions by Loïc Pottier, a refinement of the type classes mechanism by Matthieu Sozeau, a new communication model by Vincent Gross for the graphical user interface CoqIDE, that Pierre Letouzey, Pierre Boutillier and Pierre-Marie Pédrot further extended.

Several users gracefully contributed improvements of various features (Tom Prince, Enrico Tassi, Daniel Grayson, Hendrik Tews, ...).

5.1.2. Graphical user interface

Pierre Letouzey has finalized the work initiated by Vincent Gross (former ADT engineer) concerning the CoqIDE user interface: CoqIDE and Coq are now separate unix processes, enhancing the reliability and improving the user experience.

5.1.3. Type inference, tactics, unification and type classes

Matthieu Sozeau corrected important issues with the unification algorithm and enhanced it to support universes. He improved the type-class implementation, adding support for forward-reasoning instances. To improve the power of induction tactics, Hugo Herbelin added new heuristics for second-order pattern-matching based on ideas from Chung-Kil Hur’s \( \text{Ieq} \) plugin.

Pierre Letouzey extended the pattern-matching feature of the tactic language.

5.1.4. Internal architecture of the Coq software

Pierre Letouzey also initiated a large reorganization of the internal components of Coq, since these components are currently too much interdependent. This work aims at better isolating components and explicating the interfaces between them. In addition to the initial goal of simplifying the compilation of Coq, having a clearer architecture is also expected to help new contributors when they discover and interact with this large and complex code-base. It also brings new prospects in direct communications between tools developed around Coq. This is a long-term effort that extends beyond the Coq 8.4 release.
Pierre Boutillier worked on the build system generator for Coq users contributions. It now handles correctly developments involving ML files. Files to build developments are also used by CoqIDE to infer the required arguments when it opens a file of a development.

5.1.5. Efficiency

Pierre Letouzey has pursued his effort concerning the improvement of many aspects of the internals of Coq. In particular, with Yann Régis-Gianas, he enabled a faster load of libraries by default, thanks to laziness, and also a better sharing of structures in memory (via better hash-consing), with lower memory footprint and some speedup as visible result. Many bugs have also been addressed.

Starting from September, Xavier Clerc has worked on the codebase in order to profile typical executions. Some hotspots were identified, most notably in comparison functions: some minor modifications led to a gain of a few percents on average. Some tests led to envision the use of a Coq-specialized version of comparison functions, superseding the generic OCaml ones.

5.1.6. General maintenance

Hugo Herbelin, Pierre Letouzey, Pierre Boutillier, Stéphane Glondu and Matthieu Sozeau worked on the general maintenance of the system.

5.1.7. Development Action

A new “Action de Développement Technologique” about Coq has started September 2011. It gathers the πr² team, the Marelle team and the CPR team from CNAM, Hugo Herbelin acting as the coordinator. It supports visits and meetings between developers and aims at strengthening the community of Coq users and contributors.

5.1.8. Formalisation in Coq

Stéphane Glondu is working with Mehdi Dogguy on the formalisation in Coq of a type system for a timed asynchronous π-calculus that guarantees confluence.

5.2. Pangolin

Participant: Yann Régis-Gianas.

Yann Régis-Gianas maintained a prototype version of Pangolin. He used it to prove concrete complexity bounds for a set of functional programs using the method described in his FOPARA 2011 paper [ 16 ].

5.3. Other software developments

Stéphane Glondu is involved in the maintenance of OCaml-related packages in Debian, which include OCaml itself, Coq, Ssreflect (an extension of Coq developed at INRIA-MSR joint center) and Ocsigen (a web framework developed at PPS). The Ubuntu distribution naturally benefits from this work.

In collaboration with François Pottier (INRIA Gallium), Yann Régis-Gianas maintained Menhir, an LR parser generator for OCaml.
5. Software

5.1. NBac

Participant: Bertrand Jeannet.

NBAC (Numerical and Boolean Automaton Checker) is a verification/slicing tool for reactive systems containing combination of Boolean and numerical variables, and continuously interacting with an external environment. NBAC can also handle the same class of hybrid systems as the HyTech tool. It aims at handling efficiently systems combining a non-trivial numerical behaviour with a complex logical (Boolean) behaviour.

NBAC is connected to two input languages: the synchronous dataflow language LUSTRE, and a symbolic automaton-based language, AUTOC/AUTO, where a system is defined by a set of symbolic hybrid automata communicating via valued channels. It can perform reachability analysis, co-reachability analysis, and combination of the above analyses. The result of an analysis is either a verdict to a verification problem, or a set of states together with a necessary condition to stay in this set during an execution. NBAC is founded on the theory of abstract interpretation.

It has been used for verification and debugging of LUSTRE programs. It is connected to the LUSTRE toolset. It has also been used for controller synthesis of infinite-state systems. The fact that the analyses are approximated results simply in the obtention of a possibly non-optimal controller. In the context of conformance testing of reactive systems, it is used by the test generator STG for selecting test cases.

5.2. Prometheus

Participant: Gregor Goessler.

The BIP component model (Behavior, Interaction model, Priority) has been designed to support the construction of heterogeneous embedded systems involving different models of computation, communication, and execution, at different levels of abstraction. By separating the notions of behavior, interaction model, and execution model, it enables both heterogeneous modeling, and separation of concerns.

The verification and design tool Prometheus implements the BIP component framework. Prometheus is regularly updated to implement new developments in the framework and the analysis algorithms. It has allowed us to carry out several complex case studies from the system-on-chip and bioinformatics domains.

5.3. Implementations of Synchronous Programs

Participant: Alain Girault.

13 http://pop-art.inrialpes.fr/people/bjeannet/nbac/
14 http://www-verimag.imag.fr/Lustre-V6.html
5.3.1. Fault Tolerance

We have been cooperating for several years with the INRIA team AOSTE (INRIA Sophia-Antipolis and Rocquencourt) on the topic of fault tolerance and reliability of safety critical embedded systems. In particular, we have implemented several new heuristics for fault tolerance and reliability within their software SYNDEX\textsuperscript{16}. Our first scheduling heuristic produces static multiprocessor schedules tolerant to a specified number of processor and communication link failures [64]. The basic principles upon which we rely to make the schedules fault tolerant is, on the one hand, the active replication of the operations [65], and on the other hand, the active replication of communications for point-to-point communication links, or their passive replication coupled with data fragmentation for multi-point communication media (i.e., buses) [66]. Our second scheduling heuristic is multi-criteria: it produces a static schedule multiprocessor schedule such that the reliability is maximized, the power consumption is minimized, and the execution time is minimized [3][17]. Our results on fault tolerance are summarized in a web page\textsuperscript{17}.

5.4. Apron and BddApron Libraries

\textbf{Participant:} Bertrand Jeannet.

5.4.1. Principles

The APRON library\textsuperscript{18} is dedicated to the static analysis of the numerical variables of a program by abstract interpretation [51]. Many abstract domains have been designed and implemented for analysing the possible values of numerical variables during the execution of a program (see Figure 1). However, their API diverge largely (datatypes, signatures, ...), and this does not ease their diffusion and experimental comparison w.r.t. efficiency and precision aspects.

The APRON library aims to provide:

- a uniform API for existing numerical abstract domains;
- a higher-level interface to the client tools, by factorizing functionalities that are largely independent of abstract domains.

From an abstract domain designer point of view, the benefits of the APRON library are:

- the ability to focus on core, low-level functionalities;
- the help of generic services adding higher-level services for free.

For the client static analysis community, the benefits are a unified, higher-level interface, which allows experimenting, comparing, and combining abstract domains.

In 2011, the Taylor1plus domain [62], which is the underlying abstract domain of the tool FLUCTUAT [58] has been improved. Glue code has also been added to enable the connection of an abstract domain implemented in OCaml to the APRON infrastructure written in C (this requires callbacks from C to OCaml that are safe w.r.t. garbage collection). This will enable the integration in APRON of the MaxPlus polyhedra library written by X. Allamigeon [38] in the context of the ANR ASSOPT project.

The BDDAPRON library\textsuperscript{19} aims at a similar goal, by adding finite-types variables and expressions to the concrete semantics of APRON domains. It is built upon the APRON library and provides abstract domains for the combination of finite-type variables (Booleans, enumerated types, bitvectors) and numerical variables (integers, rationals, floating-point numbers). It first allows to manipulate expressions that freely mix, using BDDs and MTBDDs, finite-type and numerical APRON expressions and conditions. It then provides abstract domains that combines BDDs and APRON abstract values for representing invariants holding on both finite-type variables and numerical variables.

\textsuperscript{16}http://www-rocinria.fr/syndex
\textsuperscript{17}http://pop-art.inrialpes.fr/~girault/Projets/FT
\textsuperscript{18}http://apron.cri.ensmp.fr/library/
\textsuperscript{19}http://pop-art.inrialpes.fr/~bjeannet/bjeannet-forge/bddapron/index.html
5.4.2. Implementation and Distribution

The APRON library (Fig. 2) is written in ANSI C, with an object-oriented and thread-safe design. Both multi-precision and floating-point numbers are supported. A wrapper for the OCAML language is available, and a C++ wrapper is on the way. It has been distributed since June 2006 under the LGPL license and available at http://apron.cri.ensmp.fr. Its development has still progressed much since. There are already many external users (ProVal/Démons, LRI Orsay, France — CEA-LIST, Saclay, France — Analysis of Computer Systems Group, New-York University, USA — Sierum software analysis platform, Kansas State University, USA — NEC Labs, Princeton, USA — EADS CCR, Paris, France — IRIT, Toulouse, France) and is currently packaged as a REDHAT and DEBIAN package.

The BDDAPRON library is written in OCAML, using polymorphism features of OCAML to make it generic. It is also thread-safe. It provides two different implementations of the same domain, each one presenting pros and cons depending on the application. It is currently used by the CONCURINTERPROC interprocedural and concurrent program analyzer.

5.5. Prototypes

5.5.1. Logical Causality

Participants: Lacramioara Astefanoaei, Gregor Goessler [contact person].

We have developed LoCA, a new prototype tool written in Scala that implements the analysis of logical causality described in 6.6.2. LoCA currently supports causality analysis in BIP. The core analysis engine is implemented as an abstract class, such that support for other models of computation (MOC) can be added by instantiating the class with the basic operations of the MOC.

5.5.2. Automatic Controller Generation

Participants: Emil Dumitrescu, Alain Girault [contact person].

We have developed a software tool chain to allow the specification of models, the controller synthesis, and the execution or simulation of the results. It is based on existing synchronous tools, and thus consists primarily in the use and integration of SIGALI\(^{20}\) and Mode Automata\(^{21}\). It is the result of a collaboration with Eric Rutten from the SARDES team.

---

\(^{20}\) [http://www.irisa.fr/vertecs/Logiciels/sigali.html]

\(^{21}\) [http://www-verimag.imag.fr]
Useful component templates and relevant properties can be materialized, on one hand by libraries of task models, and, on the other hand, by properties and synthesis objectives.

5.5.3. Rapture

Participant: Bertrand Jeannet.

**RAPTURE**\(^{22}\) [78], [53] is a verification tool that was developed jointly by BRICS (Denmark) and INRIA in years 2000–2002. The tool is designed to verify reachability properties on Markov Decision Processes (MDP), also known as Probabilistic Transition Systems. This model can be viewed both as an extension to classical (finite-state) transition systems extended with probability distributions on successor states, or as an extension of Markov Chains with non-determinism. We have developed a simple automata language that allows the designer to describe a set of processes communicating over a set of channels à la CSP. Processes can also manipulate local and global variables of finite type. Probabilistic reachability properties are specified by defining two sets of initial and final states together with a probability bound. The originality of the tool is to provide two reduction techniques that limit the state space explosion problem: automatic abstraction and refinement algorithms, and the so-called essential states reduction.

5.5.4. The Interproc family of static analyzers

Participants: Bertrand Jeannet [contact person], Pascal Sotin.

These analyzers and libraries are of general use for people working in the static analysis and abstract interpretation community, and serve as an experimental platform for the ANR project ASOPT (see § 8.1.2).

> a generic fix-point engine written in OCAML. It allows the user to solve systems of fix-point equations on a lattice, using a parameterized strategy for the iteration order and the application of widening. It also implements recent techniques for improving the precision of analysis by alternating

\(^{22}\) http://pop-art.inrialpes.fr/people/bjeannet/rapture/rapture.html
post-fixpoint computation with widening and descending iterations in a sound way [ 70 ].

a simple interprocedural static analyzer that infers properties on the numerical variables of programs in a toy language. It is aimed at demonstrating the use of the previous library and the above-described APRON library, and more generally at disseminating the knowledge in abstract interpretation. It is also deployed through a web-interface 25. It is used as the experimental platform of the ASOPT ANR project.

**fixpointinterproc** extends Interproc with concurrency, for the analysis of multithreaded programs interacting via shared global variables. It is also deployed through a web-interface 26.

**pinterproc** extends Interproc with pointers to local variables. It is also deployed through a web-interface 27.

### 5.5.5. Heptagon/BZR

**Participant:** Gwenaël Delaval.

**Heptagon** is a dataflow synchronous language, inspired from **Lucid Synchrone**28. Its compiler is meant to be simple and modular, allowing this language to be a good support for the prototyping of compilation methods of synchronous languages. It is developed within the **SynchronicsINRIA** large-scale action.

**Heptagon** has been used to built **BZR** 29, which is an extension of the former with contracts constructs. These contracts allow to express dynamic temporal properties on the inputs and outputs of **Heptagon** node. These properties are then enforced, within the compilation of a **BZR** program, by discrete controller synthesis, using the **Sigali** tool 30. The synthesized controller is itself generated in **Heptagon**, allowing its analysis and compilation towards different target languages (C, **Java**, **VHDL**).
5. Software

5.1. The CiME rewrite toolbox

Participants: Évelyne Contejean [contact], Claude Marché, Andrei Paskevich, Xavier Urbain.

CiME is a rewriting toolbox. Distributed since 1996 as open source, at URL http://cime.lri.fr. Beyond a few dozens of users, CiME is used as back-end for other tools such as the TALP tool developed by Enno Ohlebusch at Bielefeld university for termination of logic programs; the MU-TERM tool (http://www.dsic.upv.es/~slucas/csr/termination/muterm/) for termination of context-sensitive rewriting; the CARIBOO tool (developed at INRIA Nancy Grand-Est) for termination of rewriting under strategies; and the MTT tool (http://www.lcc.uma.es/~duran/MTT/) for termination of Maude programs. CiME2 is no longer maintained, and the currently developed version is CiME3, available at http://a3pat.ensiie.fr/pub. The main new feature of CiME3 is the production of traces for Coq. CiME3 is also developed by the participants of the A3PAT project at the CNAM, and is distributed under the Cecill-C license.

5.2. The Why platform

Participants: Claude Marché [contact], Romain Bardou, François Bobot, Jean-Christophe Filliâtre, Guillaume Melquiond, Andrei Paskevich.


The Why platform is a set of tools for deductive verification of Java and C source code. In both cases, the requirements are specified as annotations in the source, in a special style of comments. For Java (and Java Card), these specifications are given in JML and are interpreted by the Krakatoa tool. Analysis of C code must be done using the external Frama-C environment, and its Jessie plugin which is distributed in Why.

The platform is distributed as open source, under GPL license, at http://why.lri.fr/. The internal VC generator and the translators to external provers are no longer under active development, as superseded by the Why3 system described below.

The Krakatoa and Jessie front-ends are still maintained, although using now by default the Why3 VC generator. These front-ends are described in a specific web page http://krakatoa.lri.fr/. They are used for teaching (University of Evry, Ecole Polytechnique, etc.), used by several research groups in the world, e.g at Fraunhofer Institute in Berlin [71], and at Universidade do Minho in Portugal [50].

5.3. The Why3 system

Participants: Jean-Christophe Filliâtre [contact], François Bobot, Claude Marché, Guillaume Melquiond, Andrei Paskevich.


Why3 is the next generation of Why. Why3 clearly separates the purely logical specification part from generation of verification conditions for programs. It features a rich library of proof task transformations that can be chained to produce a suitable input for a large set of theorem provers, including SMT solvers, TPTP provers, as well as interactive proof assistants.

It is distributed as open source, under GPL license, at http://why3.lri.fr/.

---

Why3 is used as back-end of our own tools Krakatoa and Jessie, but also as back-end of the GNATprove tool (Adacore company), and in a near future of the WP plugin of Frama-C. Why3 has been used to develop and prove a significant part of the programs of our team gallery http://proval.lri.fr/gallery/index.en.html, and will be used soon for teaching (Master Parisien de Recherche en Informatique).

5.4. The Alt-Ergo theorem prover

Participants: Sylvain Conchon [contact], Évelyne Contejean, Stéphane Lescuyer, Alain Mebsout, Mohamed Iguernelala.

Criteria for Software Self-Assessment: A-3-up, SO-4, SM-4-up, EM-4, SDL-5, OC-4.

Alt-Ergo is an automatic, little engine of proof dedicated to program verification, whose development started in 2006. It is fully integrated in the program verification tool chain developed in our team. It solves goals that are directly written in the Why’s annotation language; this means that Alt-Ergo fully supports first order polymorphic logic with quantifiers. Alt-Ergo also supports the standard [90] defined by the SMT-lib initiative. It is currently used in our team to prove correctness of C and Java programs as part of the Why platform and the new Why3 system. Alt-Ergo is also called as an external prover by the Pangolin tool developed by Y. Regis Gianas, INRIA project-team Gallium http://code.google.com/p/pangolin-programming-language/. Alt-Ergo is usable as a back-end prover in the SPARK verifier for ADA programs, since Oct 2010. It is planed to be integrated in next generation of Airbus development process.

Alt-Ergo is distributed as open source, under the CeCILL-C license, at URL http://alt-ergo.lri.fr/.

5.5. Bibtex2html

Participants: Jean-Christophe Filliâtre [contact], Claude Marché.


Bibtex2html is a generator of HTML pages of bibliographic references. Distributed as open source since 1997, under the GPL license, at http://www.lri.fr/~filliatr/bibtex2html/. We estimate that between 10000 and 100000 web pages have been generated using Bibtex2html.

Bibtex2html is also distributed as a package in most Linux distributions. Package popularity contests show that it is among the 20% most often installed packages.

5.6. OCamlgraph

Participants: Jean-Christophe Filliâtre [contact], Sylvain Conchon.

OCamlgraph is a graph library for Objective Caml. It features many graph data structures, together with many graph algorithms. Data structures and algorithms are provided independently of each other, thanks to OCaml module system. OCamlgraph is distributed as open source, under the LGPL license, at http://ocamlgraph.lri.fr/. It is also distributed as a package in several Linux distributions. OCamlgraph is now widely spread among the community of OCaml developers.

5.7. Mlpost

Participants: Jean-Christophe Filliâtre [contact], Stéphane Lescuyer, Romain Bardou, François Bobot.

Mlpost is a tool to draw scientific figures to be integrated in LaTeX documents. Contrary to other tools such as TikZ or MetaPost, it does not introduce a new programming language; it is instead designed as a library of an existing programming language, namely Objective Caml. Yet it is based on MetaPost internally and thus provides high-quality PostScript figures and powerful features such as intersection points or clipping. Mlpost is distributed as open source, under the LGPL license, at http://mlpost.lri.fr/. Mlpost was presented at JFLA’09 [51].
5.8. Functory

Participants: Jean-Christophe Filliâtre [contact], Kalyan Krishnamani.

Functory is a distributed computing library for Objective Caml. The main features of this library include (1) a polymorphic API, (2) several implementations to adapt to different deployment scenarios such as sequential, multi-core or network, and (3) a reliable fault-tolerance mechanism. Functory was presented at JFLA 2011 [31] and at TFP 2011 [27].

5.9. The Flocq library

Participants: Sylvie Boldo [contact], Guillaume Melquiond.


The Flocq library for the Coq proof assistant is a comprehensive formalization of floating-point arithmetic: core definitions, axiomatic and computational rounding operations, high-level properties [23]. It provides a framework for developers to formally certify numerical applications.

It is distributed as open source, under a LGPL license, at http://flocq.gforge.inria.fr/. It was first released in 2010.

5.10. The Gappa tool

Participant: Guillaume Melquiond [contact].


Given a logical property involving interval enclosures of mathematical expressions, Gappa tries to verify this property and generates a formal proof of its validity. This formal proof can be machine-checked by an independent tool like the Coq proof-checker, so as to reach a high level of confidence in the certification [66] [19].

Since these mathematical expressions can contain rounding operators in addition to usual arithmetic operators, Gappa is especially well suited to prove properties that arise when certifying a numerical application, be it floating-point or fixed-point. Gappa makes it easy to compute ranges of variables and bounds on absolute or relative roundoff errors.

Gappa is being used to certify parts of the mathematical libraries of several projects, including CRlibm, FLIP, and CGAL. It is distributed as open source, under a Cecill-B / GPL dual-license, at http://gappa.gforge.inria.fr/. Part of the work on this tool was done while in the Arénaire team (INRIA Rhône-Alpes), until 2008.

5.11. The Interval package for Coq

Participant: Guillaume Melquiond [contact].


The Interval package provides several tactics for helping a Coq user to prove theorems on enclosures of real-valued expressions. The proofs are performed by an interval kernel which relies on a computable formalization of floating-point arithmetic in Coq.

It is distributed as open source, under a LGPL license, at http://www.lri.fr/~melquion/soft/coq-interval/. Part of the work on this library was done while in the Mathematical Components team (Microsoft Research–INRIA Joint Research Center).

In 2010, the Flocq library was used to straighten and fill the floating-point proofs of the Interval package.

5.12. The Alea library for randomized algorithms

Participants: Christine Paulin-Mohring [contact], David Baelde.

The ALEA library is a Coq development for modeling randomized functional programs as distributions using a monadic transformation. It contains an axiomatisation of the real interval $[0, 1]$ and its extension to positive real numbers. It introduces definition of distributions and general rules for approximating the probability that a program satisfies a given property.

It is distributed as open source, at http://www.lri.fr/~paulin/ALEA . It is currently used as a basis of the Certicrypt environment (MSR-INRIA joint research center, Imdea Madrid, INRIA Sophia-Antipolis) for formal proofs for computational cryptography [53]. It is also experimented in LABRI as a basis to study formal proofs of probabilistic distributed algorithms.

5.13. The Coccinelle library for term rewriting

Participant: Évelyne Contejean [contact].

Coccinelle is a Coq library for term rewriting. Besides the usual definitions and theorems of term algebras, term rewriting and term ordering, it also models some of the algorithms implemented in the CiME toolbox, such as matching, matching modulo associativity-commutativity, computation of the one-step reducts of a term, RPO comparison between two terms, etc. The RPO algorithm can effectively be run inside Coq, and is used in the Color development (http://color.inria.fr/) as well as for certifying Spike implicit induction theorems in Coq (Sorin Stratulat).

Coccinelle is developed by Évelyne Contejean, available at (http://www.lri.fr/~contejea/Coccinelle), and is distributed under the Cecill-C license.
5. Software

5.1. Mica: A Modal Interface Compositional Analysis Toolbox

Participant: Benoît Caillaud.

http://www.irisa.fr/s4/tools/mica/

Mica is an Ocaml library developed by Benoît Caillaud implementing the Modal Interface algebra published in [18]. The purpose of Modal Interfaces is to provide a formal support to contract based design methods in the field of system engineering. Modal Interfaces enable compositional reasoning methods on I/O reactive systems.

In Mica, systems and interfaces are represented by extension. However, a careful design of the state and event heap enables the definition, composition and analysis of reasonably large systems and interfaces. The heap stores states and events in a hash table and ensures structural equality (there is no duplication). Therefore complex data-structures for states and events induce a very low overhead, as checking equality is done in constant time.

Thanks to the Inter module and the mica interactive environment, users can define complex systems and interfaces using Ocaml syntax. It is even possible to define parameterized components as Ocaml functions.

Mica is available as an open-source distribution, under the CeCILL-C Free Software License Agreement (http://www.cecill.info/licences/Licence_CeCILL-C_V1-en.html).

5.2. Synet: A General Petri-Net Synthesis Toolbox

Participant: Benoît Caillaud.

http://www.irisa.fr/s4/tools/synet/

Synet is a software tool for the synthesis of bounded and unbounded Petri-nets, based on the theory of regions [31]. It can synthesize Petri-nets from automata or regular expression and can be configured by command-line options to synthesize nets modulo graph isomorphism or language equality. Petri nets computed by Synet can be displayed using the GraphViz 2D graph layout software, or saved to a file for further transformation and analysis.

The tool actually implements two linear-algebraic synthesis methods: A first method uses the simplex algorithm and the second one is based on the computation of extremal rays of polyhedral cones, using Chernikova’s algorithm [34]. Both methods imply that the input graphs are given by extension. Nevertheless, Synet yields good performances on many practical use-cases and is the only tool supporting unbounded net synthesis.

The main application of Synet is the synthesis of communicating distributed protocols and controllers [30]. Synthesis is constrained to produce so-called distributables nets [33], a class of nets that can be turned into networks of communicating automata by automated methods. This allows to divide the synthesis problem in two steps: Given the specification of a protocol as a finite automaton, (i) synthesize (if exists) a distributable net, and then (ii) derive a network of communicating automata from the distributable net. While the second step is automatic and straightforward, the first step is in essence a computer assisted design task, where the distributed Petri-net synthesis algorithm helps the designer to refine the protocol specification into a graph isomorphic to the marking graph of a distributable net.
5. Software

5.1. FGb

Participant: J.C. Faugère [contact].

FGb/Gb is a powerful software for computing Gröbner bases; it is written in C/C++ (approximately 250000 lines counting the old Gb software).

5.2. FGb

Participant: Jean-Charles Faugere [correspondant].

FGb is a powerful software for computing Groebner bases. It includes the new generation of algorithms for computing Gröbner bases polynomial systems (mainly the F4,F5 and FGLM algorithms). It is implemented in C/C++ (approximately 250000 lines), standalone servers are available on demand. Since 2006, FGb is dynamically linked with Maple software (version 11 and higher) and is part of the official distribution of this software. See also the web page http://www-salsa.lip6.fr/~jcf/Software/FGb/index.html.

- ACM: I.1.2 Algebraic algorithms
- Programming language: C/C++

5.3. RAGlib

Participant: M. Safey El Din [contact].

RAGLib is a Maple library for computing sampling points in semi-algebraic sets.

5.4. Epsilon

Participant: D. Wang [contact].

Epsilon is a library of functions implemented in Maple and Java for polynomial elimination and decomposition with (geometric) applications.
SECRET Project-Team (section vide)
5. Software

5.1. Tookan

Participants: Graham Steel [correspondant], Romain Bardou.

See also the web page http://secgroup.ext.dsi.unive.it/projects/security-apis/pkcs11-security/tookan/.

Tookan is a security analysis tool for cryptographic devices such as smartcards, security tokens and Hardware Security Modules that support the most widely-used industry standard interface, RSA PKCS#11. Each device implements PKCS#11 in a slightly different way since the standard is quite open, but finding a subset of the standard that results in a secure device, i.e. one where cryptographic keys cannot be revealed in clear, is actually rather tricky. Tookan analyses a device by first reverse engineering the exact implementation of PKCS#11 in use, then building a logical model of this implementation for a model checker, calling a model checker to search for attacks, and in the case where an attack is found, executing it directly on the device. Tookan has been used to find at least a dozen previously unknown flaws in commercially available devices.

The first results using Tookan were published in 2010 [56] and a six-month licence was granted to Boeing to use the tool. In 2011, this transfer activity has continued, principally in combination with a major UK bank. In June, Tookan was used by Steel and Focardi two days of testing on devices belonging to the bank. Following these results, in September, a more significant contract was signed granting the bank 18 months of use of Tookan to test all their in-house equipment. Initial feedback has been very positive.

Tookan is the subject of a CSATT transfer action resulting in the hiring of an engineer, Romain Bardou, who started on September 1st. Early progress in re-implementing key parts of Tookan to improve modularity and overall code quality has been excellent. The next steps for Tookan are still being investigated: the Tookan project is the subject of a ‘qualification’ procedure by IT2 who will evaluate its suitability as the basis for a start-up company. At the same time other options are being considered, such as partnership with an existing SME. A decision is expected in mid-2012.

5.2. Orchids

Participants: Jean Goubault-Larrecq [correspondant], Hedi Benzina, Baptiste Gourdin, Nasr-Eddine Yousfi.

The ORCHIDS real-time intrusion detection system was created in 2003-04 at SECSI. After a few years where research and development around ORCHIDS was relatively quiet, several new things happened, starting from the end of 2010.

First, several companies and institutions expressed interest in ORCHIDS, among which, notably, EADS Cassidian, Thalès, Galois Inc. (USA), the French Direction Générale de l’Armement (DGA).

Second, Baptiste Gourdin was hired as a development engineer (Dec. 2010-Nov. 2011) on an Action de Développement Technologique (ADT). He improved Orchids in several ways. Its user interface benefitted from a complete revamping. New features were implemented, such as conformance with the IODEF and IDMEF standards, connection with vulnerability and network topology databases, the possibility to do forensics that synchronize past events to the state that the above databases were in at the time of the events, among others.

Nasr-Eddine Yousfi has followed up on Baptiste Gourdin, starting from December 2011, on an ITI engineer position allotted by INRIA’s CSATT.

Hedi Benzina implemented a tool on top of ORCHIDS, RuleGen, which allows one to write simple security policies that compile to ORCHIDS rules.

The efforts done in 2011 around ORCHIDS should be seen as the first steps in the creation of an open source consortium, which will be consolidated in the next years.
5.3. AKISS and SubVariant

Participant: Ştefan Ciobăcă.

AKISS (http://www.lsv.ens-cachan.fr/~ciobaca/akiss/) is a tool implementing a procedure for verifying trace equivalence (or equivalently may-testing equivalence) for bounded security processes with no else branches employing cryptographic primitives modeled by an optimally reducing rewrite system.

Trace equivalence can be used to model strong secrecy, vote-privacy and other security properties.

AKISS uses a fully-abstract encoding of symbolic traces into Horn clauses, thereby extending the KISS tool (http://www.lsv.ens-cachan.fr/~ciobaca/kiss/), which can only check static equivalence.

In order to get rid of the equational theory modeling the cryptographic primitives, AKISS employs algorithms for computing strongly complete sets of variants and complete set of unifiers of the SubVariant tool. AKISS is described in an article submitted to ESOP, in Chapter 5 of Ştefan Ciobăcă’s PhD thesis [12].

SubVariant (http://www.lsv.ens-cachan.fr/~ciobaca/subvariant/) is a tool for computing finite strongly complete set of variants modulo a convergent optimally reducing term rewriting system. SubVariant can also compute complete sets of equational unifiers for equational theories implemented by a convergent optimally reducing term rewriting system.

Complete sets of variants and the finite variant property were introduced in [59]. In [33], Ştefan Ciobăcă defines strongly complete sets of variants, which are more natural and more useful. Chapter 3 in Ştefan Ciobăcă’s PhD thesis describes extensively the algorithms behind SubVariant.
5. Software

5.1. ECPP

F. Morain has been continuously improving his primality proving algorithm called ECPP, originally developed in the early 1990s. Binaries for version 6.4.5 have been available since 2001 on his web page. Proving the primality of a 512 bit number requires less than a second on an average PC. His personal record is around 25,000 decimal digits, with the fast version he started developing in 2003. All of the code is written in C, and based on publicly available packages (GMP, mpfr, mpc, mpfrx).

5.2. SEA

Together with E. Schost and L. DeFeo, F. Morain has developed a new implementation of the SEA algorithm that computes the cardinality of elliptic curves over finite fields (large prime case, case $p = 2$). It uses NTL and includes the more recent algorithms for solving all subtasks. The large prime case is relevant to cryptographical needs. The $p = 2$ case, though not directly useful, is a good testbed for the FAAST program of LDeFeo (see 5.4). This program forms a gforge project.

5.3. TIFA

The TIFA library (short for Tools for Integer FActorization) was initially developed in 2006 and has been continuously improved during the last few years. TIFA is made up of a base library written in C99 using the GMP library, together with stand-alone factorization programs and a basic benchmarking framework to assess the performance of each algorithm.

As of November 2011, the library includes the following algorithms:

- CFRAC (Continued FRACtion factorization [64])
- ECM (Elliptic Curve Method)
- Fermat (McKee’s “fast” variant of Fermat’s algorithm [62])
- SIQS (Self-Initializing Quadratic Sieve [35])
- SQUFOF (SQUare FOrm factorization [54])

The complete TIFA package has been registered at the French Agency for Software Protection (APP – http://app.legalis.net/) on June, 1st 2011 with the Inter Deposit Digital Number:

IDDN.FR.001.220019.000.S.A.2011.000.31235.

It is now available online at http://www.lix.polytechnique.fr/Labo/Jerome.Milan/tifa/tifa.xhtml and distributed under the Lesser General Public License, version 2.1 or later.

5.4. FAAST

The FAAST library is developed in C++ by L. De Feo and makes use of the NTL library. It implements the algorithms presented in [4], plus other algorithms needed by the author for his research on explicit isogenies. Version 0.2.0, released on July 11 2009, is available at http://www.lix.polytechnique.fr/Labo/Luca.De-Feo/FAAST/. The source code is distributed under the General Public License version 2 or higher.

FAAST is a very efficient library for lattices of extensions of finite fields. Our aim is to add support for arbitrary finite fields, making it an essential building block for efficient computer algebra systems.
5.5. Quintix

The Quintix library is a Mathemagix package available at http://www.mathemagix.org/www/main/index.en.html. It is developed in C++ within the Mathemagix computer algebra system. It implements basic arithmetic for Galois rings and their unramified extensions, basic functions for the manipulation of Reed-Solomon codes and the complete Sudan list-decoding algorithm. It also implements the root-finding algorithms presented in [30]. The source code is distributed under the General Public License version 2 or higher.

Quintix is a very efficient library for Galois rings, extensions of Galois rings and root-finding in Galois rings.

5.6. APIP

As part of his activity in the PACE ANR, J. Milan completed, under the supervision of A. Enge, the development of APIP (Another Pairings Implementation in PARI), a PARI/GP module to compute state-of-the-art cryptographic pairings over elliptic curves. This module was intended to be an experimental framework for comparing the performances of the main cryptographic pairings with an emphasis on the standard 128, 192 and 256 bit high security levels.

APIP implements the Tate, Weil, ate and twisted ate pairings together with some optimal variants of the ate and twisted ate pairings for some elliptic curve families. Due to its very flexible architecture, it makes it easy to select several algorithm variants for each step of a pairing computation for a finer analysis.

Due to its emphasis on pairings for cryptographic purposes only, it is doubtful that the APIP module will be integrated in the upstream PARI/GP code base. We hope to be able to distribute APIP as an independent module in the near future, ideally under an open-source licence.
TASC Project-Team

5. Software

5.1. CHOCO

Participants: Nicolas Beldiceanu, Alexis De Clerq, Sophie Demassey, Jean-Guillaume Fages, Narendra Jussien [correspondant], Arnaud Letort, Xavier Lorca [correspondant], Arnaud Malapert, Julien Menana, Thierry Petit, Charles Prud’homme [correspondant].

CHOCO is a Java discrete constraints library integrating within a same system explanations, soft constraints and global constraints (90000 lines of source code). This year developments were focussing on the following aspects:

1. Providing a complete solver independent specification of explanation algorithms, data structure for encoding «nogoods» and treatment algorithms. A reference implementation is being made within the new version of our solver CHOCO.
2. Design and development of a dedicated languages to specify the propagation and the search heuristics of constraint solvers.
3. Providing efficient implementation of filtering algorithms for constraints such as tree, increasing_sum, cumulative with resource overload.
4. Providing an implementation of a probabilistic model for alldifferent.


5.2. IBEX

Participants: Gilles Chabert [correspondant], Rémi Douence.

IBEX (Interval-Based EXplorer) is a C++ library for solving nonlinear constraints over real numbers (25000 lines of source code). The main feature of Ibex is its ability to build solver/paver strategies declaratively through the contractor programming paradigm. Ibex includes a parser of the QUIMPER language (QUick Interval Modeling and Programming in a bounded-ERror context) and is currently used in several academic research labs.

G. Chabert and R. Douence (ASCOLA) have contributed in 2011 to the ongoing redesign of the architecture IBEX, the goal being to make it more flexible to cope with specific problems, and more easy to use. The link to the system and documentation is http://www.emn.fr/z-info/ibex/.

5.3. Global Constraint Catalog

Participants: Nicolas Beldiceanu [correspondant], Sophie Demassey, Mats Carlsson, Helmut Simonis.

The global constraint catalog presents and classifies global constraints and describes different aspects with meta data. It consist of

1. a pdf version that can be downloaded from http://www.emn.fr/z-info/sdemasse/gccat/ (at item last working version) containing 360 constraints, 3000 pages and 700 figures,
2. an on line version accessible from the previous address,
3. meta data describing the constraints (button PL for each constraint, e.g., alldifferent.pl),
4. an online service (i.e., a constraint seeker) which provides a web interface to search for global constraints, given positive and negative ground examples.
This year developments were focussing on:

1. maintaining the catalogue,
2. deploying an on-line constraint seeker [16] (see http://seeker.mines-nantes.fr/ and http://4c.ucc.ie/~hsimonis/seekerhelp.html for explanation how to use),
3. providing the negation for constraints defined by automata (with and without counter),
4. defining properties of constraints arguments, and
5. providing modelling examples as well as points of interests and common misunderstanding for core constraints.

N. Beldiceanu, S. Demassey, M. Carlsson (SICS, Sweden) and H. Simonis (4C, Ireland) have contributed in 2011. The link to the global constraint catalog is http://www.emn.fr/z-info/sdemasse/gccat/.
5. Software

5.1. MPIGate: Multi-Protocols Interface and Gateway for telehomecare and environment monitoring and control

Participants: Shahram Nourizadeh, Hugo Cruz Sanchez, Ye-Qiong Song.

For developing AAL (Ambient Assisted Living) or more generally the environment monitoring and control systems, heterogeneous wireless and wired networks will be used. To solve firstly the interoperability problems, and then to ensure the application required QoS, we developed a software prototype called MPIGate. MPIGate includes two important components: a user interface for telehomecare and home automation, and a gateway for ensuring the interworking of the different networks. In 2010, MPIGate has been laureate of the 12th national contest for the creation of innovative technology companies by the ministry of higher education and research (“Emergence” category). During 2011, MPIGate has been implemented on an embedded linux board and integrated into LORIA smart room platform within CPER IS project (http://infositu.loria.fr) [25][45]. In its current version, the gateway ensures the communication between IP (Ethernet and Wifi), home automation network (KNX), Bluetooth and Zigbee. Heterogeneous sensors can be now easily used through MPIGate interface for further building the activity monitoring of the elderly person living along at home or other application scenarios.

5.2. SAMOVAR

Participants: Adrien Guénard, Lionel Havet, Françoise Simonot-Lion.

Wireless Sensor and Actuator Networks (WSANs) combine sensors and actuators interconnected by wireless networks in order to perform distributed sensing and acting tasks. Closed-loop controllers can therefore be deployed on WSANs. Such systems have to meet specific requirements in terms of performance, dependability, energy and cost which raises great challenges due to the unreliability of wireless communications. A way to ensure that a system meets the required properties is to model it and go through its analysis. Building a model requires both deep knowledge on the system as well as on the used framework. Therefore there is a need for frameworks well-suited to the targeted systems and to the properties to verify. We proposed an approach meeting these conditions and a simulation framework, Samovar, based on Matlab / Simulink, allowing the modeling of the network protocols (Mac and routing services) and the resources sharing policy thanks to the TrueTime toolbox. Several classes of components (application, nodes, networks and middleware) and a clear semantics for their composition are identified. Furthermore, the design of Samovar was also driven by the need to easily transfer software component model between the concrete systems and its simulated model. The modeling and simulation method as well as the Samovar framework were assessed on several case studies: cooperating robots, intelligent living environment, embedded controllers on UAV robots... The simulation framework is available from http://samovar.loria.fr/. This work is supported by INRIA through the ADT SAMOVAR.

5.3. ANR Open-PEOPLE platform

Participants: Sophie Alexandre, Jonathan Ponroy, Kévin Roussel, Olivier Zendra.

The aim of Open-PEOPLE is to provide a platform for estimating and optimizing the power and energy consumption of systems. The Open-PEOPLE project formally started in April 2009. Two systems administrator and software developers had been hired initially: Sophie Alexandre and Kévin Roussel. Another system administrator and software developer, Jonathan Ponroy, joined them in 2010 when he finished his work on the ANR MORE project where he worked previously. Sophie Alexandre contract ended in February 2011.
Since the beginning of the Open-PEOPLE project, we had made significant progress in setting up the infrastructure for the software part of the platform, for which INRIA Nancy Grand Est is responsible. We had included new features to be able to fully integrate and test software developed as Eclipse plugins, relying on the Buckminster tool. We had also created a specific extension set for SVN and Hudson, called OPCIM (Open-PEOPLE Continuous Integration Mechanism). OPCIM had been registered at APP on 13/04/2010 with number IDDN.FR.001.150008.000.S.P.2010.000.10000.

Concerning the Open-PEOPLE platform itself, we had first tackled the high-level work, working with our partners on the definition of the requirements of the platform according to the needs of industry. We had then realized the specification work to define the global perimeter of our platform, according to the previous requirements. As part of this work had also been designed exchanges formats between the various tools. We had also designed at INRIA Nancy Grand Est a Tools integration Protocol, which specified requirements for external tools to be integrated in our platform. All this design work had been materialized in several reports which were deliveries provided to ANR.

We had also designed and developed an authentication component (Eclipse plugin) for the platform, so as to be able to provide a unique, secured access gate to the platform to all the tools that are or shall be integrated into it.

We had also started and almost finished developing an Internet portal giving access and control to the Open-PEOPLE Hardware Platform, located at our partner’s UBS in Lorient. Our portal features included user account management facilities, on the admin side, and on the user side, the ability to create, save, edit, reuse and of course submit jobs, make reservations for the hardware platform resources and get back tests results.

Finally, we had started working on two important parts of the software platform.

First, a way to unify the user experience despite the fact the platform federates several tools which were not developed to interact together. This implied an important and in-depth study of the wanted ergonomics for the platform, which involved taking into account both user needs and habits and the features of the available software tools.

The second work which had begun in 2011 was the design (then implementation) of the communications of between the various tools of the platform. This skeleton will be a key part of our platform, and the quality of its design will have a tremendous impact on its maintainability and its extensibility.

Note that the Open-PEOPLE project had been successfully evaluated on 14/09/2010 by ANR. Developments done during the first two years in the project are detailed in the 2009 and 2010 activity reports. In 2011, these developments went on.

We continued the work to solidify our development platform supporting our work and that of our partners. We produced a finer grained definition of the software platform functionalities, and a more precise definition of the tools integration protocol. We worked towards the corresponding implementation documents, adding two new deliverables about the architecture of the software platform and the ergonomics of the software platform. For the latter, we extensively interviewed user about ergonomics and designed several GUI mockups. We progressed on the implementation of the software platform, especially with respect to the internet portal to remote-control the hardware platform. We participated to the definition of the hardware platform and its functionalities, and participated actively to the work on the Specification document for HW / SW interfacing. We provided the first concrete design and implementation of the HW/SW platform interfacing, with our implementation of the remote control portal for the HW platform. This remote control module was completed in Fall 2011.

We also participated to the work pertaining to basic components model homogenization, by reviewing this in the context of the software platform architecture and implementation, which resulted in several incremental improvements of the underlying models. Finally, progressing towards the first release of the software part of the Open-PEOPLE platform, we realized an ergonomic study for the consumption laws editors, with mockups and user interviews and validation. We worked on the implementation of the editors for the consumption laws, which required learning new environments and development tools (related to the EMF framework and the AADL, QUDV and MathML models). As a consequence, we completed the implementation of the GUI and
engine to create units and quantities. We finalized the architecture needed to integrate external modules in the
platform.

With this progress, the first release of the whole Open-PEOPE software platform is expected early 2012.

5.4. VITRAIL

Participants: Damien Bodenes, Pierre Caserta, Olivier Zendra.

The aim of the VITRAIL operation is to provide tools for the advanced and immersive visualization of
programs. It partners with the University of Montréal, University of Montpellier and Pareo team of INRIA
Nancy Grand Est.

Last years, in VITRAIL, we had developed software to instrument and trace Java programs at the bytecode
level. We then had developed an analysis tool able to exploit these traces to compute relevant software metrics.
We had hired Damien Bodenes as software developer, and had begun the work on a prototype able to render
a 3D world, symbolizing software, onto various visualization hardware, with the possibility to change the
display metaphor. The main part of our development work had been in 2009 the choice and validation of the
technology, and a first architecture. In 2010, the development had go on at a good pace, building on chosen
technologies and architecture. This had brought new experience, and with the first actual runs of our platform,
we had realized that with the Irrlicht platform we had chosen, we could reach unforeseeable problem when
scaling up. We had thus decided to reverse our choice to the Ogre3D 3D engine at the beginning of 2010. Our
development had then progressed steadily.

We had released in 2010 a first prototype of our platform, with all the underlying architecture, able to provide
navigation features and interaction capacities limited to the driving of the navigation, as per our plans. This
had included dual screen management.

Our first prototype, using 2 large 2D screens, with a city metaphor, had been demonstrated during the "Fête
de la Science" in November 2010 and had received a lot of attention and enthusiasm from the general public.
About 55 persons per day had visited our booth and got demonstrations.

We had also progressed significantly in our Java bytecode tracer, by improving its granularity, the completeness
of the traced information, and its performance as well. We have a unique tool which is able to trace both
program classes and JDK classes, at basic block level. In addition, it does so with a dynamic instrumentation
of classes, which means there is no need to have an instrumented version of the class files on disk. This is very
convenient, especially when changing machine of JVM, or when upgrading either the JDK or the program
itself. In addition, the performance is good enough that the instrumented programs are still fully usable in an
interactive way, without bothering the user. To the best of our knowledge, this is the only Java bytecode tracer
that offers these features nowadays.

Our software development had lead to several registrations with APP:

- VITRAIL - Visualizer had been first registered on 29/12/2009 under number
  IDDN.FR.001.530021.000.S.P.2009.000.10000.
- VITRAIL - Tracer, was registered at APP on 20/09/2010 with number IDDN.FR.001.380001.000.S.P.
  2010.000.10000.

In 2011, we acquired a workstation and three 30 inches computer screens, to be able to set up a "boxed 3D
workstation", that would provide display in front and on both sides of the operator. This would constitute the
next step in our experiments, by improving immersion with a larger field of vision (on the sides). The software
developments to do this are ongoing. We also integrated a WiiMote interaction device to our system, but our
experiments found that its spacial resolution was too poor for our needs.

We finally improved significantly our VITRAIL prototype in 2011, especially by designing and implementing
a new representation for the relations between software (hence visual) elements, with limited clutter and the
possibility to regroup links and see their direction.
5. Software

5.1. Coq

Participants: Bruno Barras [Contact], Jean-Marc Notin, Arnaud Spiwack, Enrico Tassi.

Coq is a major proof system and the primary object and/or tool of our research. Its development is now mainly coordinated by the πr² INRIA Paris-Rocquencourt project-team, and some members of the TYPICAL team are active developers of the system.

5.2. Coqfinitegroup

Participants: Cyril Cohen, Assia Mahboubi [Contact].

Coqfinitegroup is the development corresponding to the ongoing effort to formalize the proof of the Feit-Thompson theorem. It is probably the most advanced formal development of group theory today. Its current size is about 80,000 lines of (compact) Coq code. Assia Mahboubi and Cyril Cohen are actively participating to this long term formalization project.

5.3. Dedukti

Participants: Mathieu Boespflug [Contact], Gilles Dowek.

Dedukti is a universal proof checker, based on the $\lambda\pi$-calculus modulo formalism. Mainly developed by Mathieu Boespflug, it is distributed under the GNU licence. The main system includes about 2000 lines of Haskell.

5.4. Ssreflect

Participants: Assia Mahboubi [Contact], Enrico Tassi.

SSReflect is a proof language extension of Coq developed under Georges Gonthier (Microsoft Research). It was originally designed to make the formalization of the Four Color Theorem possible and has been evolving since. It is important to note that it is shipped with redesigned basic proof libraries. Members of the TYPICAL are in charge of the documentation and distribution of this extension.
5. Software

5.1. The CADP Toolbox

Participants: Iker Bellicot, Hubert Garavel [contact person], Yann Genevois, Rémi Hérilier, Frédéric Lang, Radu Mateescu, Christine McKinty, Wendelin Serwe, Damien Thivolle.

We maintain and enhance CADP (*Construction and Analysis of Distributed Processes* – formerly known as CÆSAR/ALDÉBARAN Development Package) [9], a toolbox for protocols and distributed systems engineering (see [http://cadp.inria.fr](http://cadp.inria.fr)). In this toolbox, we develop and maintain the following tools:

- **CÆSAR.ADT** [3] is a compiler that translates LOTOS abstract data types into C types and C functions. The translation involves pattern-matching compiling techniques and automatic recognition of usual types (integers, enumerations, tuples, etc.), which are implemented optimally.
- **CÆSAR** [11] is a compiler that translates LOTOS processes into either C code (for rapid prototyping and testing purposes) or finite graphs (for verification purpose). The translation is done using several intermediate steps, among which the construction of a Petri net extended with typed variables, data handling features, and atomic transitions.
- **OPEN/CÆSAR** [4] is a generic software environment for developing tools that explore graphs on the fly (for instance, simulation, verification, and test generation tools). Such tools can be developed independently of any particular high level language. In this respect, OPEN/CÆSAR plays a central role in CADP by connecting language-oriented tools with model-oriented tools. OPEN/CÆSAR consists of a set of 16 code libraries with their programming interfaces, such as:
  - **CAESAR_GRAPH**, which provides the programming interface for graph exploration,
  - **CAESAR_HASH**, which contains several hash functions,
  - **CAESAR_SOLVE**, which resolves boolean equation systems on the fly,
  - **CAESAR_STACK**, which implements stacks for depth-first search exploration, and
  - **CAESAR_TABLE**, which handles tables of states, transitions, labels, etc.

A number of tools have been developed within the OPEN/CÆSAR environment, among which:

- **BISIMULATOR**, which checks bisimulation equivalences and preorders,
- **CUNCTATOR**, which performs on-the-fly steady-state simulation of continuous-time Markov chains,
- **DETERMINATOR**, which eliminates stochastic nondeterminism in normal, probabilistic, or stochastic systems,
- **DISTRIBUTOR**, which generates the graph of reachable states using several machines,
- **EVALUATOR**, which evaluates regular alternation-free μ-calculus formulas,
- **EXECUTOR**, which performs random execution,
- **EXHIBITOR**, which searches for execution sequences matching a given regular expression,
- **GENERATOR**, which constructs the graph of reachable states,
- **PROJECTOR**, which computes abstractions of communicating systems,
- **REDUCTOR**, which constructs and minimizes the graph of reachable states modulo various equivalence relations,
- **SIMULATOR**, **XSIMULATOR**, and **OCIS**, which allow interactive simulation, and
• TERMINATOR, which searches for deadlock states.

- **BCG (Binary Coded Graphs)** is both a file format for storing very large graphs on disk (using efficient compression techniques) and a software environment for handling this format. BCG also plays a key role in CADP as many tools rely on this format for their inputs/outputs. The BCG environment consists of various libraries with their programming interfaces, and of several tools, such as:
  - BCG_DRAW, which builds a two-dimensional view of a graph,
  - BCG_EDIT, which allows to modify interactively the graph layout produced by BCG_DRAW,
  - BCG_GRAPH, which generates various forms of practically useful graphs,
  - BCG_INFO, which displays various statistical information about a graph,
  - BCG_IO, which performs conversions between BCG and many other graph formats,
  - BCG_LABELS, which hides and/or renames (using regular expressions) the transition labels of a graph,
  - BCG_MERGE, which gathers graph fragments obtained from distributed graph construction,
  - BCG_MIN, which minimizes a graph modulo strong or branching equivalences (and can also deal with probabilistic and stochastic systems),
  - BCG_STEADY, which performs steady-state numerical analysis of (extended) continuous-time Markov chains,
  - BCG_TRANSIENT, which performs transient numerical analysis of (extended) continuous-time Markov chains, and
  - XTL (eXecutable Temporal Language), which is a high level, functional language for programming exploration algorithms on BCG graphs. XTL provides primitives to handle states, transitions, labels, successor and predecessor functions, etc.

  For instance, one can define recursive functions on sets of states, which allow to specify in XTL evaluation and diagnostic generation fixed point algorithms for usual temporal logics (such as HML [60], CTL [53], ACTL [55], etc.).

- The connection between explicit models (such as BCG graphs) and implicit models (explored on the fly) is ensured by OPEN/CÆSAR-compliant compilers, e.g.:
  - BCG_OPEN, for models represented as BCG graphs,
  - CÆSAR.OPEN, for models expressed as LOTOS descriptions,
  - EXP.OPEN, for models expressed as communicating automata,
  - FSP.OPEN, for models expressed as FSP [66] descriptions,
  - LNT.OPEN, for models expressed as LOTOS NT descriptions, and
  - SEQ.OPEN, for models represented as sets of execution trace.

The CADP toolbox also includes TGV (Test Generation based on Verification), developed by the VERIMAG laboratory (Grenoble) and the VERTECS project team at INRIA Rennes.

The CADP tools are well-integrated and can be accessed easily using either the EUCALYPTUS graphical interface or the SVL [6] scripting language. Both EUCALYPTUS and SVL provide users with an easy and uniform access to the CADP tools by performing file format conversions automatically whenever needed and by supplying appropriate command-line options as the tools are invoked.

5.2. **The TRAIAN Compiler**

**Participants:** Hubert Garavel [contact person], Frédéric Lang.
We develop a compiler named TRAIAN for translating descriptions written in the LOTOS NT language (see § 3.2 ) into C programs, which will be used for simulation, rapid prototyping, verification, and testing.

The current version of TRAIAN performs lexical analysis, syntactic analysis, abstract syntax tree construction, static semantics analysis, and C code generation for LOTOS NT types and functions.

Although this version of TRAIAN is still incomplete (it does not handle LOTOS NT processes), it already has useful applications in compiler construction [ 8 ]. The recent compilers developed by the VASY project team — including AAL, EVALUATOR 4.0 (see § 6.1.6 ), EXP.OPEN 2.0 (see § 6.1.4 ), LNT2LOTOS (see § 6.2.2 ), NTIF (see § 3.2 ), PIC2LNT (see § 6.2.3 ), and SVL (see § 6.1.4 ) — all contain a large amount of LOTOS NT code, which is then translated into C code by TRAIAN.

Our approach consists in using the SYNTAX tool (developed at INRIA Rocquencourt) for lexical and syntactic analysis together with LOTOS NT for semantical aspects, in particular the definition, construction, and traversal of abstract trees. Some involved parts of the compiler can also be written directly in C if necessary. The combined use of SYNTAX, LOTOS NT, and TRAIAN proves to be satisfactory, in terms of both the rapidity of development and the quality of the resulting compilers.

The TRAIAN compiler can be freely downloaded from the VASYWeb site (see http://vasy.inria.fr/traian ).
5. Software

5.1. QI: Quadrics Intersection

QI stands for “Quadrics Intersection”. QI is the first exact, robust, efficient and usable implementation of an algorithm for parameterizing the intersection of two arbitrary quadrics, given in implicit form, with integer coefficients. This implementation is based on the parameterization method described in [10], [32], [33], [34] and represents the first complete and robust solution to what is perhaps the most basic problem of solid modeling by implicit curved surfaces.

QI is written in C++ and builds upon the LiDIA computational number theory library [27] bundled with the GMP multi-precision integer arithmetic [26]. QI can routinely compute parameterizations of quadrics having coefficients with up to 50 digits in less than 100 milliseconds on an average PC; see [10] for detailed benchmarks.

Our implementation consists of roughly 18,000 lines of source code. QI has being registered at the Agence pour la Protection des Programmes (APP). It is distributed under the free for non-commercial use INRIA license and will be distributed under the QPL license in the next release. The implementation can also be queried via a web interface [28].

Since its official first release in June 2004, QI has been downloaded six times a month on average and it has been included in the geometric library EXACUS developed at the Max-Planck-Institut für Informatik (Saarbrücken, Germany). QI is also used in a broad range of applications; for instance, it is used in photochemistry for studying the interactions between potential energy surfaces, in computer vision for computing the image of conics seen by a catadioptric camera with a paraboloidal mirror, and in mathematics for computing flows of hypersurfaces of revolution based on constant-volume average curvature.

5.2. Isotop: Topology and Geometry of Planar Algebraic Curves

ISOTOP is a Maple software for computing the topology of an algebraic plane curve, that is, for computing an arrangement of polylines isotopic to the input curve. This problem is a necessary key step for computing arrangements of algebraic curves and has also applications for curve plotting. This software has been developed since 2007 in collaboration with F. Rouillier from INRIA Paris - Rocquencourt. It is based on the method described in [31] which incorporates several improvements over previous methods. In particular, our approach does not require generic position.

Isotop is registered at the APP (June 15th 2011) with reference IDDN.FR.001.240007.000.S.P.2011.000.10000. This version is competitive with other implementations (such as ALCIIX and INSULATE developed at MPII Saarbrücken, Germany and TOP developed at Santander Univ., Spain). It performs similarly for small-degree curves and performs significantly better for higher degrees, in particular when the curves are not in generic position.

We are currently working on an improved version integrating our new bivariate polynomial solver [22].

5.3. CGAL: Computational Geometry Algorithms Library

Born as a European project, CGAL (http://www.cgal.org) has become the standard library for computational geometry. It offers easy access to efficient and reliable geometric algorithms in the form of a C++ library. CGAL is used in various areas needing geometric computation, such as: computer graphics, scientific visualization, computer aided design and modeling, geographic information systems, molecular biology, medical imaging, robotics and motion planning, mesh generation, numerical methods...
In computational geometry, many problems lead to standard, though difficult, algebraic questions such as computing the real roots of a system of equations, computing the sign of a polynomial at the roots of a system, or determining the dimension of a set of solutions. We want to make state-of-the-art algebraic software more accessible to the computational geometry community, in particular, through the computational geometric library CGAL. On this line, we contributed a model of the Univariate Algebraic Kernel concept for algebraic computations [30] (see Sections 8.2.2 and 8.4). This CGAL package improves, for instance, the efficiency of the computation of arrangements of polynomial functions in CGAL [36]. We are currently developing a model of the Bivariate Algebraic Kernel based on our new bivariate polynomial solver [22]. This work is done in collaboration with F. Rouillier at INRIA Paris - Rocquencourt and L. Peñaranda at the university of Athens.
5. Software

5.1. The veriT solver

Participants: Diego Caminha Barbosa de Oliveira, David Déharbe, Pascal Fontaine [correspondant], Bruno Woltzenlogel Paleo.

The veriT solver is an SMT (Satisfiability Modulo Theories) solver developed in cooperation with David Déharbe from the Federal University of Rio Grande do Norte in Natal, Brazil. The solver can handle large quantifier-free formulas containing uninterpreted predicates and functions, and arithmetic on integers and reals. It features a very efficient decision procedure for difference logic, as well as a simplex-based reasoner for full linear arithmetic. It also has some support for user-defined theories, quantifiers, and lambda-expressions. This allows users to easily express properties about concepts involving sets, relations, etc. The prover can produce an explicit proof trace when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols and arithmetic. To support the development of the tool, a regression platform using INRIA’s grid infrastructure is used; it allows us to extensively test the solver on thousands of benchmarks in a few minutes.

The veriT solver is available as open source under the BSD license, and distributed through the web site http://www.veriT-solver.org. It entered for the third time the international competition of SMT solvers SMT-COMP 2011, a joint event with the SMT workshop 2011 and the CAV conference. As in the previous competitions, it performed decently against the other participating SMT solvers. It embeds an original symmetry reduction technique that greatly improved its efficiency on some categories of formulas. This technique was immediately incorporated also in other competing solvers, in particular Z3 (Microsoft) and CVC3 (University of New-York and University of Iowa).

Efforts in 2011 have been focused on the extension of the expressiveness of the tool (with improvements in the handling of quantifiers), and on its efficiency (which was significantly improved at different levels, including a purpose-built SAT solver underlying veriT). A lot of work was also devoted to improve the proof production of the tool, with the definition of a precise proof language. This proof language has been presented to the community as a standard for describing SMT proofs [17]. We are collaborating on this with Laurent Théry and Benjamin Grégoire (Marelle, INRIA Sophia-Antipolis), Laurent Voisin (Systerel), and Frédéric Besson (Celtique, INRIA Rennes).

Future research and implementation efforts will be directed to furthermore extend the accepted language, and increase the efficiency. We target applications where validation of formulas is crucial, such as the validation of TLA\(^+\) and B specifications, and work together with the developers of the respective verification platforms to make veriT even more useful in practice.

The software will be supported by an INRIA ADT, which will start at the beginning of 2012.

5.2. The TLA+ proof system

Participants: Stephan Merz, Hernán-Pablo Vanzetto.

TLAPS, the TLA\(^+\) proof system, is a platform for developing and mechanically verifying TLA\(^+\) proofs. It is developed at the Joint MSR-INRIA Centre. The TLA\(^+\) proof language is declarative and based on standard mathematical logic; it supports hierarchical and non-linear proof construction and verification. TLAPS consists of a proof manager that interprets the proof language and generates a collection of proof obligations that are sent to backend verifiers that include theorem provers, proof assistants, SMT solvers, and decision procedures.
TLAPS is publically available at http://msr-inria.inria.fr/~doligez/tlaps/, it is distributed under a BSD-like license. It handles the non-temporal part of TLA* with the exception of computing enabledness predicates and can currently be used to prove safety, but not liveness properties. Its backends include a tableau prover for first-order logic, an encoding of TLA* in the proof assistant Isabelle, as well as an SMT translation and a custom decision procedure for Presburger arithmetic. Our main contribution in 2011 has been the implementation of a new SMT backend that handles formulas including linear arithmetic, elementary set theory, functions, tuples, and records (see section 6.4). Other efforts in 2011 concerned improvements and stabilization of the fingerprinting technique that avoids reproving proof obligations that have remained unchanged since a previous prover run.
5. Software

5.1. TGV

Participant: Thierry Jéron.

TGV (Test Generation with Verification technology) is a tool for test generation of conformance test suites from specifications of reactive systems [4]. It is based on the IOLTS model, a well-defined theory of testing, and on-the-fly test generation algorithms coming from verification technology. Originally, TGV allows test generation focused on well-defined behaviors formalized by test purposes. The main operations of TGV are (1) a synchronous product which identifies sequences of the specification accepted by a test purpose, (2) abstraction and determinisation for the computation of next visible actions, (3) selection of test cases by the computation of reachable states from the initial states and co-reachable states from accepting states. TGV has been developed in collaboration with Vérimag Grenoble and uses libraries of the CADP toolbox (VERIMAG and VASY). TGV can be seen as a library that can be linked to different simulation tools through well-defined APIs. An academic version of TGV is distributed in the CADP toolbox and allows test generation from Lotos specifications by a connection to its simulator API. TGV has been registered at APP (Agence de Protection des Programmes) under deposit number IDDN.FR.001.310012.00.R.P.1997.000.2090.

5.2. STG

Participant: Thierry Jéron.

STG (Symbolic Test Generation) is a prototype tool for the generation and execution of test cases using symbolic techniques. It takes as input a specification and a test purpose described as IOSTS, and generates a test case program also in the form of IOSTS. Test generation in STG is based on a syntactic product of the specification and test purpose IOSTS, an extraction of the subgraph corresponding to the test purpose, elimination of internal actions, determinisation, and simplification. The simplification phase now relies on NBAC, which approximates reachable and coreachable states using abstract interpretation. It is used to eliminate unreachable states, and to strengthen the guards of system inputs in order to eliminate some Inconclusive verdicts. After a translation into C++ or Java, test cases can be executed on an implementation in the corresponding language. Constraints on system input parameters are solved on-the-fly (i.e., during execution) using a constraint solver. The first version of STG was developed in C++, using Omega as constraint solver during execution. This version has been deposited at APP under number IDDN.FR.001.510006.000.S.P.2004.000.10600.

A new version in OCaml has been developed in the last years. This version is more generic and will serve as a library for symbolic operations on IOSTS. Most functionalities of the C++ version have been re-implemented. Also a new translation of abstract test cases into Java executable tests has been developed, in which the constraint solver is LUCKYDRAW (VERIMAG). This version has also been deposit at APP and is available for download on the web as well as its documentation and some examples.

Finally, in collaboration with ULB, we implemented a prototype SMACS, derived from STG, that is devoted to the control of infinite system modeled by STS.

5.3. SIGALI

Participant: Hervé Marchand.
SIGALI is a model-checking tool that operates on ILTS (Implicit Labeled Transition Systems, an equational representation of an automaton), an intermediate model for discrete event systems. It offers functionalities for verification of reactive systems and discrete controller synthesis. It is developed jointly by the ESPRESSO and VERTECS teams. The techniques used consist in manipulating the system of equations instead of the set of solutions, which avoids the enumeration of the state space. Each set of states is uniquely characterized by a predicate and the operations on sets can be equivalently performed on the associated predicates. Therefore, a wide spectrum of properties, such as liveness, invariance, reachability and attractivity, can be checked. Algorithms for the computation of predicates on states are also available [6] [28]. SIGALI is connected with the Polychrony environment (ESPRESSO project-team) as well as the Matou environment (VERIMAG), thus allowing the modeling of reactive systems by means of Signal Specification or Mode Automata and the visualization of the synthesized controller by an interactive simulation of the controlled system. SIGALI is registered at APP.
ALEA Project-Team (section vide)
5. Software

5.1. Tralics

Participants: José Grimm [corresponding participant].

Tralics is a LaTeX to XML translator. It is a free software, distributed under the CeCILL license version two, in binary form for Linux, Windows and MacOS X. Sources are also available via its web page http://www-sop.inria.fr/apics/tralics. The development of the LaTeX to XML translator, named Tralics, was continued. Latest release is version 2.14.4, dated 07-11-2011 (see section 6.1).

5.2. RARL2

Participants: Jean-Paul Marmorat, Martine Olivi [corresponding participant].

RARL2 (Réalisation interne et Approximation Rationnelle L2) is a software for rational approximation (see section 3.1.4 ) http://www-sop.inria.fr/apics/RARL2/rarl2-eng.html. This software takes as input a stable transfer function of a discrete time system represented by either of

• its internal realization,
• its first $N$ Fourier coefficients,
• discretized values on the circle.

It computes a local best approximant which is stable, of prescribed McMillan degree, in the $L^2$ norm.

It is akin to the arl2 function of Endymion (see section 5.5) from which it differs mainly in the way systems are represented: a polynomial representation is used in Endymion, while RARL2 uses realizations. It is implemented in Matlab. This software handles multi-variable systems (with several inputs and several outputs), and uses a parametrization with the following advantages:

• it incorporates the stability requirement in a built-in manner,
• it allows the use of differential tools,
• it is well-conditioned, and computationally efficient.

An iterative research strategy on the degree of the local minima, similar in principle to that of arl2, increases the chance of obtaining the absolute minimum by generating, in a structured manner, several initial conditions. RARL2 performs the rational approximation step in our applications to filter identification (section 4.2) as well as sources or cracks recovery (section 4.1). It was released to the universities of Delft, Maastricht, Cork and Brussels. The parametrization embodied in RARL2 was recently used for a multi-objective control synthesis problem provided by ESTEC-ESA, The Netherlands. An extension of the software to the case of triple poles approximants is now available. It provides satisfactory results in the source recovery problem and it is used by FindSources3D (see 5.7).

5.3. RGC

Participants: Fabien Seyfert [corresponding participant], Jean-Paul Marmorat.
The identification of filters modeled by an electrical circuit that was developed by the team (see section 4.2) led us to compute the electrical parameters of the underlying filter. This means finding a particular realization \((A, B, C, D)\) of the model given by the rational approximation step. This 4-tuple must satisfy constraints that come from the geometry of the equivalent electrical network and translate into some of the coefficients in \((A, B, C, D)\) being zero. Among the different geometries of coupling, there is one called “the arrow form” [52] which is of particular interest since it is unique for a given transfer function and is easily computed. The computation of this realization is the first step of RGC. Subsequently, if the target realization is not in arrow form, one can nevertheless show that it can be deduced from the arrow-form by a complex-orthogonal change of basis. In this case, RGC starts a local optimization procedure that reduces the distance between the arrow form and the target, using successive orthogonal transformations. This optimization problem on the group of orthogonal matrices is non-convex and has many local and global minima. In fact, there is not even uniqueness of the filter realization for a given geometry. Moreover, it is often relevant to know all solutions of the problem, because the designer is not even sure, in many cases, which one is being handled. Moreover, the assumptions on the reciprocal influence of the resonant modes may not be equally well satisfied for all such solutions, hence some of them should be preferred for the design. Today, apart from the particular case where the arrow form is the desired form (this happens frequently up to degree 6) the RGC software provides no guarantee to obtain a single realization that satisfies the prescribed constraints. The software Dedale-HF (see 5.6), which is the successor of RGC, solves with guarantees this constraint realization problem.

5.4. PRESTO-HF

**Participant:** Fabien Seyfert.

PRESTO-HF: a toolbox dedicated to lowpass parameter identification for microwave filters [http://www-sop.inria.fr/apics/personnel/Fabien.Seyfert/Presto_web_page/presto_pres.html](http://www-sop.inria.fr/apics/personnel/Fabien.Seyfert/Presto_web_page/presto_pres.html). In order to allow the industrial transfer of our methods, a Matlab-based toolbox has been developed, dedicated to the problem of identification of low-pass microwave filter parameters. It allows one to run the following algorithmic steps, either individually or in a single shot:

- determination of delay components caused by the access devices (automatic reference plane adjustment),
- automatic determination of an analytic completion, bounded in modulus for each channel,
- rational approximation of fixed McMillan degree,
- determination of a constrained realization.

For the matrix-valued rational approximation step, Presto-HF relies either on hyperion (see 5.5) (Unix or Linux only) or RARL2 (platform independent), two rational approximation engines developed within the team. Constrained realizations are computed by the RGC software. As a toolbox, Presto-HF has a modular structure, which allows one for example to include some building blocks in an already existing software.

The delay compensation algorithm is based on the following strong assumption: far off the passband, one can reasonably expect a good approximation of the rational components of \(S_{11}\) and \(S_{22}\) by the first few terms of their Taylor expansion at infinity, a small degree polynomial in \(1/s\). Using this idea, a sequence of quadratic convex optimization problems are solved, in order to obtain appropriate compensations. In order to check the previous assumption, one has to measure the filter on a larger band, typically three times the pass band.

This toolbox is currently used by Thales Alenia Space in Toulouse and a license agreement has been recently negotiated with Thales airborne systems. XLim (University of Limoges) is a heavy user of Presto-HF among the academic filtering community and some free license agreements are currently being considered with the microwave department of the University of Erlangen (Germany) and the Royal Military College (Kingston, Canada).

5.5. Endymion

**Participant:** José Grimm.
The core of the Endymion system (a follow-up to hyperion) is formed by a library that handles numbers (short integers, arbitrary size rational numbers, floating point numbers, quadruple and octuple precision floating point numbers, arbitrary precision real numbers, complex numbers), polynomials, matrices, etc. Specific data structures for the rational approximation algorithm arl2 and the bounded extremal problem bep are also available. One can mention for instance splines, Fourier series, Schur matrices, etc. These data structures are manipulated by dedicated algorithms (matrix inversion, roots of polynomials, a gradient-based algorithm for minimizing $\psi$, Newton method for finding a critical point of $\psi$, etc), and input-output functions that allow one to save data on disk, restore them, plot them, etc. Interactivity is provided through a symbolic interpreter based upon a Lisp interpreter.

The development of Endymion, http://www-sop.inria.fr/apics/endymion/index.html has come to an end. The software is still maintained and sources are available on the ftp server.

5.6. Dedale-HF

**Participant:** Fabien Seyfert.

Dedale-HF is a software dedicated to solve exhaustively the coupling matrix synthesis problem in reasonable time for the users of the filtering community. For a given coupling topology, the coupling matrix synthesis problem (C.M. problem for short) consists in finding all possible electromagnetic coupling values between resonators that yield a realization of given filter characteristics (see section 6.6). Solving the latter problem is crucial during the design step of a filter in order to derive its physical dimensions as well as during the tuning process where coupling values need to be extracted from frequency measurements (see Figure 3).

![Figure 3. Overall scheme of the design and tuning process of a microwave filter.](image-url)
Dedale-HF consists in two parts: a database of coupling topologies as well as a dedicated predictor-corrector code. Roughly speaking each reference file of the database contains, for a given coupling topology, the complete solution to the C.M. problem associated to particular filtering characteristics. The latter is then used as a starting point for a predictor-corrector integration method that computes the solution to the C.M. problem of the user, i.e., the one corresponding to user-specified filter characteristics. The reference files are computed off line using Groebner basis techniques or numerical techniques based on the exploration of a monodromy group. The use of such a continuation technique combined with an efficient implementation of the integrator produces a drastic reduction, by a factor of 20, of the computational time.

Access to the database and integrator code is done via the web on http://www-sop.inria.fr/apics/Dedale/WebPages. The software is free of charge for academic research purposes: a registration is however needed in order to access full functionality. Up to now 90 users have registered world wide (mainly: Europe, U.S.A, Canada and China) and 4000 reference files have been downloaded.

As mentioned in 6.6 an extension of this software that handles symmetrical networks is under construction.

5.7. FindSources3D

Participants: Maureen Clerc [EPI Athena], Juliette Leblond [corresponding participant], Jean-Paul Mar-morat, Théo Papadopoulo [EPI Athena].

FindSources3D is a software dedicated to source recovery for the inverse EEG problem, in 3-layer spherical settings, from pointwise data (see http://www-sop.inria.fr/apics/FindSources3D/). Through the algorithm described in section 4.1, it makes use of RARL2 (section 5.2) for the rational approximation step in plane sections. The data transmission preliminary step (“cortical mapping”) is solved using boundary element methods through the software OpenMEEG (its CorticalMapping features) developed by the Athena Team (see http://www-sop.inria.fr/athena/software/OpenMEEG/). A first release of FindSources3D is now available, which will be demonstrated and distributed within the medical teams we are in contact with (see figure 4 and those in [30]).

5.8. Sollya

Participants: Sylvain Chevillard, Christoph Lauter [LIP6], Mioara Joldeș [Arénaire team until October; now with Uppsala University].

Sollya is an interactive tool where the developers of mathematical floating-point libraries (libm) can experiment before actually developing code. The environment is safe with respect to floating-point errors, i.e., the user precisely knows when rounding errors or approximation errors happen, and rigorous bounds are always provided for these errors.

Amongst other features, it offers a fast Remez algorithm for computing polynomial approximations of real functions and also an algorithm for finding good polynomial approximants with floating-point coefficients to any real function. It also provides algorithms for the certification of numerical codes, such as Taylor Models, interval arithmetic or certified supremum norms.

It is available as a free software under the CeCILL-C license at http://sollya.gforge.inria.fr/.
ASPI Project-Team (section vide)
5. Software

5.1. RealfluiDS

Participants: Dante de Santis, Gianluca Geraci, Pietro Marco Congedo, Rémi Abgrall [corresponding member].

RealfluiDS is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multimaterial, MHD flows and turbulent flows. There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in the 3D one. This software implements the more recent residual distribution schemes. The code has been parallelized with and without overlap of the domains. An Uncertainty Quantification library has been added to the software. A partitioning tool exists in the package, which uses Scotch. In the coming years, all the know how of RealfluiDS will be transferred to Aerosol.

5.2. AeroSol

Participants: Damien Genêt [corresponding member], Maxime Mogé, François Pellegrini, Vincent Perrier [corresponding member].

The software AeroSol is jointly developed in the teams Bacchus and Cagire. It is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes. The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Pumas. Maxime Mogé has been hired on a young engineer position (IJD) obtained in the ADT OuBa HOP for participating to the parallelization of the library, and arrived on November, 1st 2011.

Current features include

- **development environment** use of CMake for compilation, CTest for automatic tests and memory checking, lcov and gcov for code coverage reports.
- **In/Out** link with the XML library for handling with parameter files. Reader for GMSH, and writer on the VTK-ASCII legacy format.
- **Quadrature formula** up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles.
- **Finite elements** up to fourth degree for Lagrange finite elements on lines, triangles and quadrangles.
- **Geometry** elementary geometrical functions for first order lines, triangles, quadrangles.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order.
- **Linear Solvers** link with the external linear solver UMFPack.
- **Memory handling** discontinuous and continuous discretizations based on PaMPA for triangular and quadrangular meshes.
- **Numerical schemes** continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Scalar stabilized residual distribution schemes with explicit Euler time iteration have been implemented for steady problems.

5.3. SLOWS

Participant: Mario Ricchiuto [corresponding member].
SLOWS (Shallow-water FLOWS) is a C-platform allowing the simulation of free surface shallow water flows with friction. Arbitrary bathymetries are allowed, defined either by some complex piecewise analytical expression, or by $xy$-z data files, the classical Manning model for friction is used, and an Exner model is implemented for sediment transport. The equations are discretized with a residual based approach which is an adaptation of the schemes developed for aeronautics applications. Due to the inherent unsteadiness of these flows, the time discretization plays an important role. Three different approaches are available, based on conditionally depth-positivity preserving implicit schemes, or on conditionally depth-positivity preserving genuinely explicit discretizations, or on an unconditionally depth-positivity preserving space-time approach.

5.4. COCA

Participants: Mario Ricchiuto [corresponding member], Gérard Vignoles.

COCA (CodeOxydationCompositesAutocicatrisants) is a fortran-90 code for the simulation of the oxidation process in self-healing composite materials, developed in collaboration with the Laboratoire des Composites ThermoStructuraux in Bordeaux (UMR-5801 LCTS). This process involves the chemical oxidation of some of the matrix components of the composite, and the production of a liquid oxide that flows and fills material cracks, acting as a diffusion barrier against oxygen and thus protecting the ceramic fibers of the material. COCA simulates this process using a finite element discretization of the model equations. In its current version only transverse cracks are available. COCA makes use of PaStiX to solve the algebraic systems arising from the discretization.

5.5. PaStiX

Participant: Pierre Ramet [corresponding member].

This work is supported by the French “Commissariat à l’Énergie Atomique CEA/CESTA” in the context of structural mechanics and electromagnetism applications.

PaStiX (http://pastix.gforge.inria.fr) (Parallel Sparse matriX package) is a scientific library that provides a high performance parallel solver for very large sparse linear systems based on block direct and block ILU(k) iterative methods. Numerical algorithms are implemented in single or double precision (real or complex): LLt (Cholesky), LDLt (Crout) and LU with static pivoting (for non symmetric matrices having a symmetric pattern). This latter version is now used in RealfluiDS (see Section 5.1). The PaStiX library is released under INRIA CeCILL licence.

The PaStiX library uses the graph partitioning and sparse matrix block ordering package Scotch (see Section 5.7). PaStiX is based on an efficient static scheduling and memory manager, in order to solve 3D problems with more than 50 million of unknowns. The mapping and scheduling algorithm handles a combination of 1D and 2D block distributions. This algorithm computes an efficient static scheduling of the block computations for our supernodal parallel solver which uses a local aggregation of contribution blocks. This can be done by taking into account very precisely the computational costs of the BLAS 3 primitives, the communication costs and the cost of local aggregations. We also improved this static computation and communication scheduling algorithm to anticipate the sending of partially aggregated blocks, in order to free memory dynamically. By doing this, we are able to reduce the aggregated memory overhead, while keeping good performance.

Another important point is that our study is suitable for any heterogeneous parallel/distributed architecture when its performance is predictable, such as clusters of multicore nodes. In particular, we now offer a high performance version with a low memory overhead for multicore node architectures, which fully exploits the advantage of shared memory by using an hybrid MPI-thread implementation.

Direct methods are numerically robust methods, but the very large three dimensional problems may lead to systems that would require a huge amount of memory despite any memory optimization. A studied approach consists in defining an adaptive blockwise incomplete factorization that is much more accurate (and numerically more robust) than the scalar incomplete factorizations commonly used to precondition iterative solvers. Such incomplete factorization can take advantage of the latest breakthroughs in sparse direct methods and particularly should be very competitive in CPU time (effective power used from processors and good scalability) while avoiding the memory limitation encountered by direct methods.
5.6. HIPS

**Participant:** Pierre Ramet [corresponding member].

HIPS (Hierarchical Iterative Parallel Solver) is a scientific library that provides an efficient parallel iterative solver for very large sparse linear systems.

The key point of the methods implemented in HIPS is to define an ordering and a partition of the unknowns that relies on a form of nested dissection ordering in which cross points in the separators play a special role (Hierarchical Interface Decomposition ordering). The subgraphs obtained by nested dissection correspond to the unknowns that are eliminated using a direct method and the Schur complement system on the remaining of the unknowns (that correspond to the interface between the sub-graphs viewed as sub-domains) is solved using an iterative method (GMRES or Conjugate Gradient at the time being). This special ordering and partitioning allows for the use of dense block algorithms both in the direct and iterative part of the solver and provides a high degree of parallelism to these algorithms. The code provides a hybrid method which blends direct and iterative solvers. HIPS exploits the partitioning and multistage ILU techniques to enable a highly parallel scheme where several subdomains can be assigned to the same process. It also provides a scalar preconditioner based on the multistage ILUT factorization.

HIPS can be used as a standalone program that reads a sparse linear system from a file; it also provides an interface to be called from any C, C++ or Fortran code. It handles symmetric, unsymmetric, real or complex matrices. Thus, HIPS is a software library that provides several methods to build an efficient preconditioner in almost all situations.

Since August 2008, HIPS is publicly available at [http://hips.gforge.inria.fr](http://hips.gforge.inria.fr) under the INRIA CeCILL licence.

5.7. Scotch

**Participant:** François Pellegrini [corresponding member].

Scotch ( [http://www.labri.fr/~pelegrin/scotch/](http://www.labri.fr/~pelegrin/scotch/) ) is a software package for parallel and sequential sparse matrix ordering, parallel and sequential graph partitioning, as well as sequential static mapping, and mesh and hypergraph partitioning.

The initial purpose of Scotch was to compute high-quality partitions and static mappings of valuated graphs representing parallel computations and target architectures of arbitrary topologies. The original contribution consisted in developing a “divide and conquer” algorithm in which processes are recursively mapped onto processors by using graph bisection algorithms that are applied both to the process graph and to the architecture graph. This allows the mapper to take into account the topology and heterogeneity of the valuated graph which models the interconnection network and its resources (processor speed, link bandwidth). As new multicore, multinode parallel machines tend to be less uniform in terms of memory latency and communication bandwidth, this feature is regaining interest.

The software has then been extended in order to produce vertex separators instead of edge separators, using a multilevel framework. Recursive vertex separation is used to compute orderings of the unknowns of large sparse linear systems, which both preserve sparsity when factorizing the matrix and exhibit concurrency for computing and solving the factored matrix in parallel.

Version 5.0 of Scotch, released on August 2007, was the first version to comprise parallel routines. This extension, called PT-Scotch (for “Parallel Threaded Scotch”), is based on a distributed memory model, and makes use of the MPI and, optionally, Posix thread APIs. A distributed graph structure has been defined, which allows users to reserve vertex indices on each processor for future local adaptive refinement. Its parallel graph ordering routine provides orderings which are of the same quality as the ones yielded by the sequential Scotch ordering routine, while competing software ParMETIS experiences a severe loss of quality when the number of processors increase. Scotch5.0 was released under the CeCILL-C free/libre software license, and has been registered at APP (“Agence pour la Protection des Programmes”).
Version 5.1 of Scotch, released on September 2008, extended the parallel features of PT-Scotch, which can now compute graph partitions in parallel by means of a parallel recursive bipartitioning framework. Release 5.1.10 had made Scotch the first full 64-bit implementation of a general purpose graph partitioner, so that PT-Scotch has been able to successfully break the “32-bit” barrier and partition a graph above 2 billion vertices, spread across 2048 processors, at the French CCRT computer center.

Version 6.0, about to be released, offers new sequential features: static mapping with fixed vertices, static remapping, and static remapping with fixed vertices.

Scotch has been integrated in numerous third-party software, which indirectly contribute to its diffusion. For instance, it is used by the ZOLTAN module of the TRILINOS software (SANDIA Labs), by Code_ASTER Libre, a GPLed thermal and mechanical analysis software developed by French state-owned electricity producer EDF, by the parallel solvers MUMPS (ENSEEITH/IRIT, LIP and LaBRI), SuperLUDist (U.C. Berkeley), PaStiX (LaBRI) and HIPS (LaBRI), as well as by several other scientific computing software.

5.8. MMG3D

**Participant:** Cécile Dobrzynski [corresponding member].

MMG3D is a tetrahedral fully automatic remesher. Starting from a tetrahedral mesh, it produces quasi-uniform meshes with respect to a metric tensor field. This tensor prescribes a length and a direction for the edges, so that the resulting meshes will be anisotropic. The software is based on local mesh modifications and an anisotropic version of Delaunay kernel is implemented to insert vertices in the mesh. Moreover, MMG3D allows one to deal with rigid body motion and moving meshes. When a displacement is prescribed on a part of the boundary, a final mesh is generated such that the surface points will be moved according this displacement. MMG3D is used in particular in GAMMA for their mesh adaptation developments, but also at EPFL (maths department), Dassault Aviation, Lemma (a french SME), etc. MMG3D can be used in FreeFem++ (http://www.freefem.org), a free software which eases the solving of PDEs. More details can be found on http://www.math.u-bordeaux1.fr/~dobj/logiciels/mmg3d.php.

5.9. Montjoie

**Participant:** Marc Duruflé [corresponding member].

Montjoie is a finite element code initially handling only quadrilateral/hexaedral elements. Because of the tensorization of these elements, efficient algorithms can be written for the computation of finite element matrices. It can handle tetrahedra, prisms, pyramids, hexaedra with continuous finite element, edge elements and discontinuous Galerkin formulations. A local order of approximation can be used in each element of the mesh.

5.10. PLATO

**Participants:** Hervé Guillard [PUMAS], Laure Combe [PUMAS,contact], Cédric Lachat, Pierre Ramet [corresponding member].

The development of PLATO (A platform for Tokamak simulation) (http://www-sop.inria.fr/pumas/plato.php) is being supported by an ADT action of the D2T. PLATO is a suite of data and softwares dedicated to the geometry and physics of Tokamaks and its main objective is to provide the Inria large scale initiative “FUSION” teams working with plasma fluid models with a common development tool. The construction of this platform will integrate the following developments.

1. A (small) database corresponding to axi-symmetrical solutions of the equilibrium plasma equations for realistic geometrical and magnetic configurations (ToreSupra, JET and ITER). The construction of meshes is always an important time consuming task. Plato will provide meshes and solutions corresponding to equilibrium solutions that will be used as initial data for more complex computations.
2. A set of tool for the handling, manipulation and transformation of meshes and solutions using different discretisations (P1, Q1, P3, etc)
3. Numerical templates allowing the use of 3D discretization schemes using finite element schemes in the poloidal plane and spectral Fourier or structured finite volume representations in the toroidal one.

4. Several applications (Ideal MHD and drift approximation) used in the framework of the Inria large scale initiative “FUSION”.

5.11. PaMPA

Participants: Cécile Dobrzynski, Hervé Guillard [PUMAS], Laurent Hascoët [Tropics], Cédric Lachat, François Pellegrini [Corresponding member].

PaMPA (“Parallel Mesh Partitioning and Adaptation”) is a middleware library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. It is based on a distributed data structure that represents meshes as a set of entities (elements, faces, edges, nodes, etc.), linked by relations (that is, computation dependencies).

Version 0.1 allows users to declare a distributed mesh, declare values attached to the entities of the meshes (e.g. temperature attached to elements, pressures to the faces, etc.), exchange values between overlapping entities located at the boundaries of subdomains assigned to different processors, and iterate over the relations of entities (e.g. iterate over the faces of elements).

PaMPA is already used as the data structure manager for two solvers being developed at INRIA: Plato and Aerosol.

PaMPA will soon interface with Scotch for mesh redistribution, and with MMG3D to offer parallel remeshing features (in this particular example, for tetrahedral elements).
5. Software

5.1. Nonsmooth dynamics: Siconos

**Participants:** Vincent Acary, Maurice Bremond, Olivier Bonnefon.

In the framework of the European project Siconos, Bipop was the leader of the Work Package 2 (WP2), dedicated to the numerical methods and the software design for nonsmooth dynamical systems. The aim of this work is to provide a common platform for the simulation, modeling, analysis and control of abstract nonsmooth dynamical systems. Besides usual quality attributes for scientific computing software, we want to provide a common framework for various scientific fields, to be able to rely on the existing developments (numerical algorithms, description and modeling software), to support exchanges and comparisons of methods, to disseminate the know-how to other fields of research and industry, and to take into account the diversity of users (end-users, algorithm developers, framework builders) in building expert interfaces in Python and end-user front-end through Scilab.

After the requirement elicitation phase, the Siconos Software project has been divided into 5 work packages which are identified to software products:

1. **SICONOS/NUMERICS** This library contains a set of numerical algorithms, already well identified, to solve non smooth dynamical systems. This library is written in low-level languages (C,F77) in order to ensure numerical efficiency and the use of standard libraries (Blas, Lapack, ...)

2. **SICONOS KERNEL** This module is an object-oriented structure (C++) for the modeling and the simulation of abstract dynamical systems. It provides the users with a set of classes to describe their nonsmooth dynamical system (dynamical systems, intercations, nonsmooth laws, ...) and to perform a numerical time integration and solving.

3. **SICONOS/FRONT-END.** This module is mainly an auto-generated wrapper in Python which provides a user-friendly interface to the Siconos libraries. A scilab interface is also provided in the Front-End module.

4. **SICONOS/CONTROL** This part is devoted to the implementation of control strategies of non smooth dynamical systems.

5. **SICONOS/MULTIBODY.** This part is dedicated to the modeling and the simulation of multi-body systems with 3D contacts, impacts and Coulomb’s friction. It uses the Siconos/Kernel as simulation engine but relies on a industrial CAD library (OpenCascade and pythonOCC) to deal with complex body geometries and to compute the contact locations and distances.

Further informations may be found at [http://siconos.gforge.inria.fr/](http://siconos.gforge.inria.fr/)

5.2. Humanoid motion analysis and simulation

**Participant:** Pierre-Brice Wieber.

The HuMANs toolbox offers tools for the modelling, control and analysis of humanoid motion, be it of a robot or a human. It is a C/C++/Scilab/Maple-based set of integrated tools for the generation of dynamical models of articulated bodies with unilateral contact and friction, their simulation with an event-driven integration scheme, their 3D visualization, the computation of stability measures, optimal positions and trajectories, the generation of control laws and observers, the reconstruction of movements from different sensing systems.

5.3. AMELIF

**Participants:** Pierre-Brice Wieber, François Keith.
The AMELIF framework is an integrative framework that proposes an API for the representation and simulation of virtual scenes including articulated bodies. AMELIF was devised to realize interactive scenario studies with haptic feedback while providing an interface enabling fast and general prototyping of humanoids (avatars or robots). It is entirely developed in C++ and is cross-platform. The framework is articulated around a core library, upon which several modules have been developed for collision detection, dynamic simulation (contact handling in a time stepping scheme), 3D rendering, haptic interaction, posture generation. This framework is developed mostly at the CNRS/AIST UMI JRL, but we started using it in the Bipop team and therefore started contributing actively to its development.

5.4. Optimization

Participant: Claude Lemaréchal.

Essentially two possibilities exist to distribute our optimization software: library programs (say Modulopt codes), communicated either freely or not, depending on what they are used for, and on the other hand specific software, developed for a given application.

The following optimization codes have been developed in the framework of the former Promath project. They are generally available at http://www-rocq.inria.fr/~gilbert/modulopt/ ; M1QN3 is also distributed under GPL.

5.4.1. Code M1QN3

Optimization without constraints for problems with many variables \( (n \geq 10^3) \), has been used for \( n = 10^6 \). Technically, uses a limited-memory BFGS algorithm with Wolfe’s line-search (see Chap. 4 of [3] for the terminology).

5.4.2. Code M2QN1

Optimization with simple bound-constraints for (small) problems: \( D \) is a parallelotope in \( \mathbb{R}^n \). Uses BFGS with Wolfe’s line-search and active-set strategy.

5.4.3. Code N1CV2

Minimization without constraints of a convex nonsmooth function by a proximal bundle method (Chap. XV of [10], Chap. 9 of [3]).

5.4.4. Modulopt

In addition to codes such as above, the Modulopt library contains application problems, synthetic or from the real world. It is a field for experimentation, functioning both ways: to assess a new algorithm on a set of test-problems, or to select among several codes one best suited to a given problem.

5.5. Simulation of fibrous materials

Participants: Florence Bertails-Descoubes, Gilles Daviet.

The goal of the MECHE ADT, which started in September 2009 and was completed in fall 2011, was to develop a software for simulating the dynamics of assemblies of thin rods (such as hair), subject to contact and friction. This software combines a panel of well-accepted models for rods (ranging from reduced coordinates to maximal coordinates models, and including models recently developed by some members of the group) with classical as well as innovative schemes for solving the problem of frictional contact (incorporating the most recent results of the group, as well as the new contact solver we published in [21]). The aim of this software is twofold: first, we were able to compare and analyze the performance of nonsmooth schemes for the frictional contact problem, in terms of realism (capture of dry friction, typically), robustness, and computational efficiency. This study was conducted onto the different rod models that are available in the software. Second, we believe such a software will help us understand the behavior of a fibrous material (such as hair) through virtual experiments, thanks to which we hope to identify and understand some important emergent phenomena. A careful validation study against experiments started to be conducted in 2011 in collaboration with physicists from L’Oréal. Once this discrete elements model will be fully validated, our ultimate goal would be to build a continuous macroscopic model for the hair medium relying on nonsmooth laws.
An associate engineer, Gilles Daviet, has been hired in September 2009 to work full-time on the MECHE project. His contract was extended until October the 31st, funded by the L’Oréal collaboration project.
5. Software

5.1. T-Gems

T-GEMS is a Geometric Kernel for modeling curves and surfaces.

5.2. Ink Simulation in Maya

We have built 3D ink simulation plug-in in Autodesk Maya.
CAGIRE Team

5. Software

5.1. AeroSol

Participants: Damien Genet [Bacchus], Maxime Mogé, Francois Pellegrini [Bacchus], Vincent Perrier [correspondant].

The software AeroSol is jointly developed in the team Bacchus and the team Cagire. It is a high order finite element library written in C++. The code design has been carried for being able to perform efficient computations, with continuous and discontinuous finite elements methods on hybrid and possibly curvilinear meshes. The distribution of the unknowns is made with the software PaMPA, developed within the team Bacchus and the team Pumas. Maxime Mogé has been hired on a young engineer position (IJD) obtained in the ADT OuBa HOP for participating to the parallelization of the library, and arrived on November, 1st 2011.

Current features include

- **development environment** use of CMake for compilation, CTest for automatic tests and memory checking, lcov and gcov for code coverage reports.
- **In/Out** link with the XML library for handling with parameter files. Reader for GMSH, and writer on the VTK-ASCII legacy format.
- **Quadrature formula** up to 11th order for Lines, Quadrangles, Hexaedra, Pyramids, Prisms, up to 14th order for tetrahedron, up to 21st order for triangles.
- **Finite elements** up to fourth degree for Lagrange finite elements on lines, triangles and quadrangles.
- **Geometry** elementary geometrical functions for first order lines, triangles, quadrangles.
- **Time iteration** explicit Runge-Kutta up to fourth order, explicit Strong Stability Preserving schemes up to third order.
- **Linear Solvers** link with the external linear solver UMFPack.
- **Memory handling** discontinuous and continuous discretizations based on PaMPA for triangular and quadrangular meshes.
- **Numerical schemes** continuous Galerkin method for the Laplace problem (up to fifth order) with non consistent time iteration or with direct matrix inversion. Scalar stabilized residual distribution schemes with explicit Euler time iteration have been implemented for steady problems.
5. Software

5.1. SeLaLib

SeLaLib (the Semi-Lagrangian Library) is a library providing numerical methods for the kinetic models of plasma physics, in particular different types of Vlasov equations including the gyrokinetic model coupled to field solvers based on the Poisson, Maxwell or gyrokinetic quasi-neutraly equations. The PDEs are solved on structured mapped meshes or a collection of patches of such meshes, where the meshes are defined by a mapping for a cartesian logical grid. It is developed with an ADT and is strongly related to the INRIA large scale initiative Fusion.

One of its aims is to provide numerical building blocks for the GYSELA code developed at CEA Cadarache in collaboration with the Calvi project-team. GYSELA is used by physicists for simulating the development of turbulence in magnetic fusion plasmas in particular in view of the ITER project.
CLASSIC Project-Team (section vide)
5. Software

5.1. Bocop

Participants: Pierre Martinon [correspondant], Vincent Grélard, Frédéric Bonnans.

The Bocop project aims to develop an open-source toolbox for solving optimal control problems, with collaborations with industrial and academic partners. Optimal control (optimization of dynamical systems governed by differential equations) has numerous applications in transportation, energy, process optimization, and biology. This project is supported by INRIA in the framework of an ADT, Action de Développement Technologique, 2010-2012.

The software uses some packages from the COIN-OR library, in particular the well-known interior-point nonlinear programming solver Ipopt. It also features a user-friendly interface in Scilab.

See the web page http://www.bocop.org.

5.2. BiNoPe-HJ

Participants: Hasnaa Zidani [correspondant], Olivier Bokanowski, Nicolas Forcadel, Jun-Yi Zhao.

This project aims at developing sequential and parallel MPI/openMP C++ solvers for the approximation of Hamilton-Jacobi-Bellman (HJB) equations in a d-dimensional space. The main goal is to provide an HJB solvers that can work in dimension d (limited by the machine’s capacity). The solver outputs can be visualized with Matlab or Paraview (via VTK files).

The HJB Solver has been actively developed under a partnership between COMMANDS and the SME HPC-project in the period between December 2009 to November 2011. See also http://www.ensta-paristech.fr/~zidani/BiNoPe-HJ.

We release two versions:

- HJB-SEQUENTIAL-REF: sequential version that can run on any machine
- HJB-PARALLEL-REF: parallel version that can run only on multi-core architectures.

5.3. Shoot

Participant: Pierre Martinon [correspondant].

Shoot was designed for the resolution of optimal control problems via indirect methods (necessary conditions, Pontriagyn’s Maximum Principle). Such methods transform the original problem into finding a zero of a certain shooting function. The package offers several choices of integrators and solvers, and can handle control discontinuities. Features also include the use of variational equations to compute the Jacobian of the shooting function, as well as homotopy and grid shooting techniques for easier initialization.

See also the web page http://www.cmap.polytechnique.fr/~martinon/codes.html.
CONCHA Project-Team

5. Software

5.1. C++ library Concha

Participants: Roland Becker, Daniela Capatina, Robert Luce, David Trujillo.

The objectives of our library CONCHA are to offer a flexible and extensible software with respect to:

- Numerical methods and
- Physical models.

The aim is to have a flexible code which could easily switch between the different discretizations, in order to provide a toolbox for rapid testing of new ideas.

The software architecture is designed in such a way that a group of core developers can contribute in an efficient manner, and that independent development of different physical applications is possible. Further, in order to accelerate the integration of new members and in order to provide a basis for our educational purposes (see Section 9.3), the software proposes different entrance levels. The basic structure consists of a common block, and several special libraries which correspond to the different fields of applications described in Sections 4.1 – 4.4. Hyperbolic solvers, Low-Mach number flow solvers, DNS, and viscoelastic flows. A more detailed description of each special library may be found below. In order to coordinate the cooperative development of the library, Concha is based on the INRIA-Gforge.

5.2. User interface and python interface

Participants: Roland Becker, David Trujillo.

A graphical user interface facilitate the use of the C++-library. It has been developed by Guillaume Baty (former technical staff) in collaboration with Pierre Puiseux (associate professor at LMAP). All members of the team have been involved in the testing of the interface. The first objective is to provide an easy way of installation and to facilitate the usage. To this end we use the python language with Qt in order to take advantage of higher level libraries, which allow us to reduce development time.

![Graphical user interface: option panel (left) and process panel (right) of the install tool.](image-url)
We are confronted with heterogenous backgrounds and levels of implication of the developers and users. It seems therefore crucial to be able to respond to the different needs. Our aim is to facilitate the development of the library, and at the same time, to make it possible that our colleagues involved in physical modeling can have access to the functionality of the software with a reasonable investment of time. Two graphical user interfaces have been developed: one for the installation of the library and another one for the building and execution of projects. They are based on common database and scripts written in python. The scripts can also be launched in a shell. In Figure 5 the user interface of the install tool is shown. The option panel allows to choose the components for conditional compilation and the compilation type (debug and release).

![Figure 5. User interface of the install tool.](image)

In Figure 6 the user interface of the project tool is shown. A project consists of a number of sources files and a parameter file used by the C++ executable. The sources define classes derived from the library, which are used to specify certain data such as boundary conditions and employed finite element spaces. The parameter file contains algorithmic information and physical parameters. It is generated from a database by the python utilities.

The tools offered by this development platform are based on a python interface for the library, called pyConcha. It offers a common interface, based on a plugin-system, which allows the development of command line tools in parallel. This year the consolidation of the interface part of pyConcha has been an important task. The pyConcha library is now a framework rather than a simple interface to Concha C++ library. It allows now creation of plugins, so that each user-programmer can customize pyConcha to his own goals. Previously, two main programs working: concha-install.py to install library, and concha-project.py for (semi-)end-users. Both are now plugins of pyConcha, and can be launched by pyConcha at startup. A plugin visualization could now be developed in an independent way, and launched by pyConcha on demand.

The structure of pyConcha framework is clearly separated in various modules(layers): Command Line Interface module, Graphical User Interface module and Handlers modules, see Figure 7. A great effort has been made for internationalization of pyConcha.

5.3. Parallelization

Participants: Roland Becker, Elies Bergounioux, David Trujillo.
The parallelization of the library is done in collaboration with the INRIA-team Runtime, Marie-Christine Counilh and Olivier Aumage and with Séphanie Delage (CRI, UPPA). Elies Bergounioux worked on this topic and was financed as an IID by the ADT AMPLI. The strategy for the parallelization is based on a hybrid approach using OpenMP and MPI.

5.4. Euler equations

Participants: Roland Becker, Robert Luce, Eric Schall, David Trujillo.

Based on the library CONCHA we have develop a solver for hyperbolic PDE’s based on DGFEM. So far different standard solvers for the Euler equations such as Lax-Friedrichs, Steger-Warming, and HLL have been implemented for test problems. A typical example is the scram jet test case shown in Figure 8.

5.5. Incompressible flow solvers

Participants: Roland Becker, Daniela Capatina, Robert Luce, David Trujillo.

We have started the validation of the implementation of different finite element methods for incompressible flows at hand of standard benchmark problems as the Stokes flow around a symmetric cylinder [65] and the stationary flow around a slightly non symmetric cylinder [74], see Figure 9.

5.6. DNS

Participants: Roland Becker, David Trujillo, Elies Bergounioux.

For the direct numerical simulation of incompressible turbulent flows, we have started to develop a special solver based on structured meshes with a fast multigrid algorithm incorporating projection-like schemes. The main idea is to use non-conforming finite elements for the velocities with piecewise constant pressures, leading to a special structure of the discrete Schur complement, when an explicit treatment of the convection and diffusion term is used.
Figure 8. Computed Mach-number distribution for the Scramjet test problem.

Figure 9. Flow fields for the Stokes (above) and Navier-Stokes (below) benchmark.
5.7. Polymer flow

Participants: Roland Becker, Daniela Capatina, Julie Joie, Didier Graebling.

Based on our library CONCHA we have implemented a three-field formulation with unknowns \((u, p, \tau)\) for the two-dimensional Navier-Stokes equations, based on nonconforming finite elements. The extension to the Giesekus-model for polymers has been achieved, see Section 6.7. In the case of Newtonian flows, the extra-tensor can be eliminated in order to reduce storage and computing time. This procedure serves as a preconditioner in the general case. The aim is to provide software tools for the problems in Section Viscoelastic flows.

5.8. Validation and comparison with other CFD-software

Participants: Daniela Capatina, Didier Graebling, Julie Joie, Eric Schall.

We intend to compare computations based on CONCHA with other codes at hand of the prototypical test problems described above. This allows us to evaluate the potential of our numerical schemes concerning accuracy, computing time and other practical expects such as integration with mesh generators and post-processing. At the same time, this, unfortunately very time-consuming, benchmarking activity allows us to validate our own library. The following commercial and research tools might be considered: Aéro3d (INRIA-Smash), AVBP (CERFACS), ELSA (ONERA), Fluent (ANSYS), and OpenFOAM (OpenCfd), and Polyflow® (ANSYS). So far, we have compared our code for the Giesekus model of polymer flows with the commercial software Polyflow®, see Section 4.2.
5. Software

5.1. Simulation of viscous fluid-structure interactions

Participants: Takéo Takahashi [correspondant], Jean-François Scheid, Jérôme Lohéac.

A number of numerical codes for the simulation for fluids and fluid-structure problems has been developed by the team. These codes are mainly written in MATLAB Software with the use of C++ functions in order to improve the sparse array process of MATLAB. We have focused our attention on 3D simulations which require large CPU time resources as well as large memory storage. In order to solve the 3D Navier-Stokes equations which model the viscous fluid, we have implemented an efficient 3D Stokes sparse solver for MATLAB and a 3D characteristics method to deal with the nonlinearity of Navier-Stokes equations. This year, we have also started to unify our 2D fluid-structure codes (fluid alone, fluid with rigid bodies and fluid with fishes).

Another code has been developed in the case of self-propelled deformable object moving into viscous fluid. Our aim is to build a deformable ball which could swim in a viscous fluid. In order to do this we have started a collaboration with a team from the CRAN (Research Centre for Automatic Control). This software solves numerically 3D Stokes equations using finite elements methods. The source code is written for use with MATLAB thanks to a C++ library developed by ALICE.

- Version: v0.5
- Programming language: MATLABc++
5. Software

5.1. Package “ClustOfVar”

This R package is dedicated to cluster analysis of a set of variables. Variables can be quantitative, qualitative or a mixture of both. A new version of the package is available via the link http://cran.r-project.org/web/packages/ClustOfVar/index.html since novembre 2011. The new version improves the computational time of the "kmeansvar" function used for k-means type clustering of variables. This function is now able to deal with datasets of several thousands of variables like genomic data. The package is detailed in a paper submitted for publication [44]. It has been presented in several conferences [34], [35].

5.2. Package “PCAmixdata”

This package is dedicated to factorial analysis and rotation of quantitative data, qualitative data, or mixed data. The PCAMIX method, proposed in this package includes the ordinary principal component analysis (PCA) and multiple correspondence analysis (MCA) as special cases. Orthogonal varimax rotation of the principal components of PCAMIX is also implemented in this package. Theoretical and practical results about the new rotation algorithm available in the package is in revision for publication [45] and has been presented in [36].
DEFI Project-Team

5. Software

5.1. RODIN

**Participant:** Grégoire Allaire [correspondant].

One of the aims of the RODIN project is to develop a new shape optimization software for solid structures in the framework of the SYSTUS code developed by ESI-group. The work has just started.

5.2. FreeFem++ Toolboxes

5.2.1. Shape optimization toolbox in FreeFem++

**Participants:** Grégoire Allaire, Olivier Pantz.

We propose several FreeFem++ routines which allow the users to optimize the thickness, the geometry or the topology of elastic structures. All examples are programmed in two space dimensions. These routines have been written by G. Allaire, B. Boutin, C. Dousset, O. Pantz. A web page of this toolbox is available at [http://www.cmap.polytechnique.fr/~allaire/freefem_en.html](http://www.cmap.polytechnique.fr/~allaire/freefem_en.html).

We also have written a C++ code to solve the Hamilton Jacoby equation used in the Level-set shape optimization method. This code has been linked with FreeFem++ routines.

5.2.2. Inverse Problems for Stokes Flows

**Participants:** Armin Lechleiter [correspondant], Tobias Rienmüller.

This software solves shape reconstruction inverse problems for the Stokes-Brinkmann equations modelling porous penetrable inclusions inside a free flow. This problem is motivated by non-destructive testing in pipes and reservoirs. The factorization method is used to solve the inverse problem.

5.2.3. Inverse shape and medium problem for thin coatings

**Participant:** Nicolas Chaulet.

We developed a FreeFem++ toolbox which retrieve an obstacle and two coefficients that define a generalized impedance boundary condition form a few far field data in dimension 2. The reconstruction algorithm relies on regularized non linear optimization technique. The toolbox also contains a forward solver for the scattering of acoustic waves by obstacle on which a generalized impedance boundary condition is applied using an approximate Dirichlet-to-Neuman map to bound the computational domain.

5.2.4. Inverse shape problems for axisymmetric eddy current problems

**Participants:** Armin Lechleiter, Zixian Jiang [correspondant].

This FreeFem++ toolbox solves inverse problems for an axisymmetric eddy current model using shape optimization techniques. The underlying problem is to find inclusions in a tubular and unbounded domain. The direct scattering problems are solved using an adaptive finite element method, and Dirichlet-to-Neumann operators are used to implement the transparent boundary conditions. Based on the shape derivative of an inclusion with respect to the domain, the toolbox offers regularized iterative algorithms to solve the inverse problem.

5.2.5. Contact managements

**Participant:** Olivier Pantz.
We have developed a toolbox running under Freefem++ in order to take into account the non-intersection constraints between several deformable bodies. This code has been used to treat contacts between red blood cells in our simulations, but also between genuine non linear elastic structure. It can handle both contacts and self-contacts. Moreover, a toolbox based on the Penalization method has also been developed.

5.2.6. De-Homogenization

Participant: Olivier Pantz.

We have developed a code under Freefem++ that implements our De-Homogenization method. It has been used to solve the compliance minimization problem of the compliance of an elastic shape. In particular, it enables us to recover well known optimal Michell’s trusses for shapes of low density.

5.3. Scilab and Matlab Toolboxes

5.3.1. Shape optimization toolbox in Scilab

Participant: Grégoire Allaire [correspondant].

Together with Georgios Michailidis, we improved a Scilab toolbox for 2-d shape and topology optimization by the level set method which was originally produced by Anton Karrman and myself. The routines, a short user’s manual and several examples are available on the web page.

5.3.2. Conformal mapping method

Participant: Houssem Haddar [correspondant].

This Scilab toolbox is dedicated to the resolution of inverse 2-D electrostatic problems using the conformal mapping method introduced by Akdumann, Kress and Haddar. The toolbox treats the cases of a simply connected obstacle with Dirichlet, Neumann or impedance boundary conditions or a simply connected inclusion with a constant conductivity.

5.3.3. Direct and inverse problems in waveguides

Participants: Armin Lechleiter [correspondant], Dinh Liem Nguyen.

This Matlab toolbox includes fast solvers for direct and inverse scattering problems in planar 3D waveguides for inhomogeneous media The direct scattering problems are solved using an spectral integral equation approach relying on the Lippmann-Schwinger integral equation, discretized as a Galerkin method via the fast Fourier transform. The toolbox includes preconditioning by a two-grid scheme and multipole expansions coupled to the spectral solver to allow for multiple scattering objects. The inverse problem to find the shape of the scattering object from near-field measurements is solved using a Factorization method.

5.4. Sampling methods for inverse problems

5.4.1. Samplings-2d

Participant: Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 2-D. It includes three independent components. The first one solves to scattering problem using integral equation approach and supports piecewise-constant dielectrics and obstacles with impedance boundary conditions. The second one contains various samplings methods to solve the inverse scattering problem (LSM, RGLSM(s), Factorization, MuSiC) for near-field or far-field setting. The third component is a set of post processing functionalities to visualize the results.

- License: GPL
- Type of human computer interaction: sourceforge
- OS/Middleware: Linux
- Programming language: Fortran
- Documentation: fichier

### 5.4.2. Samplings-3d

**Participant:** Houssem Haddar [correspondant].

This software is written in Fortran 90 and is related to forward and inverse problems for the Helmholtz equation in 3-D. It contains equivalent functionalities to samplings-2d in a 3-D setting.

### 5.4.3. Time domain samplings-2d

**Participants:** Houssem Haddar [correspondant], Armin Lechleiter.

This software is written in Fortran 90 and is related to forward and inverse problems for the time dependent wave equation in 2-D. The forward solver is based on a FDTD method with PMLs. The inverse part is an implementation of the linear sampling method in a near field setting and the factorization method in a far field setting.

### 5.4.4. Factorization Method for EIT

**Participant:** Giovanni Migliorati.

We developed a numerical code that implements the Factorization Method applied to the Continuous Model, in the framework of Electrical Impedance Tomography featuring an inhomogeneous background. The numerical scheme relies on the approximation by the finite element method of the solution to the dipole-like Neumann boundary-value problem. Two regularization techniques are implemented, i.e. the Tikhonov regularization embedding Morozov principle, and the classical Picard Criterion. The numerical analysis of the method and the results obtained are presented in the INRIA RR-7801, November 2011.

### 5.5. FVforBlochTorrey

**Participant:** Jing-Rebecca Li [correspondant].

Finite volume code in Fortran 90 to solve the multiple compartment Bloch Torrey equation in 2D and 3D to simulate the bulk magnetization of a sample under the influence of a diffusion gradient. We couple a mass-conserving finite volume discretization in space with a stable time discretization using an explicit Runge-Kutta-Chebyshev method and we are able to solve the Bloch-Torrey PDE in multiple compartments for an arbitrary diffusion sequence with reasonable accuracy for moderately complicated geometries in computational time that is on the order of tens of minutes per bvalue on a laptop computer. See also the web page [http://www.cmap.polytechnique.fr/~jingrebecca/](http://www.cmap.polytechnique.fr/~jingrebecca/).
5. Software

5.1. YALTA

Participants: David Avanessoff [correspondent], Catherine Bonnet, André Fioravanti [UNICAMP, correspondent].

The YALTA package is dedicated to the study of classical and fractional systems with delay in the frequency-domain. Its objective is to provide basic but important information such as, for instance, the position of the neutral chains of poles and unstable poles, as well as the root locus with respect to the delay of the system. The corresponding algorithms are based on recent theoretical results (see, for instance, [14] and [112]) and on classical continuation methods exploiting the particularities of the problem [113], [18]. Some refinements have been included this year in order to deal with systems with tricky numerical behaviour. The YALTA package will be available in the first semester of 2012.

5.2. OreModules

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen], Frédéric Chyzak [INRIA Rocquencourt, Algorithms Project].

The OreModules package [106], based on the commercial Maple package Ore_algebra [107], is dedicated to the study of linear multidimensional systems defined over certain Ore algebras of functional operators (e.g., ordinary or partial differential systems, time-delay systems, discrete systems) and their applications in mathematical systems theory, control theory and mathematical physics. OreModules is original because it combines the recent developments of the Gröbner bases over some noncommutative polynomial rings [115], [117] and new algorithms of algebraic analysis in order to effectively check classical properties of module theory (e.g., existence of a non-trivial torsion submodule, torsion-freeness, reflexiveness, projectiveness, stably freeness, freeness), it gives their system-theoretical interpretations (existence of autonomous elements or successive parametrizations, existence of minimal/injective parametrizations or Bézout equations) [121], [120], [105] and it computes important tools of homological algebra (e.g., (minimal) free resolutions, split exact sequences, extension functors, projective or Krull dimensions, Hilbert power series). The abstract language of homological algebra used in the algebraic analysis approach carries over to the implementations in OreModules: up to the choice of the domain of functional operators which occurs in a given system, all algorithms are stated and implemented in sufficient generality such that linear systems defined over the Ore algebras developed in the Ore_algebra package are covered at the same time. Applications of the OreModules package to mathematical systems theory, control theory and mathematical physics are illustrated in a large library of examples. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.

5.3. Stafford

Participants: Alban Quadrat [correspondent], Daniel Robertz [Univ. Aachen].

The Stafford package of OreModules [106] contains an implementation of two constructive versions of Stafford’s famous but difficult theorem [132] stating that every ideal over the Weyl algebra $A_n(k)$ (resp., $B_n(k)$) of partial differential operators with polynomial (resp., rational) coefficients over a field $k$ of characteristic 0 (e.g., $k = \mathbb{Q}, \mathbb{R}$) can be generated by two generators. Based on this implementation and algorithmic results developed in [128] by the authors of the package, two algorithms which compute bases of free modules over the Weyl algebras $A_n(\mathbb{Q})$ and $B_n(\mathbb{Q})$ have been implemented. The development of the Stafford package was motivated by the problem of computing injective parametrizations of underdetermined linear systems of partial differential equations with polynomial or rational coefficients (the so-called Monge problem), differential flatness, the reduction and decomposition problems and Serre’s reduction problem. To our knowledge, the Stafford package is the only implementation of Stafford’s theorems nowadays available. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/OreModules/.
5.4. QuillenSuslin

Participants: Alban Quadrat [correspondent], Anna Fabińska [Univ. Aachen].

The QUILLEN-SUSLIN package [110] contains an implementation of the famous Quillen-Suslin theorem [130], [133]. In particular, this implementation allows us to compute bases of free modules over a commutative polynomial ring with coefficients in a field (mainly $\mathbb{Q}$) and in a principal ideal domain (mainly $\mathbb{Z}$). The development of the QUILLEN-SUSLIN package was motivated by different constructive applications of the Quillen-Suslin theorem in multidimensional systems theory [110] (e.g., the Lin-Bose conjectures, the computation of (weakly) left/right/doubly coprime factorizations of rational transfer matrices, the computation of injective parametrizations of flat linear multidimensional systems with constant coefficients, the reduction and decomposition problems, Serre’s reduction problem). To our knowledge, the QUILLEN-SUSLIN package is the only implementation of the Quillen-Suslin theorem nowadays available. The binary of the package is freely available at http://wwwb.math.rwth-aachen.de/QuillenSuslin.

5.5. OreMorphisms

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges].

The OREMORPHISMS package [109] of OREMODULES [105] is dedicated to the implementation of homological algebraic tools such as the computations of homomorphisms between two finitely presented modules over certain noncommutative polynomial algebras (Ore algebras), of kernel, cokernel, image and endomorphisms, Galois transformations of linear multidimensional systems and idempotents of endomorphism rings. Using the packages STAFFORD and QUILLEN-SUSLIN, the factorization, reduction and decomposition problems can be constructively studied for different classes of linear multidimensional systems. Many linear systems studied in engineering sciences, mathematical physics and control theory have been factorized, reduced and decomposed by means of the OREMORPHISMS package. The binary of the package is freely available at http://www-sop.inria.fr/members/Alban.Quadrat/OreMorphisms/index.html.

5.6. JanetMorphisms

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges], Daniel Robertz [Univ. Aachen].

The JANETMORPHISMS package is dedicated to a new mathematic approach to quasilinear systems of partial differential equations (e.g., Burger’s equation, shallow water equations, Euler equations of a compressible fluid) based on algebraic analysis and differential algebra techniques [44]. See Section 6.2. This package computes symmetries, first integrals of motion, conservation laws, study Riemann invariants... The JANETMORPHISMS package is based on the Janet package (http://wwwb.math.rwth-aachen.de/Janet/).

5.7. PurityFiltration

Participant: Alban Quadrat [correspondent].

The PURITYFILTRATION package, built upon the OREMODULES package, is an implementation of a new effective algorithm obtained in [97], [77], [76] which computes the purity/grade filtration [102], [103] of linear functional systems (e.g., partial differential systems, differential time-delay systems, difference systems) and equivalent block-triangular matrices. See Section 6.1. This package is used to compute closed form solutions of over/underdetermined linear partial differential systems which cannot be integrated by the standard computer algebra systems such as Maple and Mathematica. This package will soon be available.

5.8. AbelianSystems

Participants: Alban Quadrat [correspondent], Mohamed Barakat [Univ. Kaiserslautern].
The ABELIANSYSTEMS package is an implementation of an algorithm developed in [97], [77], [76] for the computation of the purity/grade filtration [102], [103] in the powerful homalg package of GAP 4 dedicated to constructive homological algebra methods, and developed by Barakat (University of Kaiserslautern) and his collaborators (http://homalg.math.rwth-aachen.de/). This package both supersedes the existing PURITYFILTRATION package which uses the non-efficient Maple Gröbner basis computation (see Section 5.7), and the original homalg procedure which computes purity filtration by means of time-consuming spectral sequences. Using the homalg package philosophy, the ABELIANSYSTEMS package can be used for the computation of the purity filtration of objects in different constructive abelian categories such as sheaves over projective varieties as demonstrated in the homalg package called Sheaves (see http://homalg.math.rwth-aachen.de/).

5.9. SystemTheory

Participants: Alban Quadrat [correspondent], Thomas Cluzeau [ENSIL, Univ. Limoges], Markus Lange-Hegermann [Univ. Aachen], Mohamed Barakat [Univ. Kaiserslautern].

The SYSTEMTHEORY package is a homalg based package dedicated to mathematical systems. This package, still in development, will include the algorithms developed in the OREMODULES and ORE MORPHISMS packages. It currently contains an implementation of the ORE MORPHISMS procedures which handle the decomposition problem aiming at decomposing a module/system into direct sums of submodules/subsystems.
5. Software

5.1. ParadisEO

**Participants:** Karima Boufaras, Laetitia Jourdan, Arnaud Liefooghe, Thé Van Luong, Nouredine Melab, El-Ghazali Talbi [correspondent], Sébastien Verel.

ParadisEO (PARallel and DIStributed Evolving Objects) is a C++ white-box object-oriented framework dedicated to the flexible design of metaheuristics. See web pages [http://paradiseo.gforge.inria.fr/](http://paradiseo.gforge.inria.fr/). Based on EO, a template-based ANSI-C++ compliant evolutionary computation library, it is composed of four modules:

- Paradiseo-E0 provides tools for the development of population-based metaheuristic (evolutionary and genetic algorithm, genetic programming, particle swarm optimization, etc.)
- Paradiseo-M0 provides tools for the development of single solution-based metaheuristics (hill-climbing, tabu search, simulated annealing, iterative local search, variable neighborhood search, incremental evaluation, partial neighborhood, etc.)
- Paradiseo-MOEO provides tools for the design of multi-objective metaheuristics (MO fitness assignment, MO diversity preservation, elitism, performance indicators, easy-to-use state-of-the-art algorithms, etc)
- Paradiseo-PEO provides tools for the design of parallel and distributed metaheuristics (parallel evaluation, parallel evaluation function, island model)

Furthermore, Paradiseo also introduces tools for the design of distributed, hybrid and cooperative models:

- High level hybrid metaheuristics: coevolutionary and relay models.
- Low level hybrid metaheuristics: coevolutionary and relay models.

The ParadisEO framework has been especially designed to best suit the following objectives:

- Maximum design and code reuse: ParadisEO is based on a clear conceptual separation of the solution methods from the problems they are intended to solve. This separation confers to the user a maximum code and design reuse.
- Flexibility and adaptability: The fine-grained nature of the classes provided by the framework allows a higher flexibility compared to other frameworks.
- Utility: ParadisEO allows the user to cover a broad range of metaheuristics, problems, parallel distributed models, hybridization mechanisms, etc.
- Transparent and easy access to performance and robustness: As the optimization applications are often time-consuming the performance issue is crucial. Parallelism and distribution are two important ways to achieve high performance execution. ParadisEO is one of the rare frameworks that provide the most common parallel and distributed models. These models can be exploited in a transparent way, one has just to instantiate their associated provided classes.
- Portability: The implemented models are portable on distributed-memory machines as well as on shared-memory multiprocessors, as they use standard libraries such as MPI and PThreads.

This year, with the aim of reinforcing ParadisEO, much works has been established:

- A new design and implementation of the Paradiseo-M0 module.
- The addition of local search algorithms for multiobjective optimization.
- The addition of a new module dedicated to parallel metaheuristics on graphics cards.
All the new features is managed via the INRIA’s Gforge project http://paradiseo.gforge.inria.fr.

5.1.1. Paradiseo-MO: a new design and fitness landscape

In the previous version of ParadisEO-MO, each local search algorithm was implemented as a whole, with only a small number of components shared with the others. Moreover, there was no component to trace statistics on local search execution, and no way to implement them easily, in opposition to the ParadisEO philosophy. A new design and implementation of the ParadisEO-MO module has been achieved, allowing one to tackle an optimization problem as a whole, from its analysis to its resolution. In comparison to the previous version of the framework, the modularity has been largely improved, together with an easier reuse of basic components. Another brand new feature of the ParadisEO-MO software framework relates to sampling and statistical tools for fitness landscape analysis.

The new design is based on a clear conceptual separation of the solution methods from the problems they are intended to solve, new concepts are proposed:

- **Neighbor**: Moves and saves neighbor informations (fitness and more)
- **Neighborhood**: Describes how to compute all the neighbors
- **Evaluation**: Can be incremental or full evaluation

Different features are included to improve the set of ParadisEO-MO modular classes combined to develop single solution based metaheuristics:

- **General scheme of Local Search algorithms (LS)**
- **List of Local search algorithms**:
  - Hill-climbing (4 different methods)
  - Random Walk (3 different methods)
  - Metropolis Hasting
  - Simulated annealing
  - Tabu search
  - Iterated local search
  - Variable neighborhood search
- **New tools to perform fitness landscapes analysis**:
  - Density of states
  - Fitness distance correlation
  - Autocorrelation length and function
  - Sampling the local optima by adaptive walks
  - Neutral degree distribution
  - Evolvability of neutral networks by neutral walks
  - Fitness cloud
- **New stopping criteria and control method have been added**
- **Predefined neighborhood operators for standard problem representations**
5.1.2. Paradiseo-MOEO and multiobjective local search

This year, we particularly improved the module dedicated to multiobjective optimization in terms of local search metaheuristics. As a first step, we focused on a subclass of pure neighborhood search methods. These algorithms can be seen as a generalization of the most basic local search procedure for the multiobjective case. Generally speaking, they combine the definition of a neighborhood structure with the management of a population (or archive) of potentially efficient solutions according to a dominance relation. This archive is iteratively improved by exploring the neighborhood of its own content until no further improvement is possible, or until a stopping condition is satisfied. We denoted them as *Dominance-based Multiobjective Local Search* (DMLS). We also started to implement scalar (preference-based) solution-based local search approaches that should be incorporated in the next version of the platform.

Additionally, some hybridization approaches based on the relay mode have been proposed to hybridize easily evolutionary algorithms with local search during mutation or checkpointing in a multiobjective context. At last, archiving good-quality solutions during the execution of the algorithm is often a large part of the execution time. Thus it was important to provide advanced techniques to reduce this cost. Several solutions proposed in the literature have been implemented. All these new components have been tested and documented.

5.1.3. ParadisEO-GPU

We proposed a pioneering framework called ParadisEO-GPU for the reusable design and implementation of parallel local search metaheuristics (S-Metaheuristics) on Graphics Processing Units (GPU). We have first revisited the ParadisEO-MO software framework to allow its utilization on GPU accelerators focusing on the parallel iteration-level model, the major parallel model for S-Metaheuristics. It consists in the parallel exploration of the neighborhood of a problem solution.

The challenge is on the one hand to rethink the design and implementation of this model optimizing the data transfer between the CPU and the GPU. On the other hand, the objective is to make the GPU as transparent as possible for the user minimizing his or her involvement in its management. From a design point of view, we proposed solutions to this challenge as an extension of the ParadisEO framework. Indeed, a conceptual effort has been done to take into account the aspects related to the GPU architecture and to the ParadisEO-MO module. It has allowed to identify the generic components that are transparent to the user: memory allocation/desallocation on GPU, data transfer between the CPU and the GPU, parallel evaluation of the neighborhood on GPU, structures for the neighborhood evaluation on CPU/GPU, etc.

The first release of the new GPU-based ParadisEO framework has been experimented on the permuted perceptron problem and on the quadratic assignment problem. The preliminary results are convincing, both in terms of flexibility and easiness of reuse at implementation, and in terms of efficiency at execution on GPU.

5.1.4. New technical features

Regarding the technical aspects, the compatibility with dependencies taken into account is:

- Checked compatibility with different operating systems
- Reviewed and checked compatibility with new versions of the tools used (Cmake, g++, Visual Studio...)
- Unit test of all additional components, and experiments on classical applications
- A new Website design, with a rearrangement of information based on a set of collected statistics

5.1.5. Contributions and documentations

Many investigations were made in this context in order to help users to manipulate the framework.

New documentation:

- The API documentation is available on the ParadisEO Website
- New tutorials
– Hill Climbing
– Neighborhoods (classical and indexed)
– Simulated Annealing and Checkpointing
– Tabu Search
– Iterated Local Search
– Fitness Landscapes Analysis
– Hybrid Lesson

Moreover, a set of implementations for classical problems are now provided as contributions, available within the new version of ParadisEO:

- Single-objective problems: oneMax, queen, quadratic assignment problem, royal road, long path, building block royal road, NK landscapes, NKq landscapes, NKp landscapes, MAX-SAT, unconstrained binary quadratic programming problem, and more.
- Multi-objective problems: traveling salesman problem, quadratic assignment problem, multiple and long path problems.
GAMMA3 Project-Team (section vide)
GECO Team (section vide)
5. Software

5.1. Software

Participants: Hussein Yahia [correspondant], Antonio Turiel, Joel Sudre.

FluidExponents: software implementation of the MMF, written in Java, in a cooperative development mode on the INRIA GForge, deposited at APP in 2010. Contact: hussein.yahia@inria.fr. FluidExponents implements nonlinear signal processing on various types of input data (including NETCDF).

- Version: 0.8
5. Software

5.1. COSMAD

With the help of former engineers, I4S team has developed and maintained a Scilab toolbox devoted to modal analysis and vibration monitoring of structures or machines subjected to known or ambient (unknown) excitation. This software (COSMAD 3.64) has been registered at the APP under the number

IDDN.FR.001.210011.002.S.A.2003.000.20700

A list of test-cases (simulators, laboratory test-beds, real structures) for which COSMAD has been used is available on I4S website. The problem is to identify the eigenstructure (eigenvalues and observed components of the associated eigenvectors) of the state transition matrix of a linear dynamical system, using only the observation of some measured outputs summarized into a sequence of covariance matrices corresponding to successive time shifts. Other services are

- Output-only and Input/Output subspace-based identification,
- Automated on-line identification package,
- Subspace-based identification through moving sensors data fusion,
- Damage detection and monitoring,
- Damage localization,

The modules have been tested by different partners, especially the French industrial partners, EADS, Dassault and Sopema, within the FLITE2 project, by partners from the past CONSTRUCTIF project, and within the framework of bilateral contracts with SNECMA and SVS.

Based on intensive internal evaluation of the toolbox, on both simulated and real data sets, EADS Space Transportation and CNES have been investigating how to use the toolbox for the exploitation of the Ariane 5 flight data sets.

This Scilab toolbox continues to play the role of a programming and development environment for all our newly designed algorithms. Moreover, offering a maintained Scilab platform turns out to be a crucial factor in convincing industrial partners to undertake joint investigations with us. Just recently, SNECMA funded development for the Cosmad toolbox in 2010.

5.2. Prototypes

Three software have been deposed to the Agency of Program Protection, i.e.

- 1/ Fast multi-order Stochastic Subspace Identification (FMO-SSI) IDDN.FR.001.100017.000.S.P.2011.000.20700
- 2/ Multi-setup Stochastic Subspace Identification (MS-SI) IDDN.FR.001.100016.000.S.P.2011.000.20700
- 3/ Multi-order confidence interval computation for single-setup and multi-setup Stochastic Subspace Identification (MOCI-SSI) IDDN.FR.001.100018.000.S.P.2011.000.20700

They will be transferred to partners and industrial contracts, starting in 2011.
IPSO Project-Team (section vide)
4. Software

4.1. PREMIA

Participants: Antonino Zanette, Mathfi Research team, Agnès Sulem [correspondant].

Premia is a software designed for option pricing, hedging and financial model calibration. It is provided with its C/C++ source code and an extensive scientific documentation. https://www-rocq.inria.fr/mathfi/Premia

This Premia project keeps track of the most recent advances in the field of computational finance in a well-documented way. It focuses on the implementation of numerical analysis techniques for both probabilistic and deterministic numerical methods. An important feature of the platform Premia is the detailed documentation which provides extended references in option pricing.

Premia is thus a powerful tool to assist Research & Development professional teams in their day-to-day duty. It is also a useful support for academics who wish to perform tests on new algorithms or pricing methods without starting from scratch.

Besides being a single entry point for accessible overviews and basic implementations of various numerical methods, the aim of the Premia project is:

1. to be a powerful testing platform for comparing different numerical methods between each other;
2. to build a link between professional financial teams and academic researchers;
3. to provide a useful teaching support for Master and PhD students in mathematical finance.

- AMS: 91B28;65Cxx;65Fxx;65Lxx;65Pxx
- License: Licence Propriétaire (genuin license for the Consortium Premia)
- Type of human computer interaction: Console, interface in Nsp
- OS/Middleware: Linux, Mac OS X, Windows
- APP: The development of Premia started in 1999 and 13 are released up to now and registered at the APP agency.
- Programming language: C/C++ librairie Gtk
- Documentation: the PNL library is interfaced via doxygen
- Size of the software: 250 Mbyte, 40 Mbyte of C/C++ routines; Number of lines of code: 972000 (for the source part only)
- Publications: [1] [71] [78] [86] [88], [62]

4.1.1. Content of Premia

Premia contains various numerical algorithms (Finite-differences, trees and Monte-Carlo) for pricing vanilla and exotic options on equities, interest rate, credit and energy derivatives.

1. Equity derivatives:

The following models are considered:

- Black-Scholes model (up to dimension 10), stochastic volatility models (Hull-White, Heston, Fouque-Papanicolaou-Sircar), models with jumps (Merton, Kou, Tempered stable processes, Variance gamma, Normal inverse Gaussian), Bates model.
For high dimensional American options, Premia provides the most recent Monte-Carlo algorithms: Longstaff-Schwartz, Barraquand-Martineau, Tsitsklis-Van Roy, Broadie-Glassermann, quantization methods and Malliavin calculus based methods.

Dynamic Hedging for Black-Scholes and jump models is available.

Calibration algorithms for some models with jumps, local volatility and stochastic volatility are implemented.

2. Interest rate derivatives

The following models are considered:


Premia provides a calibration toolbox for Libor Market model using a database of swaptions and caps implied volatilities.

3. Credit derivatives: CDS, CDO

Reduced form models and copula models are considered.

Premia provides a toolbox for pricing CDOs using the most recent algorithms (Hull-White, Laurent-Gregory, El Karoui-Jiao, Yang-Zhang, Schönbucher)

4. Hybrid products

PDE solver for pricing derivatives on hybrid products like options on inflation and interest or change rates is implemented.

5. Energy derivatives: swing options

Mean reverting and jump models are considered.

Premia provides a toolbox for pricing swing options using finite differences, Monte-Carlo Malliavin-based approach and quantization algorithms.

4.1.2. Development of the PNL Library

Here are the major contribution of J. Lelong:

1. Development of the PNL.
   - A PnlArray object has been added to create arrays of PnlObjects.
   - The implementation of the PnlList type has been changed to improve linear iteration on a list.
   - Design of a new unit test framework and backport of all the previous tests.
   - Integration of the Runge Kutta Fehler 45 method for solving n dimensional ODEs.
   - New organisation of the manual.
   - Random number generators: new Sobol generators (32 and 64 bits), dynamic Mersenne Twister updated to version 0.6.1. All random number generators are now thread-safe.
   - Update of internal Lapack (and the corresponding shipped version of Blas) to version 3.2.1
   - Update of F2C.
   - LU factorization for tridiagonal matrices.
   - Cholesky block factorization for positive semi-definite matrices.

4.1.3. Premia design

Anton Kolotaev (ADT engineer), supervised by J. Lelong, has realized the Bindings for Premia with Python and F#; a library allowing the development of bindings to other languages; a Web interface Web for Premia.
Moreover he has improved the documentation facility system of Premia and has updated the Excel interface Excel for the new versions MS Excel and MS Windows.

Tasks achieved by J. Lelong:

1. New design of the enumeration type to allow the number of parameters of Premia objects to depend on the selected value within the enumeration. This change had a strong impact on the VAR system and many core functions had to be rewritten. This modification was definitely essential to improve the Nsp interface, which was broken for long as far as credit derivatives are concerned and is now working properly again.

2. Creation of the first Premia bundle for Mac OS X and automation of the building process.

3. Generic “Get”, “FGet” and “Print” functions had been introduced to simplify object creation but there were still many exceptions which were not using these generic functions. All these exceptions have been handled and now rely on the generic functions.

4. Improvement of the scripts to build the free version of Premia.

5. Integration of Cosine methods implemented by Bowen Zhang.

Tasks achieved by C. Labart:

- Improvement of the credit part of Premia: correction of memory leaks, modification of old codes to use the new copula structure. This enables to remove a large part of the code which has become hard to maintain.
- Complete rewriting of the BSDE algorithm for pricing basket options in high dimension to use a more efficient approximation technique. This has been possible thanks to the polynomial approximation tool provided by the PNL.

4.1.4. Algorithms implemented in Premia in 2011

Premia 13 was delivered to the consortium members in March 2011. It contains the following new algorithms:

4.1.4.1. Interest rate derivatives

- Pricing and hedging callable Libor exotics in forward Libor models. V. Piterbarg, *The Journal of Computational Finance* Volume 8 Number 2, Winter 2004/05


4.1.4.2. Credit risk derivatives


4.1.4.3. Electricity derivatives

- Variance optimal hedging for processes with independent increments and applications. Applications to electricity market. F.Russo, S. Goutte and N. Oudjane. *Preprint 2010*
4.1.4.4. Equity derivatives

- Pricing options under stochastic volatility: a power series approach. F. Antonelli and S. Scarlatti *Finance & Stochastics*, Volume XIII (2009), issue 1
- Saddlepoint methods for option pricing Peter Carr and Dilip Madan *The Journal of Computational Finance*, to appear
- Monte Carlo for pricing Asian options in jump models. E. Dia and D. Lamberton, *Preprint*
- Doubly Reflected BSDEs with Call Protection and their Approximation. J.F. Chassagneux S.Crepey. *Preprint* 2010

4.1.5. New algorithms for the release 14 of Premia to be delivered in March 2012 to the Consortium:

4.1.5.1. Interest rate derivatives


4.1.5.2. Credit risk derivatives

• Calibration in a local and stochastic intensity model. A. Alfonsi, C. Labart and J. Lelong, *Preprint*.

4.1.5.3. Energy and commodities

• Markov Models for Commodity Futures: Theory and Practice. L. Andersen. *Preprint*

4.1.5.4. Equity derivatives

• S.Ould-Aly Revised Bergomi model. *Preprint*


• Volatility derivatives in market models with jumps. H. Lo A. Mijatovic, *Preprint*.


• High-order discretization for stochastic correlation models. A. Alfonsi and A. Ahdida *Preprint*


• Ninomiya-Victoir scheme for variance swaps model.

• Multilevel adaptive Monte Carlo. A. Kebaier, K. Hajji


• Wiener-Hopf techniques for Path-dependent options in Bates and Heston model. Kudryavtsev O.


• Continuously monitored barrier options under Markov processes.

• Exotic derivatives in a dense class of stochastic volatility models with jumps. A. Mijatovic and M. Pistorius, preprint.

• Pricing Discretely Monitored Asian Options by Maturity Randomization. G.Fusai, D. Marazzina and M. Marena. *Preprint*

• B. Lapeyre and A. Abbas-Turki American Options Based on Malliavin Calculus and nonparametric Variance Reduction Methods on Malliavin Calculus. *Preprint*.


• Backward Stochastic Differential Equations (BSDE).

• American options in high dimension solving BSDE with penalization.
MAXPLUS Project-Team

5. Software

5.1. Boîte à outil Maxplus de SCILAB/Maxplus toolbox of Scilab

Trois chercheurs du groupe (S. Gaubert, J.-P. Quadrat, et G. Cohen) ont développé (à partir d’une première version réalisée par M. Mc Gettrick) la boîte à outils Maxplus de Scilab, qui est téléchargeable librement parmi les contributions du site Scilab, et qui est maintenant intégrée par défaut dans Scicoslab. Cette boîte à outils implémente l’ensemble du calcul numérique linéaire max-plus, elle comprend en particulier le stockage creux des matrices, et des algorithmes efficaces pour le calcul de la valeur propre basées sur les itérations sur les politiques. Elle a été utilisées par plusieurs chercheurs, voir notamment [84], [146]. Il faut aussi noter que le groupe de L. Hardouin, du LISA/Istia, a complété la boîte à outils Maxplus en interfaçant leur propre librairie C++, qui permet le calcul des séries de transfert de graphes d’événements temporisés.

English version

Three researchers of the team (S. Gaubert, J.-P. Quadrat, and G. Cohen, building on a preliminary version of M. McGettrick) have developed and released the Maxplus toolbox of Scilab, which is freely available among the contributions on the Scilab web site, and which is now included by default in Scicoslab. It implements all basic linear algebra functionalities, with a special attention to large sparse matrices, including efficient algorithms for eigenvalue computation based on policy iteration. The software has been used by several researchers in their work, including [84], [146]. It should be noted that the team of L. Hardouin, from LISA/Istia, has completed the toolbox by interfacing their own C++ library computing the transfer series of a timed event graph.

5.2. Itérations sur les politiques pour les jeux stochastiques à somme nulle/Policy iterations for zero sum stochastic games

L'algorithme d’itérations sur les politiques pour les jeux stochastiques à somme nulle pour le cas de paiements ergodiques (gain moyen par unité de temps), et dégénérés de type “multi-chaîne” a été introduit dans [103]. Plusieurs stages ont permis l’implémentation partielle en Scilab, C ou C++, et le test de ce type d’algorithmes (voir le travail de Vishesh Dhingra [115]), ou de son couplage avec la résolution de systèmes linéaires par des méthodes multigrilles algébriques (stage de Shantanu Gangal en 2007). Le travail de thèse de Sylvie Detournay, qui porte sur le couplage entre itérations sur les politiques et méthodes multigrilles algébriques, voir le § 6.4.1 ci-dessous, a permis le développement d’un programme complet. Le code écrit par Sylvie Detournay (en C) a été déposé sur INRIAGForge. Pour le moment il n’est accessible qu’aux membres de l’équipe.

English version

The policy iteration algorithm for zero sum repeated games with ergodic payoff (i.e. mean payoff per time unit), and in degenerate “multichain” cases, has been introduced in [103]. Several internships allowed us to implement in Scilab, C or C++, and to test such algorithms (see the work of Vishesh Dhingra [115]), or its combination with the resolution of linear systems by algebraic multigrid methods (internship of Shantanu Gangal in 2007). The PhD thesis work of Sylvie Detournay, who concerns the combination of policy iterations with algebraic multigrid methods, see § 6.4.1 below, allowed us to develop a complete program. The program written by Sylvie Detournay (in C language) has been posted on INRIAGForge. For the moment it can only be seen by members of the team.
5.3. TPLib: bibliothèque pour la manipulation de polyèdres tropicaux/TPLib: tropical polyhedra library

TPLib est une bibliothèque écrite en OCaml qui permet de manipuler des polyèdres tropicaux. Elle est distribuée sous licence LGPL https://gforge.inria.fr/projects/tplib .

Cette bibliothèque implémente notamment des algorithmes permettant de passer d’une représentation externe d’un polyèdre à une représentation interne, ou inversement (voir § 6.2.1 pour plus de détails). Elle fournit également toutes les primitives permettant d’utiliser les polyèdres tropicaux en tant que domaine abstrait numérique, afin de déterminer des invariants de programmes faisant intervenir les opérations min et max (voir [ 82 ]).

TPLib est aujourd’hui utilisé dans le logiciel Polymake [ 128 ], développé à la Technische Universität Darmstadt (Allemagne). Ce dernier logiciel constitue une boîte à outils permettant de manipuler des nombreux objets mathématiques (polytopes convexes, complexes polyédraux, graphes, matroïdes, polytopes tropicaux). Une interface à la bibliothèque de domaines abstraits numériques APRON [ 139 ] est également en cours de développement.

English version

TPLib is a library written in OCaml, which allows to manipulate tropical polyhedra. It is distributed under LGPL https://gforge.inria.fr/projects/tplib .

This library implements algorithms allowing to pass from an external representation of a polyhedron to an internal description, or inversely (see § 6.2.1 for more details). It also provides all the primitives allowing to use tropical polyhedra as an numerical abstract domain, in order to determine program invariants involving the operations min and max (see [ 82 ]).

TPLib is now used in the software Polymake [ 128 ], developed in Technische Universität Darmstadt (Germany). Polymake is a toolbox allowing to manipulate mathematic objects such as convex polytopes, polyhedral complexes, graphs, matroids, and tropical polytopes. An interface to the numerical abstract domain APRON [ 139 ] is also under development.
5. Software

5.1. eLYSe

Participants: Olivier Saut [correspondant], Raphael Bahegne, Vincent Hubert, Jean-Baptiste Lagaert, Mathieu Specklin.

eLYse is a numerical platform used for our computations in Biology (tumor growth), micro-fluidics and complex Newtonian fluid flows. The platform is divided in two libraries: one is devoted to the modelling equations and the other one includes the numerical solvers. For example, we are able to treat (in 2D and 3D) transport equations, diffusion equations, Navier-Stokes equations, Maxwell system and the interaction fluid-structure by level-set and penalization methods. The solvers are based on finite volume methods on cartesian grids and allow parallel computations. See also the web page http://www.math.u-bordeaux1.fr/~saut/wp/?page_id=201.

- Version: 0.4
- ACM: ACM J.2 J.3 G.1.8 G.1.10
- AMS: AMS65Z05 35Q92
- Keywords: Modélization and numerical simulations, Finite volume methods, Level Set approach, Penalization method
- APP: En cours
- Type of human computer interaction: console
- OS/Middelware: Platform developed on Mac OS X architecture.
- Programming language: C++
- Documentation: doxygen.

5.2. Kesaco

Participants: Olivier Saut [correspondant], Raphaël Bahègne, Damiano Lombardi, Mathieu Specklin.

Kesaco is a set of libraries and programs aiming at applications of mathematical modeling in clinical oncology. It features:

- A library of specialized mathematical model describing the growth of different types of cancers (secondary tumors in the lung, gliomas).
- A set of programs useful to validate mathematical models (compute the various behavior they can produce) and to build databases of numerical simulations.
- Segmentation and registration routines to use medical images directly in our numerical codes.
- Calibration methods to recover the parameters of the models using sequences of medical images. Three techniques are implemented (a genetic algorithm, a technique based on reduced order models, a sensitivity technique).
All these routines are adapted to run on a MP architecture. The webpage may be found at http://www.math.u-bordeaux1.fr/~saut/wp/?page_id=345.

- Version: 0.1
- Keywords: Modélization and numerical simulations
- APP: En cours
- Type of human computer interaction: console
- OS/Middleware: Platform developed on Mac OS X architecture.
- Required library or software: eLYSe, Insight Toolkit (http://www.itk.org)
- Programming language: C++
- Documentation: doxygen.

5.3. NaSCar

**Participant:** Michel Bergmann [correspondant].

This code is devoted to solve 3D-flows in around moving and deformable bodies. The incompressible Navier-Stokes equations are solved on fixed grids, and the bodies are taken into account thanks to penalization and/or immersed boundary methods. The interface between the fluid and the bodies is tracked with a level set function or in a Lagrangian way. The numerical code is fully second order (time and space). The numerical method is based on projection schemes of Chorin-Temam’s type. The code is written in C language and use Petsc (http://www.mcs.anl.gov/petsc/petsc-as/) library for the resolution of large linear systems in parallel.

NaSCar can be used to simulate both hydrodynamic bio-locomotion as fish like swimming and aerodynamic flows such wake generated by a wind turbine.

- Version: 1
- Keywords: numerical analyse, fluid mechanics, language C, PETSc
- Software benefit: simulate a flow around a deformable obstacle, moving into a fluid.
- APP: En cours
- Patent: non
- Type of human computer interaction: human for the moment
- OS/Middleware: unix, linux, mac os
- Required library or software: PETSc item Programming language: C
- Documentation: in progress

5.4. Other MC2 codes

- Penalization techniques on cartesian grids to solve incompressible Navier-Stokes equations
  - **Vortex:** sequential, Vortex In-Cell (VIC) scheme: hybrid vortex methods based on the combination of Lagrangian mesh-free schemes and Eulerian grid based schemes on the same flow region.
  - **NS2D(3D):** DNS, Finite Difference scheme, Multid solver, parallel MPI.

- Unstructured body fitted meshes
  - **Richards:** 2D Unstructured finite element code, implicit solver, sequential, to solve the transport-diffusion equations through a porous media including tidal forcing and mechanisms of diagenesis.
  - development inside FluidBox software in collaboration with BACCHUS. 2D-3D unstructured meshes, Stabilized Finite Elements method (SUPG), RANS turbulence model, parallel: Domain Decomposition and MPI.

- Compressible flows and elliptic problems
  - **Compressible flows:** 2D-3D finite volume scheme pour compressible Euler/NS equations on cartesian grids
  - **Elliptic problems:** 2D-3D finite difference scheme for elliptic interface problems, sequential and parallel
MICMAC Project-Team (section vide)
4. Software

4.1. The ECMPR software

Participant: Florence Forbes.

Joint work with: Radu Horaud and Manuel Iguel.

The ECMPR (Expectation Conditional Maximization for Point Registration) package implements [56] [17]. It registers two (2D or 3D) point clouds using an algorithm based on maximum likelihood with hidden variables. The method can register both rigid and articulated shapes. It estimates both the rigid or the kinematic transformation between the two shapes as well as the parameters (covariances) associated with the underlying Gaussian mixture model. It has been registered in APP in 2010 under the GPL license.

4.2. The LOCUS and P-LOCUS software

Participants: Florence Forbes, Senan James Doyle.

Joint work with: Michel Dojat.

From brain MR images, neuroradiologists are able to delineate tissues such as grey matter and structures such as Thalamus and damaged regions. This delineation is a common task for an expert but unsupervised segmentation is difficult due to a number of artefacts. The LOCUS software and its recent extension P-LOCUS automatically perform this segmentation for healthy and pathological brains. An image is divided into cubes on each of which a statistical model is applied. This provides a number of local treatments that are then integrated to ensure consistency at a global level, resulting in low sensitivity to artifacts. The statistical model is based on a Markovian approach that enables to capture the relations between tissues and structures, to integrate a priori anatomical knowledge and to handle local estimations and spatial correlations.

The LOCUS software has been developed in the context of a collaboration between Mistis, a computer science team (Magma, LIG) and a Neuroscience methodological team (the Neuroimaging team from Grenoble Institut of Neurosciences, INSERM). This collaboration resulted over the period 2006-2008 into the PhD thesis of B. Scherrer (advised by C. Garbay and M. Dojat) and in a number of publications. In particular, B. Scherrer received a “Young Investigator Award” at the 2008 MICCAI conference. Its extension for lesion detection is realized by S. Doyle with financial support from Gravit for possible industrial transfer.

The originality of this work comes from the successful combination of the teams respective strengths i.e. expertise in distributed computing, in neuroimaging data processing and in statistical methods.

4.3. The POPEYE software

Participant: Florence Forbes.

Joint work with: Vasil Khalidov, Radu Horaud, Miles Hansard, Ramya Narasimha, Elise Arnaud.

POPEYE contains software modules and libraries jointly developed by three partners within the POP STREP project: INRIA, University of Sheffield, and University of Coimbra. It includes kinematic and dynamic control of the robot head, stereo calibration, camera-microphone calibration, auditory and image processing, stereo matching, binaural localization, audio-visual speaker localization. Currently, this software package is not distributed outside POP.

4.4. The HDDA and HDDC toolboxes

Participant: Stéphane Girard.
Joint work with: Charles Bouveyron (Université Paris 1) and Gilles Celeux (Select, INRIA). The High-Dimensional Discriminant Analysis (HDDA) and the High-Dimensional Data Clustering (HDDC) toolboxes contain respectively efficient supervised and unsupervised classifiers for high-dimensional data. These classifiers are based on Gaussian models adapted for high-dimensional data [53]. The HDDA and HDDC toolboxes are available for Matlab and are included into the software MixMod [52]. Recently, a R package has been developed and integrated in The Comprehensive R Archive Network (CRAN). It can be downloaded at the following URL: http://cran.r-project.org/web/packages/HDclassif/.

4.5. The Extremes freeware
Participants: Laurent Gardes, Stéphane Girard.
Joint work with: Diebolt, J. (CNRS) and Garrido, M. (INRA Clermont-Ferrand-Theix).
The EXTREMES software is a toolbox dedicated to the modelling of extremal events offering extreme quantile estimation procedures and model selection methods. This software results from a collaboration with EDF R&D. It is also a consequence of the PhD thesis work of Myriam Garrido [54]. The software is written in C++ with a Matlab graphical interface. It is now available both on Windows and Linux environments. It can be downloaded at the following URL: http://extremes.gforge.inria.fr/.

4.6. The SpaCEM³ program
Participants: Lamiae Azizi, Senan James Doyle, Florence Forbes.
SpaCEM³ (Spatial Clustering with EM and Markov Models) is a software that provides a wide range of supervised or unsupervised clustering algorithms. The main originality of the proposed algorithms is that clustered objects do not need to be assumed independent and can be associated with very high-dimensional measurements. Typical examples include image segmentation where the objects are the pixels on a regular grid and depend on neighbouring pixels on this grid. More generally, the software provides algorithms to cluster multimodal data with an underlying dependence structure accounting for some spatial localisation or some kind of interaction that can be encoded in a graph.

This software, developed by present and past members of the team, is the result of several research developments on the subject. The current version 2.09 of the software is CeCILLB licensed.
Main features. The approach is based on the EM algorithm for clustering and on Markov Random Fields (MRF) to account for dependencies. In addition to standard clustering tools based on independent Gaussian mixture models, SpaCEM³ features include:

- The unsupervised clustering of dependent objects. Their dependencies are encoded via a graph not necessarily regular and data sets are modelled via Markov random fields and mixture models (eg. MRF and Hidden MRF). Available Markov models include extensions of the Potts model with the possibility to define more general interaction models.
- The supervised clustering of dependent objects when standard Hidden MRF (HMRF) assumptions do not hold (ie. in the case of non-correlated and non-unimodal noise models). The learning and test steps are based on recently introduced Triplet Markov models.
- Selection model criteria (BIC, ICL and their mean-field approximations) that select the "best" HMRF according to the data.
- The possibility of producing simulated data from:
  - general pairwise MRF with singleton and pair potentials (typically Potts models and extensions)
  - standard HMRF, ie. with independent noise model
  - general Triplet Markov models with interaction up to order 2
- A specific setting to account for high-dimensional observations.
- An integrated framework to deal with missing observations, under Missing At Random (MAR) hypothesis, with prior imputation (KNN, mean, etc), online imputation (as a step in the algorithm), or without imputation.
The software is available at http://spacem3.gforge.inria.fr. A user manual in English is available on the website above together with example data sets. The INRA Toulouse unit is more recently participating to this project for promotion among the bioinformatics community [20].

4.7. The FASTRUCT software

**Participant:** Florence Forbes.

**Joint work with:** Francois, O. (TimB, TIMC) and Chen, C. (former Post-doctoral fellow in Mistis).

The FASTRUCT program is dedicated to the modelling and inference of population structure from genetic data. Bayesian model-based clustering programs have gained increased popularity in studies of population structure since the publication of the software STRUCTURE [65]. These programs are generally acknowledged as performing well, but their running-time may be prohibitive. FASTRUCT is a non-Bayesian implementation of the classical model with no-admixture uncorrelated allele frequencies. This new program relies on the Expectation-Maximization principle, and produces assignment rivaling other model-based clustering programs. In addition, it can be several-fold faster than Bayesian implementations. The software consists of a command-line engine, which is suitable for batch-analysis of data, and a MS Windows graphical interface, which is convenient for exploring data.

It is written for Windows OS and contains a detailed user’s guide. It is available at http://mistis.inrialpes.fr/realisations.html.

The functionalities are further described in the related publication:

- Molecular Ecology Notes 2006 [55].

4.8. The TESS software

**Participant:** Florence Forbes.

**Joint work with:** Francois, O. (TimB, TIMC) and Chen, C. (former post-doctoral fellow in Mistis).

TESS is a computer program that implements a Bayesian clustering algorithm for spatial population genetics. It is particularly useful for seeking genetic barriers or genetic discontinuities in continuous populations. The method is based on a hierarchical mixture model where the prior distribution on cluster labels is defined as a Hidden Markov Random Field [59]. Given individual geographical locations, the program seeks population structure from multilocus genotypes without assuming predefined populations. TESS takes input data files in a format compatible to existing non-spatial Bayesian algorithms (e.g. STRUCTURE). It returns graphical displays of cluster membership probabilities and geographical cluster assignments through its Graphical User Interface.

The functionalities and the comparison with three other Bayesian Clustering programs are specified in the following publication:

- Molecular Ecology Notes 2007
MODAL Team

5. Software

5.1. MIXMOD

Participants: Christophe Biernacki, Serge Iovleff, Remi Lebret, Parmeet Bhatia.

MIXMOD (MIXture MODelling) is the core software of the MODAL team for two reasons. First, MIXMOD concerns main topics of MODAL since it is devoted to model-based supervised, unsupervised and semi-supervised classification for various data situations. Second, MIXMOD is now a well-distributed software since over 250 downloads/month are recorded for several years. Consequently, MIXMOD will be the main software for diffusing future methodological advances of the MODAL team.

MIXMOD is written in C++ (more than 10 000 lines), currently interfaced with Scilab and Matlab and distributed under GNU General Public License. An interface between MIXMOD and R is being to be developed by Rémi Lebret and will be soon available (during 2012).

Several other institutions participate in the MIXMOD development since several years: CNRS, INRIA Saclay-Île de France, Université de Franche-Comté, Université Lille 1. The software already benefits from several APP depositions and leads also to some international publications.

In addition, an INRIA ADT grant (Parmeet Bhatia) will also develop co-clustering models for continuous, binary and discrete data. It is a strategic development for MIXMOD since offering the ability to structure very large data tables both in lines and columns for different data types. In particular, it opens wide potential applications in biology, marketing, etc.

Serge Iovleff is the main supervisor of software engineers who are recruited for all the previously described tasks. More information about MIXMOD can be easily found on its web page http://www.mixmod.org/.

5.2. AAM

Participant: Serge Iovleff.

The AAM program is a R library implementing Auto-Associative models. Thus it could with few work transformed into a R package. As the AAM is a statistical model, the R language was well-suited for a diffusion inside the scientific community. It is a prototype for testing the AAM models against other kind of non-linear PCA models.

The first release was a scilab program written by Serge Iovleff and Stéphane Girard. It was rewritten in January 2009 and the code is now faster and produces enhanced graphics. The 2009 release is the result of a conjoint work of Serge Iovleff and a M1 internship of the ENS.

More information on the web site http://www.iut-info.univ-lille1.fr/~iovleff/softwares/

5.3. Kerfdr

Participant: Alain Célisse.

Computation of the local FDR: R package for biostatisticians allowing to estimate FDR and local FDR by kernel density estimation. This package allows also to deal with truncated data and to take into account supervision. More information on the website http://cran.r-project.org/web/packages/kerfdr/

---

1C. Biernacki, G. Celeux, G. Govaert and F. Langrognet, Model-Based Cluster and Discriminant Analysis with the MIXMOD Software, Computational Statistics and Data Analysis, Vol. 52, no 2, 587–600, 2006
5.4. MetaMa

**Participant:** Guillemette Marot.

metaMA is a specialised software for microarrays. It is a R package which combines either p-values or modified effect sizes from different studies to find differentially expressed genes. The main competitor of metaMA is geneMeta. Compared to geneMeta, metaMA offers an improvement for small sample size datasets since the corresponding modelling is based on shrinkage approaches.

Guillemette Marot is the main contributor and the maintainer of this packages and spent around one year full time for this package between the conception, the implementation, and the documentation. Her PhD advisors (Florence Jaffrézic, Claus-Dieter Mayer, Jean-Louis Foulley) helped her with the conception but she implemented alone the code.

First versions have been posted to the CRAN, the official website of the R software, in 2009. New versions for this package were released in August 2011 in order to take into account remarks from the main users (biologists or biostatisticians analysing gene expression data). This software is routinely used by biologists from INRA, Jouy en Josas (it has been included in a local analysis pipeline) but its diffusion on the CRAN makes it available to a wider community, as attested by the publications citing the software².

More information is available on the website [http://cran.r-project.org/web/packages/metaMA/](http://cran.r-project.org/web/packages/metaMA/)

5.5. STK++

**Participant:** Serge Iovleff.

STK++ is a multi-platform toolkit written in C++ for creating fast and easy to use data mining programs. It offers a large set of templated class in C++ which are suitable for projects ranging from small one-off projects to complete statistical application suites. A C equivalent would be gsl. However, STK++ is developed in C++ in order to get speed and reusability.

As the aim of STK++ is to aid developers to new developments, it proposes essentially interfaces classes and various concrete helping classes, like arrays, numerical methods (QR, SVD), input and output (csv files), random number generators... For instance, some part of the project will be integrated to the co-cluster project (in the MIXMOD software, see Section 5.1) actually developed by Parmeet Bathia.

The software is regularly developed since 10 years by Serge Iovleff and it is a work in progress. More information is available on the website [http://www.stkpp.org/](http://www.stkpp.org/) and source repository is here: [https://sourcesup.cru.fr/projects/stk/](https://sourcesup.cru.fr/projects/stk/)

5.6. SMVar

**Participant:** Guillemette Marot.

SMVar is a specialised software for microarrays. This R package implements the structural model for variances in order to detect differentially expressed genes from gene expression data. It performs gene expression differential analysis, based on a particular variance modelling. Its main competitor is the Bioconductor R package limma but limma assumes a common variance between the two groups to be compared while SMVar relaxes this assumption.

Guillemette Marot is the main contributor and the maintainer of this packages and spent around one year full time for this package between the conception, the implementation, and the documentation. Her PhD advisors (Florence Jaffrézic, Claus-Dieter Mayer, Jean-Louis Foulley) helped her with the conception but she implemented alone the code. She received some help from Anne de la Foye (INRA, Clermont-Ferrand) to correct the bugs in the first versions.

---

First versions have been posted to the CRAN, the official website of the R software, in 2009. New versions for this package were released in August 2011 in order to take into account remarks from the main users (biologists or biostatisticians analysing gene expression data). This software is routinely used by biologists from INRA, Jouy en Josas (it has been included in a local analysis pipeline) but its diffusion on the CRAN makes it available to a wider community, as attested by the publications citing the software\textsuperscript{3}.

More information on the website \url{http://cran.r-project.org/web/packages/SMVar/index.html}

5. Software

5.1. MAXW-DGTD

Participants: Joseph Charles, Tristan Cabel, Stéphane Lanteri [correspondant], Loula Fezoui.

MAXW-DGTD is a software suite for the simulation of time domain electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the time domain. MAXW-DGTD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes [14]. Within each element of the mesh, the components of the electromagnetic field are approximated by an arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with message passing programming using the MPI standard. Besides, a peripheral version of the software has been recently developed which is able to exploit the processing capabilities of a hybrid parallel computing system comprising multi-core CPU and GPU nodes [20]. Moreover, a recent methodological achievement has been the extension of the implemented DGTD method to deal with a Debye type dispersive propagation medium [35].

- AMS: AMS 35L50, AMS 35Q60, AMS 35Q61, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational electromagnetics, Maxwell equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middleware: Linux
- Required library or software: MPI (Message Passing Interface), CUDA
- Programming language: Fortran 77/95

5.2. MAXW-DGFD

Participants: Mohamed El Bouajaji, Stéphane Lanteri [correspondant].

MAXW-DGFD is a software suite for the simulation of time harmonic electromagnetic wave propagation. It implements a solution method for the Maxwell equations in the frequency domain. MAXW-DGFD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes. Within each element of the mesh, the components of the electromagnetic field are approximated by an arbitrary high order nodal polynomial interpolation method. The resolution of the sparse, complex coefficients, linear systems resulting from the discontinuous Galerkin formulation is performed by a hybrid iterative/direct solver whose design is based on domain decomposition principles. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with message passing programming using the MPI standard. Some recent achievements have been the implementation of non-uniform order DG method in the 2D case [17] and of a new hybridizable discontinuous Galerkin (HDG) formulation also in the 2D case [33].

- AMS: AMS 35L50, AMS 35Q60, AMS 35Q61, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational electromagnetics, Maxwell equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middleware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95
5.3. SISMO-DGTD  
**Participants:** Loula Fezoui, Nathalie Glinsky [correspondant], Stéphane Lanteri.

SISMO-DGTD is a software for the simulation of time domain seismic wave propagation. It implements a solution method for the velocity-stress equations in the time domain. SISMO-DGTD is based on a discontinuous Galerkin method formulated on unstructured triangular (2D case) or tetrahedral (3D case) meshes [5]. Within each element of the mesh, the components of the electromagnetic field are approximated by an arbitrary high order nodal polynomial interpolation method. This discontinuous Galerkin method combines a centered scheme for the evaluation of numerical fluxes at a face shared by two neighboring elements, with an explicit Leap-Frog time scheme. The software and the underlying algorithms are adapted to distributed memory parallel computing platforms thanks to a parallelization strategy that combines a partitioning of the computational domain with message passing programming using the MPI standard.

- AMS: AMS 35L50, AMS 35Q74, AMS 35Q86, AMS 65N08, AMS 65N30, AMS 65M60
- Keywords: Computational geoseismics, elastodynamic equations, discontinuous Galerkin, tetrahedral mesh.
- OS/Middleware: Linux
- Required library or software: MPI (Message Passing Interface)
- Programming language: Fortran 77/95

5.4. NUM3SIS  
**Participants:** Nora Aissiouene, Tristan Cabel, Thibaud Kloczko [SED team], Régis Duvigneau [OPALE project-team], Thibaud Kloczko [SED team], Stéphane Lanteri, Julien Wintz [SED team].

NUM3SIS [http://num3sis.inria.fr](http://num3sis.inria.fr) is a modular platform devoted to scientific computing and numerical simulation. It is designed to handle complex multidisciplinary simulations involving several fields such as Computational Fluid Dynamics (CFD), Computational Structural Mechanics (CSM) and Computational Electromagnetics (CEM). In this context, the platform provides a comprehensive framework for engineers and researchers that speeds up implementation of new models and algorithms. From a software engineering point of view, num3sis specializes and extends some layers of the meta-platform dtk, especially its core and composition layers. The core layer enables the user to define generic concepts used for numerical simulation such as mesh or finite-volume schemes which are then implemented through a set of plugins. The composition layer provides a visual programming framework that wraps these concepts inside graphical items, nodes. These nodes can then be connected to each other to define data flows (or compositions) corresponding to the solution of scientific problems. NUM3SIS provides a highly flexible, re-usable and efficient approach to develop new computational scenarios and takes advantage of existing tools. The team participates to the development of the NUM3SIS platform through the adaptation and integration of the MAXW-DGTD simulation software. This work is being carried out with the support of two engineers in the framework of an ADT (Action de Développement Technologique) program.

5.5. Medical Image Extractor  
**Participants:** Stéphane Lanteri, Julien Wintz [SED team].

---

3Service d’Experimentation et de Développement
Medical Image Extractor [link] provides functionalities needed to extract meshes from labeled MR or PET-CT medical images. It puts the emphasis on consistency, by generating both boundary surfaces, and volume meshes for each label (ideally identifying a tissue) of the input image, using the very same tetrahedrization. As this process requires user interaction, images and meshes are visualized together with tools allowing navigation and both easy and accurate refinement of the generated meshes, that can then be exported to serve as an input for other tools, within a multidisciplinary software toolchain. Using both DTK [link] and NUM3SIS SDKs, Medical Image Extractor comes within NUM3SIS’ framework. Using cutting edge research algorithms developed by different teams at Inria, spread among different research topics, namely, visualization algorithms from medical image processing, meshing algorithms from algorithmic geometry, it illustrates the possibility to bridge the gap between software that come from different communities, in an innovative and highly non invasive development fashion.

![Medical Image Extractor tool graphical user interface.](image)

*Figure 2. Medical Image Extractor tool graphical user interface.*
5. Software

5.1. SAMSON

A major objective of NANO-D is to try and integrate a variety of adaptive algorithms into a unified framework. As a result, NANO-D is developing SAMSON (Software for Adaptive Modeling and Simulation Of Nanosystems), a software platform aimed at including all developments from the group, in particular those described below.

The objective is to make SAMSON a generic application for computer-aided design of nanosystems, similar to existing applications for macrosystem prototyping (CATIA, SolidWorks, etc.).

The current architecture of SAMSON is visible in Figure 6. The code is organized into four main parts: a) the Base (in which “Core” contains, in particular, the heart of the adaptive algorithms: signaling mechanisms specifically designed for SAMSON), b) the Software Development Kit (SDK: a subset of the base that will be provided to module developers), c) Modules, and d) the SAMSON application itself.

Similar to the concept of Mathematica toolboxes, for example, the goal has been to make it possible to personalize the user interface of SAMSON for potentially many distinct applications. For example, we may want to personalize the interface of SAMSON for crystallography, drug design, protein folding, electronics, material science, nano-engineering, etc., by loading different modules at startup, depending on the user application domain.
5. Software

5.1. ORCCAD

Participants: Daniel Simon [correspondant], Soraya Arias [SED], Roger Pissard-Gibollet [SED].

ORCCAD is a software environment that allows for the design and implementation of the continuous and discrete time components of complex control systems, e.g. robotics systems which provided it first ground [64]. It also allows the specification and validation of complex missions to be performed by the system. It is mainly intended for critical real-time applications, in which automatic control aspects (servo loops) have to interact narrowly with the handling of discrete events (exception handling, mode switching). ORCCAD offers a complete and coherent vertical solution, ranging from the high level specification to real-time code generation. The ORCCAD V3 software was designed with proprietary tools that moreover are now becoming obsolete. ORCCAD V4 is currently deeply re-engineered to be compliant with open-source and free software tools (Java/Eclipse). Current targets are Linux (Posix threads) and Xenomai, a real-time development framework cooperating with the Linux kernel (http://www.xenomai.org). ORCCAD is supported by the Support Expérimentations & Développement (SED) service of INRIA-Rhône-Alpes. ORCCAD is used by the experimental robotics platforms of INRIA-Rhône-Alpes and by the Safenecs ANR project in a real-time simulator of a X4 drone. New functionalities and updates are developed jointly by the SED service and researchers of the NECS and SARDES teams. Web page: http://orccad.gforge.inria.fr.

5.2. MASim

Participants: J. Dumon [contact person], P. Bellemain [GIPSA-Lab], S. Nicolas [PROLEXIA], N. Maciol [PROLEXIA], F. Martinez [ROBOSOFT], J. Caquas [ROBOSOFT].

MASim is a tool that has been adapted from our former multiagent simulator MUSim (MUSim=MASim + ConnectSim + ConnectIHM). It integrates agent’s models, communication media including their limitations, heterogeneous network, and all the variants of the multi-agent control strategies. Besides the models and simulation engine, the simulation can be replayed through a GUI, an interactive graphical interface which is used to visualise and interpret the state of the multi-agent control system and communication topology. The validation scenario is a real-size application enough complex to enforce the pertinence of our results. The simulator MASim is now being used as an open research tool for various applications in the field of multi-agents networked systems, particularly within the FeedNetBack project (see Fig. 4).
Figure 4. A scenario’s view obtained with MASIM.
NON-A Team (section vide)
OPALE Project-Team

5. Software

5.1. NUM3SIS

Participants: Régis Duvigneau [correspondent], Thibaud Kloczko, Nora Aïssiouene.

NUM3SIS (http://num3sis.inria.fr) is a modular platform devoted to scientific computing and numerical simulation. It is not restricted to a particular application field, but is designed to host complex multidisciplinary simulations. Main application fields are currently Computational Fluid Dynamics (CFD), Computational Electro-Magnetics (CEM, in collaboration with Nachos Project-Team) and pedestrian traffic simulation.

The most important concept in NUM3SIS is the concept of node. It is a visual wrapper around derivatives of fundamental concepts such as data, algorithm or viewer. Atomic nodes are provided for convenience in order to manipulate computational data (such as grids or fields), apply computational methods (such as the building of a finite-element matrix or the construction of a finite-volume flux) and visualize computational results (such as vector or tensor fields, on a screen or in an immersive space). For a given abstract node, different implementations can be found, each of them being embedded in a plugin system that is managed by a factory.

The second important concept in NUM3SIS is the concept of composition. It consists of the algorithmic pipeline used to link the nodes together. The use of these two concepts, composition and nodes, provides a highly flexible, re-usable and efficient approach to develop new computational scenarios and take benefit from already existing tools. This is a great advantage with respect to classical monolithic softwares commonly used in these fields.

This work is being carried out with the support of two engineers in the framework of an ADT (Action de Développement Technologique) program.

5.2. FAMOSA

Participant: Régis Duvigneau [correspondant].

Opale team is developing the software platform FAMOSA (C++), that is devoted to multidisciplinary design optimization in engineering. It integrates the following components:

- an optimization library composed of various algorithms: several descent methods from steepest-descent method to quasi-Newton BFGS method (deterministic, smooth), the Multi-directional Search Algorithm (deterministic, noisy), the Covariance Matrix Adaption Evolution Strategy (semi-stochastic, multi-modal) and the Efficient Global Optimization method (deterministic, multi-modal). It also contains the Pareto Archived Evolution Strategy to solve multi-objective optimization problems;
- an evaluation library managing the performance estimation process (communication with external simulation tools);
- a metamodel library that contains tools to build a database and kriging models that are used to approximate the objective function for different purposes;
- a scenario library that allows to use the previous components to achieve various tasks:
  - Construct a design of experiments;
  - Construct a metamodel;
  - Find the design that minimizes a cost functional;
  - Find the Pareto front for two cost functionals;
  - Play a Nash game to find the equilibrium between two criteria;
  - Apply a multiple gradient descent strategy to improve simultaneously two criteria.
The FAMOSA platform is employed by Opale Project-Team to test its methodological developments in multidisciplinary design optimization (MDO). The platform is also used by the Fluid Mechanics Laboratory at Ecole Centrale de Nantes and by the K-Epsilon company (http://www.k-epsilon.com) for hydrodynamic design applications. Moreover, it is presently tested by Peugeot Automotive industry for external aerodynamic design purpose.

5.3. Plugins for AXEL

**Participants:** Régis Duvigneau [correspondant], Louis Blanchard.

Opale team is developing plugins in the framework of the algebraic modeler Axel, in collaboration with GALAAD team. These developments correspond to two research axes:

- methods for isogeometric analysis and design. In particular, two simulation tools for heat conduction and compressible flows have been implemented, in conjunction with some deterministic and semi-stochastic optimization algorithms for optimum-shape design;
- methods for geometrical modeling of bow shapes for trawler ships.

5.4. Integration platform for multidiscipline optimization applications

**Participants:** Toan Nguyen, Laurentiu Trifan.

A prototype software integration platform is developed and tested for multidiscipline optimization applications. It is based on a workflow management system called YAWL (http://www.yawlfoundation.org). The goal is to design, develop and assess high-performance distributed scientific workflows featuring resilience, fault-tolerance and exception-handling capabilities. The platform is used to experiment new resilience algorithms, including monitoring and management of application-level errors. The platform is tested against use-cases provided by the industry partners in the OMD2 project supported by the French Agence Nationale de la Recherche. This work is part of Laurentiu Trifan’s PhD thesis. (See Fig. 2.)

*Figure 1. Illustration of the graphic user interface of the NUM3SIS platform: at the top the composition space, at the bottom the visualization space.*
Figure 2. Testcase deployment on the Grid5000 infrastructure.
5. Software

5.1. Introduction

We are led to develop two types of software. The first category is prototype softwares: various softwares are developed in the framework of specific research contracts (and sometimes sold to the contractor) or during PhD theses. They may be also contributions to already existing softwares developed by other institutions such as CEA, ONERA or EDF. The second category is advanced software which are intended to be developed, enriched and maintained over longer periods. Such softwares are devoted to help us for our research and/or promote our research. We have chosen to present here only our advanced softwares.

5.2. MELINA

This software has been developed under the leadership of D. Martin for several years in order to offer to the researchers a very efficient tool (in Fortran 77 and object oriented) for easily implementing finite element based original numerical methods for solving partial differential equations. It has specific and original potential in the domain of time harmonic wave problems (integral representations, spectral DtN conditions,...). Nowadays, it is fully functional in various application areas (acoustics and aeroacoustics, elastodynamics, electromagnetism, water waves). It is an open source software with on line documentation available at http://homepage.mac.com/danielmartin/melina/
The software is regularly used in about 10 research laboratories (in France and abroad) and number of research papers have published results obtained with MELINA (see the Web site). Moreover, every 2 years, a meeting is organized which combines a workshop which teaches new users with presentations by existing users.
During the last four years, apart from various local improvements of the code, new functionalities have been developed:

• Higher order finite elements (up to $10^{	ext{th}}$ order),
• Higher order quadrature formulae,
• DtN boundary conditions in 3D.

A new C++ version of the software is under development.

5.3. MONTJOIE

Montjoie is a software for the efficient and accurate wave propagation numerical modeling in both time dependent or time harmonic regimes in various domains of application: acoustics, aeroacoustics, elastodynamics and electromagnetism. It is based essentially on the use of hexahedral-dominant (including a small part of tetrahedra, pyramids and prisms) conforming meshes and continuous or discontinuous Galerkin approximations. The use of tensor product basis functions coupled to appropriate numerical quadrature techniques leads to important gains in both computing time and memory storage. Various techniques for treating unbounded domains have been incorporated: DtN maps, local absorbing conditions, integral representations and PMLs.

We have written an interface for the use of other libraries: SELDON, a C++ linear algebra library (interfaced with BLAS and LAPACK) used for iterative linear solvers, MUMPS, PASTIX and UMFPACK for sparse direct solvers, ARPACK for eigenvalue computation, METIS and SCOTCH for mesh decomposition. Except for trivial geometries, the mesh generation is not part of the code. It can be done with Modulef, Gmsh, Ghs3D or Cubit.
This code has been developed by Marc Duruflé during his PhD thesis (in 2006). Some other contributors have brought more specific enrichments to the code. The online documentation is available at:
http://montjoie.gforge.inria.fr/.

The main contributions of 2010 have been the following:

- parallelization of the interface with ARPACK,
- unitary tests for SELDON and Montjoie aiming at stabilizing the code,
- implementation of $H(div)$ finite elements in 2D and 3D in collaboration with Morgane Bergot,
- optimization of the numerical simulation of the piano in collaboration with Juliette Chabassier (multithreading for the string nonlinear equations, improvement of the parallelization).

5.4. XLIFE++

During 2011, we performed a deep analysis of the two finite elements software developed by the lab (Melina and Montjoie) in order to propose a new software in C++ with extended capabilities and more integrated tools. The results of this analysis lead us to keep the general philosophy of Melina software (unified variational approach) but with major evolutions: integrated meshing tools, new variational description allowing FEM, BEM and DG formulation in an unified framework, global and local finite element computations, new approach to take into account essential boundary conditions and high performance computing skills (multithread and GPU computation). This new development will partly be supported by the Simposium european project dedicated to Non Destructive Testing tools (leader CEA/LIST, from september 2011 to august 2014) which requires some numerical simulation tools such as finite element library. It also a collaborative project with IRMAR (Rennes University). This new library, named xlife++ for eXtended LIbrary of Finite Element in C++, is an open source library (LGPL license) and its repository is the INRIA Gforge.
4. Software

4.1. FluidBox

Participants: Boniface Nkonga [contact], Hervé Guillard.

FluidBox is a software dedicated to the simulation of inert or reactive flows. It is also able to simulate multiphase, multi-material and MDH flows. There exist 2D and 3D dimensional versions. The 2D version is used to test new ideas that are later implemented in 3D. Two classes of schemes are available: A classical finite volume scheme and the more recent residual distribution schemes. Several low Mach number preconditioning are also implemented. The code has been parallelized with and without domain overlapping. The linear solver PaStiX is integrated in FluidBox. A partitioning tool exists in the package and uses Scotch. At present the software is only a private project but some parts of FluidBox are expected to be in the public domain by the end of the year.

4.2. PlaTo

Participants: Hervé Guillard [contact], Laure Combe.

The development of PlaTo (A platform for Tokamak simulation) (http://www-sop.inria.fr/pumas/plato.php) is being supported by an ADT action of the D2T. PlaTo is a suite of data and softwares dedicated to the geometry and physics of Tokamaks and its main objective is to provide the Inria large scale initiative “FUSION” teams working in plasma fluid models with a common development tool. The construction of this platform will integrate the following developments.

1. A (small) database corresponding to axi-symmetrical solutions of the equilibrium plasma equations for realistic geometrical and magnetic configurations (ToreSupra, JET and ITER). The construction of meshes always takes considerable time. Plato will provide meshes and solutions corresponding to equilibrium solutions that will be used as initial data for more complex computations.

2. A set of tool for the handling, manipulation and transformation of meshes and solutions using different discretisations (P1, Q1, P3, etc)

3. Numerical templates allowing the use of 3D discretization schemes using finite element schemes in the poloidal plane and spectral Fourier or structured finite volume representations in the toroidal plane.

4. Several applications (Ideal MHD and drift approximation) used in the framework of the Inria large scale initiative “FUSION”.

This year, after a definition of the PlaTo architecture, the points 1. and 2. have been developed.

4.3. PaMPA

Participants: Cécile Dobrzynski [Bacchus], Hervé Guillard, Laurent Hascoët [Tropics], Cédric Lachat, François Pellegrini [Bacchus].

PaMPA (“Parallel Mesh Partitioning and Adaptation”) is a middleware library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. An API of the future platform has been devised, and the coding of the mesh handling and redistribution routines is in progress. PaMPA will be used as a base module for the PLATO solvers, to balance dynamically, refine and coarsen its distributed mesh.
5. Software

5.1. BaPCod – a generic Branch-and-Price Code

Participants: Romain Leguay [Software Engineer], Pierre Pesneau, Ruslan Sadykov, François Vanderbeck [correspondant].

BaPCod is a prototype code that solves Mixed Integer Programs (MIP) by application of a Dantzig-Wolfe reformulation technique. The reformulated problem is solved using a branch-and-price (column generation) algorithm. This software platform, made of C++ classes, offers a “black-box” implementation that does not require user input and is not application specific. The features are

(i) the automation of the Dantzig-Wolfe reformulation process (the user defines a mixed integer programming problem in terms of variables and constraints, identifies subproblems, and can provide the associated solvers if available, but he does not need to explicitly define the reformulation, the explicit form of the columns, their reduced cost, or the Lagrangian bounds).

(ii) a default column generation procedure with standard initialization and stabilization (it may offer a selection of solvers for the master) – the issue of stabilization is discussed in [1], and

(iii) a default branching scheme – recent progress has been made on the issue of generic branching scheme in [23].

(iv) default primal heuristics specially developed for use in a decomposition framework [58].

The prototype software was/is used as background solver for 5 PhD thesis. It also served as the framework for our comparative study in an INRIA collaborative research action [1]. It has been experimented by two of our industrial partners, Exeo Solutions (Bayonne), on an inventory routing problem, and Orange Lab (France Telecom, Paris) on network design problems, time tabling problem by EURODECISION and it is currently being tested by the University Paris 6 and EDF. The prototype also enables us to be very responsive in our industrial contact.

See also the web page https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/BaPCod.

- Version: 1
REGULARITY Team

5. Software

5.1. FracLab

Participants: Paul Balança, Jacques Lévy Véhel [correspondant].

FracLab was developed for two main purposes:

1. propose a general platform allowing research teams to avoid the need to re-code basic and advanced techniques in the processing of signals based on (local) regularity.
2. provide state of the art algorithms allowing both to disseminate new methods in this area and to compare results on a common basis.

FracLab is a general purpose signal and image processing toolbox based on fractal, multifractal and local regularity methods. FracLab can be approached from two different perspectives:

• (multi-)fractal and local regularity analysis: A large number of procedures allow to compute various quantities associated with 1D or 2D signals, such as dimensions, Hölder and 2-microlocal exponents or multifractal spectra.
• Signal/Image processing: Alternatively, one can use FracLab directly to perform many basic tasks in signal processing, including estimation, detection, denoising, modeling, segmentation, classification, and synthesis.

A graphical interface makes FracLab easy to use and intuitive. In addition, various wavelet-related tools are available in FracLab.

FracLab is a free software. It mainly consists of routines developed in MatLab or C-code interfaced with MatLab. It runs under Linux, MacOS and Windows environments. In addition, a “stand-alone” version (i.e. which does not require MatLab to run) is available.

FracLab has been downloaded several thousands of times in the last years by users all around the world. A few dozens laboratories seem to use it regularly, with more than two hundreds registered users. Our ambition is to make it the standard in fractal softwares for signal and image processing applications. We have signs that this is starting to become the case. To date, its use has been acknowledged in more than two hundreds research papers in various areas such as astrophysics, chemical engineering, financial modeling, fluid dynamics, internet and road traffic analysis, image and signal processing, geophysics, biomedical applications, computer science, as well as in mathematical studies in analysis and statistics (see http://fraclab.saclay.inria.fr/ for a partial list with papers). In addition, we have opened the development of FracLab so that other teams worldwide may contribute. Additions have been made by groups in Australia, England, the USA, and Serbia.
SELECT Project-Team

5. Software

5.1. MIXMOD software

**Participants:** Gilles Celeux [Correspondant], Erwan Le Pennec.

MIXMOD is being developed in collaboration with Christophe Biernacki, Florent Langrognet (Université de Franche-Comté) and Gérard Govaert (Université de Technologie de Compiègne). MIXMOD (MIXture MODelling) software fits mixture models to a given data set with either a clustering or a discriminant analysis purpose. MIXMOD uses a large variety of algorithms to estimate mixture parameters, e.g., EM, Classification EM, and Stochastic EM. They can be combined to create different strategies that lead to a sensible maximum of the likelihood (or completed likelihood) function. Moreover, different information criteria for choosing a parsimonious model, e.g., the number of mixture component, some of them favoring either a cluster analysis or a discriminant analysis view point, are included. Many Gaussian models for continuous variables and multinomial models for discrete variable are available. Written in C++, MIXMOD is interfaced with Scilab and MATLAB. The software, the statistical documentation and also the user guide are available on the Internet at the following address: http://www.mixmod.org.

Since this year, MIXMOD has a proper graphical user interface (Version 1) which has been presented at the MIXMOD day in Lyon in December 2010. A version of MIXMOD in R is forthcoming.

Erwan Le Pennc with the help of Serge Cohen has proposed a spatial extension in which the mixture weights can vary spatially.
5. Software

5.1. Introduction

In 2011, SEQUEL continued the development of software for computer games (notably Go) and also developed two novel libraries for functional regression and data mining.

5.2. Computer Games

Participant: Rémi Coulom.

We developed three main softwares for computer games:

- **Crazy Stone** is a top-level Go-playing program that has been developed by Rémi Coulom since 2005. Crazy Stone won several major international Go tournaments in the past. In 2011, its strength improved to 5 dan on the KGS Go Server. It is distributed as a commercial product by Unbalance Corporation (Japan). 5-month work in 2011. URL: http://remi.coulom.free.fr/CrazyStone/

- **Crazy Hanafuda** is a new program to play the Japanese game of Hanafuda. 3 weeks of work in 2011. Discussion are in progress for licensing it.

- **CLOP** is a tool for automatic parameter optimization of game-playing programs. Distributed as freeware (GPL). One month of work in 2011. Available at: http://remi.coulom.free.fr/CLOP/

5.3. Functional Regression

Participant: Hachem Kadri.

A software package in C++ of algorithms for nonlinear functional data analysis using our operator-valued kernel framework (see sec. 6.4.1 ) is under development. A beta-version of the software can be downloaded at: https://gforge.inria.fr/frs/?group_id=982 .

The aim of this library is to grow and be shared in our scientific community, and also to be a software resource for our group.

5.4. Data mining library

Participant: Sertan Girgin.

A fully stand-alone library for data mining has been developed, including many classical algorithms for supervised and non supervised learning. This library is available as an internal resource for the group.
5. Software

5.1. inlinedocs

**Participant:** Toby Hocking [correspondant].

Generates Rd files from R source code with comments, providing for quick, sustainable package development. The syntax keeps code and documentation close together, and is inspired by the Don’t Repeat Yourself principle.

See also the web page [http://inlinedocs.r-forge.r-project.org/](http://inlinedocs.r-forge.r-project.org/).

- Version: 1.8
- Contact: toby.hocking@inria.fr

5.2. directlabels

**Participant:** Toby Hocking [correspondant].

The directlabels package provides an extensible framework for automatically placing direct labels onto multicolor lattice or ggplot2 plots. It includes heuristics for examining "lattice" and "ggplot" objects and inferring an appropriate Positioning Method for placing the labels. Furthermore, the design of directlabels makes it simple to create Positioning Methods for specific plots or libraries of portable Positioning Methods that can be re-used.

See also the web page [http://directlabels.r-forge.r-project.org/](http://directlabels.r-forge.r-project.org/).

- Version: 2.2
- Contact: toby.hocking@inria.fr

5.3. clusterpath

**Participant:** Toby Hocking [correspondant].

The clusterpath package provides an R/C++ implementation of the algorithms described in [16].

See also the web page [http://clusterpath.r-forge.r-project.org/](http://clusterpath.r-forge.r-project.org/).

- Version: 1.0
- Contact: toby.hocking@inria.fr

5.4. UGM

**Participant:** Mark Schmidt [correspondant].

UGM is a set of Matlab functions implementing various tasks in probabilistic undirected graphical models of discrete data with pairwise (and unary) potentials. Specifically, it implements a variety of methods for the following four tasks:

- Decoding: Computing the most likely configuration.
- Inference: Computing the partition function and marginal probabilities.
- Sampling: Generating samples from the distribution.
- Parameter Estimation: Given data, computing maximum likelihood (or MAP) estimates of the parameters.
The first three tasks are implemented for arbitrary discrete undirected graphical models with pairwise potentials. The last task focuses on Markov random fields and conditional random fields with log-linear potentials. The code is written entirely in Matlab, although more efficient mex versions of some parts of the code are also available. See also the web page [http://www.di.ens.fr/~mschmidt/Software/UGM.html](http://www.di.ens.fr/~mschmidt/Software/UGM.html).

- **Version:** 2011
- **Contact:** mark.schmidt@inria.fr

### 5.5. alphaBeta

**Participant:** Mark Schmidt [correspondant].

The code contains implementations of several available methods for the problem of computing an approximate minimizer of the sum of a set of unary and pairwise real-valued functions over discrete variables. This equivalent to the problem of MAP estimation, also known as decoding, in a pairwise undirected graphical model. The code focuses on scenarios where the pairwise energies encourage neighboring variables to take the same state. The particular methods contained in the package are iterated conditional mode, alpha-beta swaps, alpha-expansions, and alpha-expansion beta-shrink moves. See also the web page [http://www.di.ens.fr/~mschmidt/Software/alphaBeta.html](http://www.di.ens.fr/~mschmidt/Software/alphaBeta.html).

- **Version:** 1
- **Contact:** mark.schmidt@inria.fr

### 5.6. Matlab Software from “Graphical Model Structure Learning with L1-Regularization”

**Participant:** Mark Schmidt [correspondant].

This package contains the code used to produce the results in Mark Schmidt’s thesis: Roughly, there are five components corresponding to five of the thesis chapters:

- Chapter 2: L-BFGS methods for optimizing differentiable functions plus an L1-regularization term.
- Chapter 3: L-BFGS methods for optimizing differentiable functions with simple constraints or regularizers.
- Chapter 4: An L1-regularization method for learning dependency networks, and methods for structure learning in directed acyclic graphical models.
- Chapter 5: L1-regularization and group L1-regularization for learning undirected graphical models, using either the L2, Linf, or nuclear norm of the groups.
- Chapter 6: Overlapping group L1-regularization for learning hierarchical log-linear models, and an active set method for searching through the space of higher-order groups.


- **Version:** 1
- **Contact:** mark.schmidt@inria.fr

### 5.7. Hybrid deterministic-stochastic methods for data fitting

**Participant:** Mark Schmidt [correspondant].
Many structured data-fitting applications require the solution of an optimization problem involving a sum over a potentially large number of measurements. Incremental gradient algorithms (both deterministic and randomized) offer inexpensive iterations by sampling only subsets of the terms in the sum. These methods can make great progress initially, but often slow as they approach a solution. In contrast, full gradient methods achieve steady convergence at the expense of evaluating the full objective and gradient on each iteration. We explore hybrid methods that exhibit the benefits of both approaches. Rate of convergence analysis and numerical experiments illustrate the potential for the approach. See also the web page http://www.cs.ubc.ca/~mpf/2011-hybrid-for-data-fitting.html.

- Version: 1
- Contact: mark.schmidt@inria.fr
- Participants outside of Sierra: Michael Friedlander (Scientific Computing Laboratory, Department of Computer Science, University of British Columbia)

5.8. Multi-task regression using minimal penalties

Participant: Matthieu Solnon [correspondant].

This toolbox implements statistical algorithms designed to perform multi-task kernel ridge regressions, as described in [33]. See also the web page http://www.di.ens.fr/~solnon/articles/multi-task_regression/multitask_minpen_en.html.

- Version: 1
- Contact: matthieu.solnon@ens.fr
4. Software

4.1. DDNS2

**Participant:** Caterina Calgaro [correspondant].

The DDNS2 code is a parallel solver for unsteady incompressible Navier-Stokes flows in 2D geometries and primitive variables written in Fortran 95 with MPI as a message-passage library. Mixed finite element methods, with hierarchical basis, are used to discretize the equations and a non overlapping domain decomposition approach leads to an interface problem which involves a Lagrange multiplier corresponding to the velocity (the FETI approach). A dynamical multilevel method is developed locally on each subdomain. Several numerical estimates on the evolution of linear and nonlinear terms allow to construct the multilevel strategy which produces auto-adaptive cycles in time during which different mesh sizes, one for each subdomain, can be considered.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A1, SO3, SM1, EM1, SDL1, DA1, CD4, MS4, TPM1.

4.2. NS3ED

**Participant:** Caterina Calgaro [correspondant].

The NS3ED code is a solver for steady incompressible Navier-Stokes flows in three-dimensional exterior domains, written in C++. The truncated problem is discretized using an exponential mesh and an equal-order velocity-pressure finite element method, with additional stabilization terms. A bloc-triangular preconditioner is performed for the generalized saddle-point problem.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A1, SO3, SM1, EM1, SDL1, DA1, CD4, MS4, TPM1.

4.3. ns2ddv-M

**Participants:** Caterina Calgaro [correspondant], Emmanuel Creusé [correspondant], Thierry Goudon, Manuel Bernard.

The NS2DDV-M code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. This original approach for variable density flows is described in [49]. The NS2DDV-M code will be available on the SIMPAF team web page before the end of 2011.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A3, SO3-up4, SM2-up3, EM3, SDL4, DA1, CD4, MS4, TPM4.

Software web site: http://math.univ-lille1.fr/~simpaf/SITE-NS2DDV/home.html

4.4. ns2ddv-C++

**Participants:** Caterina Calgaro [correspondant], Emmanuel Creusé [correspondant], Thierry Goudon.
The NS2DVD-C++ code is based on a hybrid method coupling FV and FE approaches for solving the variable density Navier-Stokes equation in dimension 2. The code is developed around the GetFem++ and the Bamg softwares. It allows in particular mesh refinement strategies so that very relevant simulations can be reached (as the falling droplet with very high density ratios, see for example [47]). The current version of the code consider the additional terms in the Kazhikhov-Smagulov model.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A1, SO3-up4, SM1, EM2, SDL1, DA1, CD4, MS4, TPM1.

**4.5. RTcodes**

**Participants:** Pauline Lafitte [correspondant], Jean-François Coulombel, Christophe Besse, Thierry Goudon.

We have developed a set of numerical codes, written in Scilab, to compute the solutions of the system coupling the Euler equations to the radiation through energy exchanges, in the non equilibrium regime. This covers several situations in the hierarchy of asymptotic problems. The code treats the one-dimensional framework. In particular the code can be used to investigate radiative shocks profiles. The main advantage of our numerical codes is that they do not require any refinement near the singularities. The numerical tests show a very good agreement with the theoretical predictions.

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A2, SO3, SM2, EM1, SDL1.

**4.6. FPcodes**

**Participants:** Pauline Lafitte [correspondant], Thierry Goudon.

We have developed a numerical code, written in Scilab, to compute the solutions of the two-phase flows equations describing particles interacting with a fluid through friction forces. The code treats one-dimensional situation and is well adapted to describe gravity driven flows in either bubbling or flowing regimes. In particular, it can be used to describe the evolution of pollutants in the atmosphere. The numerical strategy, based on a asymptotic-based scheme, is described in details in [50].

Here is the self-assessment of the team effort following the grid provided by Inria (see: http://www.inria.fr/institut/organisation/instances/commission-d-evaluation): A2, SO3, SM2, EM1, SDL1.

**4.7. CLAToolBox**

**Participants:** Christophe Besse [correspondant], Pauline Klein.

As a byproduct of the review paper [39], a user-friendly interface is offered to trial and compare various numerical methods to solve the 1D Schrödinger equation with absorbant boundary conditions. We also mention [42] for a numerical investigation of blow-up phenomena in the nonlinear Schrödinger equation.

**4.8. SPARCS**

**Participants:** Christophe Besse, Thierry Goudon [correspondant], Ingrid Lacroix-Violet.

SPARCS is the code developed by Thales Alenia Space for the simulation of the charge phenomena the spacecrafts are subject to. The current version of the code, according to the PhD thesis of O. Chanrion and M. Chane-Yook performed in collaboration with the team Caiman at Sophia Antipolis, is specialized to geostationary atmospheres. The model consists in the stationary Vlasov-Poisson system, but where instationary effects are taken into account with the boundary condition for the electric field. We participate, in particular through the post doc of N. Vauchelet, to the elaboration of an improved version of the code which includes parallization optimized procedures, the modeling of the natural difference of potential between different dielectric surfaces of the spacecraft, as well as the possible presence of devices emitting charged particles.

---

SMASH Project-Team (section vide)
5. Software

5.1. MoGo

Participants: Olivier Teytaud [correspondent], Hassen Doghmen, Jean-Baptiste Hoock.

MoGo and its Franco-Taiwanese counterpart MoGoTW is a Monte-Carlo Tree Search program for the game of Go, which made several milestones of computer-Go in the past (first wins against professional players in 19x19; first win with disadvantageous side in 9x9 Go); MoGo has had new developments as follows:

- A Meta-MCTS module (inspired by the collaboration with Tristan Cazenave in the ANR EXPLO-RA project), which provided both a huge opening book in 9x9 and an approximate solving of 7x7 Go[24].
- Following the “poolRave” modification, introduction of machine learning and statistics into MCTS, such as:
  - Bernstein Races [41] (for offline educating Monte-Carlo simulations).
These developments have been summarized and compared in [93].
- Variants of Go: we tested variants of Go, in particular blind variants; this suggests that in such frameworks playing theoretically suboptimal moves helps a lot, because such unnatural moves are harder to memorize. A preliminary related publication is [25]; some additional results are to be published. Another interesting variant is random-Go, starting from a randomly generated board; such situations are much harder for humans, and, interestingly, our program was competitive in front of a 6D player (ranked 4th in a world amateur championship and former French champion) on a 19x19 board[40].

MoGo’s development team was awarded the 2010 ChessBase award for the best contribution to Computer-Games.

5.2. Covariance Matrix Adaptation Evolution Strategy

Participant: Nikolaus Hansen [correspondent].

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is one of the most powerful continuous domain evolutionary algorithms. The CMA-ES is considered state-of-the-art in continuous domain evolutionary computation\(^3\) and has been shown to be highly competitive on different problem classes. The algorithm is widely used in research and industry as witnessed by hundreds of published applications. We provide source code for the CMA-ES in C, Java, Matlab, Octave, Python, and Scilab including the latest variants of the algorithm.

Links: http://www.lri.fr/~hansen/cmaes_inmatlab.html

5.3. Comparing Continuous Optimizers

Participants: Nikolaus Hansen [correspondent], Raymond Ros, Anne Auger, Marc Schoenauer.

COCO (Comparing Continuous Optimizers) is a platform for systematic and sound comparisons of real-parameter global optimizers. COCO provides benchmark function testbeds (noiseless and noisy) and tools for processing and visualizing data generated by one or several optimizers. The code for processing experiments is provided in Matlab and C. The post-processing code is provided in Python. The code has been improved and used for the GECCO 2009 and 2010 workshops on “Black Box Optimization Benchmarking” (BBOB) (see Section 6.4), and will serve as a basis for the test platform in the CSDL project.

5.4. GridObservatory

Participants: Michèle Sebag, Cécile Germain-Renaud [correspondent], Julien Nauroy, Yusik Kim.

The Grid Observatory software suite collects and publishes traces of the EGI (European Grid Initiative) grid usage. With the release and extensions of its portal, the Grid Observatory has made a database of grid usage traces available to the wider computer science community. These data are stored on the grid, and made accessible through a web portal without the need of grid credentials. More than 100 users are currently registered. The GO is supported by an INRIA ADT (Action de Développement Technologique).

In 2011, the suite has been extended to energy consumption. The first barrier to improved energy efficiency of IT systems is the lack of large-scale collections of experimental data. The Green Computing Observatory (GCO), part of the GO initiative monitors a large computing center (Laboratoire de l’Accélérateur Linéaire - LAL) within the EGI grid, and publishes the data through the Grid Observatory. The GCO is supported by the CNRS PEPS program, and by University Paris-Sud through the MRM (Moyens de Recherche Mutualisés) program.

Portal site: http://www.grid-observatory.org
5. Software

5.1. CarbonQuant

**Participants:** Mireille Bossy [correspondant], Jacques Morice, El Hadj Aly Dia.

CarbonQuant is a simulator project of CO2 allowances prices on a EU-ETS type market, by an indifference price approach.

It aims to demonstrate the high potentiality of stochastic control solvers, to quantify sensibilities of a carbon market with respect to its design.

Starting in September 2011, CarbonQuant is an ADT Inria.

See also the web page [http://carbonvalue.gforge.inria.fr](http://carbonvalue.gforge.inria.fr).

- Version: 0.1
TROPICS Project-Team

5. Software

5.1. AIRONUM

**Participant:** Alain Dervieux [correspondant].

AIRONUM is an experimental software that solves the unsteady compressible Navier-Stokes equations with K-epsilon, LES-VMS and hybrid turbulence modelling on parallel platforms with Mpi as parallel programming concept. The mesh model is unstructured tetrahedrization, with possible mesh motion.

See also the web page [http://www-sop.inria.fr/tropics/aironum](http://www-sop.inria.fr/tropics/aironum).

- Version: v 1.0
- Programming language: FORTRAN95 (mostly). About 100,000 lines.

AIRONUM was developed by INRIA and university of Montpellier. It is used by INRIA, university of Montpellier and university of Pisa (I). AIRONUM is used as an experimental platform for:

- Numerical approximation of compressible flows, such as upwind mixed element volume approximation with superconvergence on regular meshes.
- Numerical solution algorithms for the implicit time advancing of the compressible Navier-Stokes equations, such as parallel scalable deflated additive Schwarz algorithms.
- Turbulence modelling such as the Variational Multiscale Large eddy Simulation and its hybridization with RANS statistical models.

5.2. TAPENADE

**Participants:** Laurent Hascoët [correspondant], Valérie Pascual.

TAPENADE is an Automatic Differentiation tool that transforms an original source program into a new source program that computes derivatives of the original program. Automatic Differentiation produces analytical derivatives, that are exact up to machine precision. The reverse mode of Automatic Differentiation is able to compute gradients at a cost which is independent from the number of input variables. TAPENADE accepts source programs written in Fortran 77, Fortran 90, or C. It provides differentiation in the following modes: tangent, multi-directional tangent, and reverse. Documentation is provided on the web site of the research team and as the INRIA technical report RT-0300. TAPENADE runs under Linux or Windows operating systems, and requires installation of Java jdk1.6 or upward.

See also the web page [http://www-sop.inria.fr/tropics/](http://www-sop.inria.fr/tropics/).

- Version: v 3.6, september 2011
- ACM: D.3.4 Compilers; G.1.0 Numerical algorithms; G.1.4 Automatic differentiation; I.1.2 Analysis of algorithms
- AMS: 65K10; 68N20
- APP: IDDN.FR.001.040038.000.S.P.2002.000.31235
- Keywords: automatic differentiation, adjoint, gradient, optimisation, inverse problems, static analysis, data-flow analysis, compilation
- Programming language: Java

TAPENADE implements the results of our research about models and static analyses for AD. TAPENADE is can be downloaded and installed on most architectures. Alternatively, it can be used as a web server. TAPENADE differentiates computer programs according to the model described in section 3.1 Higher-order derivatives can be obtained through repeated application of tangent AD on tangent and/or reverse AD.
TAPENADE performs sophisticated data-flow analysis, flow-sensitive and context-sensitive, on the complete source program to produce an efficient differentiated code. Analyses include Type-Checking, Read-Write analysis, and Pointer analysis. AD-specific analysis include:

- **Activity analysis:** This detects variables whose derivative is either null or useless, to reduce the number of derivative instructions.
- **Adjoint Liveness analysis:** This detects the source statements that are dead code for the computation of derivatives.
- **TBR analysis:** In reverse mode, this reduces the set of source variables that need to be recovered.

TAPENADE is not open-source. Academic usage is free. Industrial or commercial usage require a paying license, as detailed on the team’s web page. The software has been downloaded several hundred times, and the web tool served several thousands of true connections (not robots). The tapenade-users mailing list is over one hundred registered users.
4. Software

4.1. Software

This section briefly comments on all the software distributed by ABS. On the one hand, the software released in 2011 is briefly described as the context is presented in the sections dedicated to new results. On the other hand, the software made available before 2011 is briefly specified in terms of applications targeted.

In any case, the web page advertising a given software also makes related publications available.

4.1.1. vorpatch and compatch: Modeling and Comparing Protein Binding Patches

Participants: Frédéric Cazals, Noël Malod-Dognin.

Context. Our work on the problem of modeling and comparing atomic resolution protein interfaces has been discussed in sections 5.4.1 and 5.1.1. The programs undertaking these two tasks are respectively named vorpatch and compatch.


4.1.2. voratom: Modeling with Toleranced Models

Participants: Frédéric Cazals, Tom Dreyfus.

Context. Our TOLeranced Model framework has been described in sections 5.2.1 and 5.2.2. The corresponding software package includes programs to (i) perform the segmentation of (probability) density maps, (ii) construct toleranced models, (iii) explore toleranced models (geometrically and topologically), (iv) compute Maximal Common Induced Sub-graphs (MCIS) and Maximal Common Edge Sub-graphs (MCES) to assess the pairwise contacts encoded in a TOM.

Distribution. Binaries for the aforementioned programs are made available from http://cgal.inria.fr/abs/voratom/.

4.1.3. wsheller: Selecting Water Layers in Solvated Protein Structures

Participants: Frédéric Cazals, Christine Roth.

Context. Given a snapshot of a molecular dynamics simulation, a classical problem consists of quenching that structure—minimizing the potential energy of the solute together with selected layers of solvent molecules. The program wsheller provides a solution to the water layer selection, and incorporates a topological control of the layers selected.


4.1.4. intervor: Modeling Macro-molecular Interfaces

Participant: Frédéric Cazals.

In collaboration with S. Loriot, from the GEOMETRY FACTORY.

Context. Modeling the interfaces of macro-molecular complexes is key to improve our understanding of the stability and specificity of such interactions. We proposed a simple parameter-free model for macro-molecular interfaces, which enables a multi-scale investigation—from the atomic scale to the whole interface scale. Our interface model improves the state-of-the-art to (i) identify interface atoms, (ii) define interface patches, (iii) assess the interface curvature, (iv) investigate correlations between the interface geometry and water dynamics / conservation patterns / polarity of residues.
**Distribution.** The following web site http://cgal.inria.fr/abs/Intervor serves two purposes: on the one hand, calculations can be run from the web site; on the other hand, binaries are distributed for Linux. To the best of our knowledge, this software is the only publicly available one for analyzing Voronoi interfaces in macro-molecular complexes.

### 4.1.5. vorlume: Computing Molecular Surfaces and Volumes with Certificates

**Participant:** Frédéric Cazals.

*In collaboration with S. Loriot, from the GEOMETRY FACTORY.*

**Context.** Molecular surfaces and volumes are paramount to molecular modeling, with applications to electrostatic and energy calculations, interface modeling, scoring and model evaluation, pocket and cavity detection, etc. However, for molecular models represented by collections of balls (Van der Waals and solvent accessible models), such calculations are challenging in particular regarding numerics. Because all available programs are overlooking numerical issues, which in particular prevents them from qualifying the accuracy of the results returned, we developed the first certified algorithm, called vorlume. This program is based on so-called certified predicates to guarantee the branching operations of the program, as well as interval arithmetic to return an interval certified to contain the exact value of each statistic of interest—in particular the exact surface area and the exact volume of the molecular model processed.

**Distribution.** Binaries for Vorlume is available from http://cgal.inria.fr/abs/Vorlume.

### 4.1.6. ESBTL: theEasy Structural Biology Template Library

**Participant:** Frédéric Cazals.

*In collaboration with S. Loriot (the Geometry Factory), and J. Bernauer, from the EPI AMIB.*

**Context.** The ESBTL (Easy Structural Biology Template Library) is a lightweight C++ library that allows the handling of PDB data and provides a data structure suitable for geometric constructions and analyses.

**Distribution.** The source C++ code is available from http://esbtl.sourceforge.net/.

### 4.1.7. A_purva: Comparing Protein Structure by Contact Map Overlap Maximization

**Participant:** Noël Malod-Dognin.

*In collaboration with N. Yanev, University of Sofia, and IMI at Bulgarian Academy of Sciences, Bulgaria, and R. Andonov, INRIA Rennes - Bretagne Atlantique, and IRISA/University of Rennes 1, France.*

**Context.** Structural similarity between proteins provides significant insights about their functions. Maximum Contact Map Overlap maximization (CMO) received sustained attention during the past decade and can be considered today as a credible protein structure measure. The solver A_purva is an exact CMO solver that is both efficient (notably faster than the previous exact algorithms), and reliable (providing accurate upper and lower bounds of the solution). These properties make it applicable for large-scale protein comparison and classification.

**Distribution.** The software is available from http://apurva.genouest.org.
4. Software

4.1. Varna

Participants: Yann Ponty [correspondant], Alain Denise.

VARNA is a tool for the automated drawing, visualization and annotation of the secondary structure of RNA, designed as a companion software for web servers and databases. VARNA implements four drawing algorithms, supports input/output using the classic formats dbn, ct, bpseq and RNAML and exports the drawing, either as a bitmap (JPEG, PNG) or as a vector picture (SVG, EPS and XFIG). It also allows manual modification and structural annotation of the resulting drawings using either an interactive point and click approach, within a web server or through command-line arguments. VARNA is a free software distributed under the terms of the GPLv3.0 license and available at http://varna.lri.fr. VARNA is currently used by RNA scientists (Cited by 44 research articles since its presentation in Fall of 2009), web servers such as the BOULDEALE webserver (http://www.microbio.me/boulderae/), the TFOlD webserver (http://tfold.ibisc.univ-evry.fr/TFold/), the CYLOFOLD webserver (http://cylofold.abc.cncfrf.gov/), and by databases such as the IRESITE database (http://iresite.org/), SRNATARBASE (http://ceb.bmi.ac.cn/srnatarbase/) and the RFAM database (http://rfam.sanger.ac.uk/), the main source of sequence/structure data for RNA scientist, to display secondary structures. It is also used as an integrated component within JALVIEW, arguably one of the leading sequence alignment editor (http://www.jalview.org/).

4.2. GeneValorization

Participant: Sarah Cohen-Boulakia [correspondant].

High-throughput technologies provide fundamental information concerning thousands of genes. Many of the current research laboratories daily use one or more of these technologies and end-up with lists of genes. Assessing the originality of the results obtained includes being aware of the number of publications available concerning individual or multiple genes and accessing information about these publications. Faced with the exponential growth of publications available and number of genes involved in a study, this task is becoming particularly difficult to achieve. We introduce GENEVERALORIZATION, a web-based tool which gives a clear and handful overview of the bibliography available corresponding to the user input formed by (i) a gene list (expressed by gene names or ids from ENTREZGENE) and (ii) a context of study (expressed by keywords). From this input, GENEVERALORIZATION provides a matrix containing the number of publications with co-occurrences of gene names and keywords. Graphics are automatically generated to assess the relative importance of genes within various contexts. Links to publications and other databases offering information on genes and keywords are also available. To illustrate how helpful GENEVERALORIZATION is, we have considered the gene list of the OncotypeDX prognostic marker test. it is available at http://bioguide-project.net/gv.

4.3. HSIM

Participant: Patrick Amar [correspondant].

HSIM is a simulation tool for studying the dynamics of biochemical processes in a virtual bacteria. The model is given using a language based on probabilistic rewriting rules that mimics the reactions between biochemical species. HSIM is a stochastic automaton which implements an entity-centered model of objects. This kind of modelling approach is an attractive alternative to differential equations for studying the diffusion and interaction of the many different enzymes and metabolites in cells which may be present in either small or large numbers. This software is freely available at http://www.lri.fr/~pa/Hsim; A compiled version is available for the Windows, Linux and MacOSX operating systems.
4.4. Cartaj

Participants: Alain Denise [correspondant], Alexis Lamiable.

Cartaj is a software that automatically predicts the topological family of three-way junctions in RNA molecules, from their secondary structure only. The Cartaj software http://cartaj.lri.fr that implements our method can be used online. It is also meant for being part of RNA modelling softwares and platforms. The methodology and the results of Cartaj are presented in [16].
ANUBIS Project-Team (section vide)
4. Software

4.1. SOFA

Participants: Hervé Delingette [correspondant], Brina Goyette, Federico Spadoni, Stéphanie Marchesseau, Hugo Talbot.

SOFA is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop newer algorithms, but can also be used as an efficient prototyping tool. Based on an advanced software architecture, it allows to:
- create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA;
- modify most parameters of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an XML file;
- build complex models from simpler ones using a scene-graph description; efficiently simulate the dynamics of interacting objects using abstract equation solvers; reuse and easily compare a variety of available methods. It is mainly developed by the Inria team project Shaman, Evasion and Asclepios.

See also the web page http://www.sofa-framework.org/ .

- ACM: J.2 Physics, J.3 LIFE AND MEDICAL SCIENCES
- Software benefit: Simulation of the human body
- License: GPL
- License: LGPL
- Type of human computer interaction: console, opengl, qt
- OS/Middleware: linux, windows, mac
- Required library or software: Qt - GPL - GLEW - BSD/MIT - Tinyxml - zlib
- Programming language: C/C++
- Documentation: - each function of the core API and each class in the SOFA modules - doxygen

- ACM: J.3
- Programming language: C/C++

4.2. MedINRIA

Participants: Benoît Bleuzé, Olivier Clatz [correspondant], Vincent Garcia, Michael Knopke, Stephan Schmitt, Maxime Sermesant, John Stark, Nicolas Toussaint.

MedInria is a free collection of softwares developed by the Asclepios research project in collaboration with the Athena, Parietal and Visages Inria research projects. It aims at providing to clinicians state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms. MedInria is available for Microsoft windows XP/Vista/7, Linux Fedora Core, MacOSX, and is fully multithreaded.

See also the web page http://med.inria.fr/ .

- Version: 2.0
- Keywords: Medical Image Processing
- License: Proprietary Licence
- Type of human computer interaction: QT
- OS/Middleware: Windows - Linux - MacOSX
- Required library or software: DTI Track (Proprietary), vtkINRIA3D (CeCillB), Baladin (Proprietary), DT-REFInd (Proprietary)
- Programming language: C++
5. Software

5.1. OpenMEEG

Participants: Théodore Papadopoulo, Maureen Clerc, Emmanuel Olivi, Alexandre Gramfort [Parietal project-team].

OpenMEEG provides state-of-the art tools for low-frequency bio-electromagnetism, notably solving forward problems related to EEG and MEG [5]. It implements the symmetric BEM, thus providing excellent accuracy. OpenMEEG is a free open software written in C++. It can be accessed either through a command line interface or through Python/Matlab interfaces.

OpenMEEG is multiplatform (Linux, MacOS, Windows) and it is distributed under the French opensource license CeCILL-B. See also the web page http://www-sop.inria.fr/athena/software/OpenMEEG/ .

5.2. Diffusion MRI

Participants: Aurobrata Ghosh, Rachid Deriche.

The algorithms previously developed within the ODYSSEE Project team and related to the Diffusion Tensor and Q-Ball imaging are available upon request from the INRIA source forge (https://gforge.inria.fr). One can use all the estimation and visualization tools developed, ranging from estimation, regularization, segmentation to Q-ball estimation, fiber ODF estimation and tractography algorithms. New visualization tools for Q-Ball images represented by spherical harmonic decomposition have also been developed.

The software library comprises geometric and variational methods devised to estimate, regularize, segment and perform tractography in DT (Diffusion Tensor) and HARDI (High Angular Resolution) MRI images. The library is multi-platform (Linux, Windows and OS X) ans is embedded into two open-source high level languages, TCL and Python.

Thanks to the ADT MedInria-NT, this library is in the process to be partly integrated within the interactive medical imaging platform MedINRIA.
5. Software

5.1. AcypiCyc

**Participants:** Hubert Charles [EPI], Patrice Baa Puyoule [Contact, Patrice.Baa-Puyoulet@lyon.inra.fr], Stefano Colella [Contact, stefano.colella@lyon.inra.fr], Ludovic Cottret, Marie-France Sagot [EPI], Augusto Vellozo [Contact, augusto@cycadsys.org], Amélie Véron.

Database of the metabolic network of *Acyrthosiphon pisum*.

http://acypicyc.cycadsys.org/

5.2. BaobabLuna

**Participants:** Marília Braga [Contact, mdvbraga@gmail.com], Marie-France Sagot [EPI], Eric Tannier.

Manipulation of signed permutations in the context of genomic evolution.

http://pbil.univ-lyon1.fr/software/luna/

5.3. Cassis

**Participants:** Christian Baudet [EPI, Contact, christian.baudet@univ-lyon1.fr], Christian Gautier [EPI], Claire Lemaître [Contact, claire.lemaître@inria.fr], Marie-France Sagot [EPI], Eric Tannier.

Algorithm for precisely detecting genomic rearrangement breakpoints.

http://pbil.univ-lyon1.fr/software/Cassis/

5.4. Cravela

**Participants:** Ana Teresa Freitas, Nuno Mendes [EPI, Contact, ndm@kdbio.inesc-id.pt], Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].

Framework for the identification and evaluation of miRNA precursors (finished), targets (in development) and regulatory modules (in development).

http://www.cravela.org/

5.5. C3P

**Participants:** Frédéric Boyer, Anne Morgat [EPI, ext. member], Alain Viari [EPI, Contact, alain.viari@inria.fr].

Merging two or more graphs representing biological data (e.g. pathways, ...).

http://www.inrialpes.fr/helix/people/viari/cccpart

5.6. CycADS

**Participants:** Hubert Charles [EPI], Patrice Baa Puyoule [Contact, Patrice.Baa-Puyoulet@lyon.inra.fr], Stefano Colella [Contact, stefano.colella@lyon.inra.fr], Ludovic Cottret, Marie-France Sagot [EPI], Augusto Vellozo [Contact, augusto@cycadsys.org].

Cyc annotation database system.

http://www.cycadsys.org/

5.7. Ed’Nimbus

**Participants:** Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Marie-France Sagot [EPI].

Algorithm for detecting and filtering repeats in sequences prior to multiple alignments.
5.8. GeM

Participants: Gisèle Bronner, Christian Gautier [EPI, Contact, christian.gautier@univ-lyon1.fr], Bruno Spataro.

Database for comparative genomic analysis of complete vertebrate genomes.
http://pbil.univ-lyon1.fr/gem/gem_home.php

5.9. Gobbolino

Participants: Vicente Acuña [EPI], Etienne Birmelé [EPI, délégation], Ludovic Cottret, Pierluigi Crescenzi, Fabien Jourdan, Vincent Lacroix, Alberto Marchetti-Spaccamela [EPI, ext. member], Andrea Marino, Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI], Leen Stougie [EPI, ext. member].

Algorithm to enumerate all metabolic stories in a metabolic network given a set of metabolites of interest. Code available on request.

5.10. kisSnp

Participants: Vincent Lacroix [EPI], Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Nadia Pisanti, Marie-France Sagot [EPI], Nicolas Schnel.

Algorithm for identifying SNPs without a reference genome by comparing raw reads.
http://alcovna.genouest.org/kissnp/

5.11. kisSplice

Participants: Rayan Chikhi, Janice Kielbassa [EPI], Vincent Lacroix [Contact, EPI], Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Gustavo Sacomoto [EPI], Marie-France Sagot [EPI], Raluca Uricaru.

Algorithm for de novo calling alternative splicing events from RNA-seq data.
http://alcovna.genouest.org/kisssplice/

5.12. MetExplore

Participants: Michael Barrett, Hubert Charles [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Fabien Jourdan, Marie-France Sagot [EPI], Florence Vinson, David Wildridge.

Web server to link metabolomic experiments and genome-scale metabolic networks.
http://metexplore.toulouse.inra.fr/metexplore/

5.13. Migal

Participants: Julien Allali [Contact, julien.allali@labri.fr], Marie-France Sagot [EPI].

Algorithm for comparing RNA structures.

5.14. MotusWEB

Participants: Ludovic Cottret, Fabien Jourdan, Vincent Lacroix [EPI, Contact, vincent.lacroix@univ-lyon1.fr], Odile Rogier, Marie-France Sagot [EPI].

Algorithm for searching and inferring coloured motifs in metabolic networks (web-based version - offers different functionalities from the downloadable version).
http://pbil.univ-lyon1.fr/software/motus_web/

5.15. Motus

Participants: Ludovic Cottret, Fabien Jourdan, Vincent Lacroix [EPI, Contact, vincent.lacroix@univ-lyon1.fr], Odile Rogier, Marie-France Sagot [EPI].
Algorithm for searching and inferring coloured motifs in undirected graphs (downloadable version - offers different functionalities from the web-based version).
http://pbil.univ-lyon1.fr/software/motus/

5.16. PhEVER
Participants: Christian Gautier [EPI], Vincent Lotteau, Leonor Palmeira [Contact, mlpalmeira@ulg.ac.be], Chantal Rabourdin-Combe, Simon Penel.
Database of homologous gene families built from the complete genomes of all available viruses, prokaryotes and eukaryotes and aimed at the detection of virus/virus and virus/host lateral gene transfers.
http://pbil.univ-lyon1.fr/databases/phever/

5.17. PepLine
Participants: Jérôme Garin, Alain Viari [EPI, Contact, alain.viari@inria.fr].
Pipeline for the high-throughput analysis of proteomic data.
http://www.grenoble.prabi.fr/protehome/software/pepline

5.18. Pitufu
Participants: Vicente Acuña [EPI], Ludovic Cottret [Contact, Ludovic.Cottret@toulouse.inra.fr], Alberto Marchetti-Spaccamela [EPI, ext. member], Paulo Vieira Milreu [EPI, Contact, pvmilreu@gmail.com], Marie-France Sagot [EPI], Leen Stougie [EPI, ext. member], Fabio Vudani-Martínez.
Algorithm to enumerate all minimal sets of precursors of target compounds in a metabolic network.
http://sites.google.com/site/pitufosoftware/

5.19. PSbR
Participants: Yoan Diekmann, Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr], Eric Tannier.
Algorithm for testing the evolution and conservation of common clusters of genes.
http://pbil.univ-lyon1.fr/members/sagot/htdocs/team/software/PSbR/

5.20. Repseek
Participants: Guillaume Achaz [Contact, achaz@abi.snv.jussieu.fr], Eric Coissac, Alain Viari [EPI].
Finding approximate repeats in large DNA sequences.
http://wwwabi.snv.jussieu.fr/~public/RepSeek/

5.21. Smile
Participants: Laurent Marsan, Marie-France Sagot [EPI, Contact, marie-france.sagot@inria.fr].
Motif inference algorithm taking as input a set of biological sequences.

5.22. Tuiuiu
Participants: Alair Pereira do Lago, Pierre Peterlongo [Contact, pierre.peterlongo@inria.fr], Nadia Pisanti, Gustavo Sacomoto [EPI], Marie-France Sagot [EPI].
Multiple repeat search filter with edit distance.
http://mobyle.genouest.org/cgi-bin/Mobyle/portal.py?form=tuiuiu

5.23. UniPathway
Participants: Eric Coissac, Anne Morgat [EPI, Contact, anne.morgat@inria.fr], Alain Viari [EPI].
Database of manually curated pathways developed with the Swiss-Prot group.
http://www.unipathway.org
4. Software

4.1. Software

4.1.1. Continuation of M3N

A large part of the software currently in use in the project-team was initiated and developed within former projects (Menusin, M3N).

4.1.2. CellSys

Participants: Dirk Drasdo [correspondent], Stefan Höhme [Research Associate, University of Leipzig], Adrian Friebel [PhD student, University of Leipzig], Tim Johann [Software Engineer, University of Leipzig], Nick Jagiella [PhD student].

Computer simulation software for individual cell (agent)-based models of tumour and tissue growth solved either by systems of coupled equations of motion for each individual cell or by Kinetic Monte Carlo methods [77].
4. Software

4.1. aevol (artificial evolution)

Participants: Guillaume Beslon, Stephan Fischer, Carole Knibbe, David P. Parsons, Bérénice Batut.

- Contact: Carole Knibbe (carole.knibbe@inrialpes.fr).
- Aevol is a simulation software dedicated to the study of genome evolution. It allows to carry out in silico experimental evolution. Populations of digital organisms reproduce and mutate randomly, with both small mutations and large chromosomal rearrangements, in a steady or varying environment. A curve-fitting task is used to determine the fitness of the organisms and thus their rate of reproduction. The number of genes, their order, their sequences, their intergenic distances are all free to evolve.
- URL: http://gforge.liris.cnrs.fr/projects/aevol/

4.2. DMT4SP (Data Mining Tool For Sequential Patterns)

Participant: Christophe Rigotti.

- Contact: Christophe.Rigotti@insa-lyon.fr.
- Summary: The dmt4sp prototype is a command line tool to extract episodes and episode rules, supporting various constraints, over a single sequence or several sequences of events. Three kinds of patterns can be extracted: (1) serial episodes, (2) serial episode rules having a single event type in the consequent, and (3) quantitative episodes (aka grouping of “homogeneous” occurrences of the serial episodes with respect to the time gap between events).
- Url: http://liris.cnrs.fr/~crigotti/dmt4sp.html
5. Software

5.1. Identification of biological systems

We are currently considering the possibility to implement our Matlab algorithms into the Matlab toolbox *Contsid*, developed by the System Identification team of the CRAN (http://www.iris.cran.uhp-nancy.fr/contsid/).
5. Software

5.1. Supervision software

We are developing a software for the supervision of bioreactors: this platform, named ODIN, has been built for the smart management of bioreactors (data acquisition, fault diagnosis, automatic control algorithm,...). This software was developed in C++ and uses a Scilab engine to run the advanced algorithms developed within BIOCORE. It has been implemented and validated with four different applications.
5. Software

5.1. YASS – local homology search

_Actively maintained._
Software self-assessment following the mechanisms provided by INRIA Evaluation Committee for software evaluation: A-4, SO-3, SM-2, EM-3, SDL-4, DA-4, CD-4, MS-4, TPM-4
Software web site: http://bioinfo.lifl.fr/yass/

Objective: YASS is an open source software devoted to the classical problem of genomic pairwise alignment, and use most of our knowledge to design and implement efficient seeding techniques these last years. YASS is frequently used, it always receives more than 300 web queries per month (excluding INRIA and Univ-Lille1 local queries), and is also frequently downloaded and cited.

5.2. Carnac – RNA structure prediction

_Actively maintained._
Software web site: http://bioinfo.lifl.fr/carnac/

The CARNAC program is for RNA structure prediction by comparative analysis. The web interface also offers 2D visualisation tools and alignment functionalities with gardenia. It has proven to be very fast and very specific compared to its competitors [21].

5.3. TFM-Explorer – Identification and analysis of transcription factor binding sites

_Actively maintained._
Software web site: http://bioinfo.lifl.fr/TFM/

The TFM suite is a set of tools for analysis of transcription factor binding sites. locating and analyzing transcription factor binding sites using Position Weight Matrices. In this suite, the TFM-EXPLORER tool is designed to analyze regulatory regions of eukaryotic genomes using comparative genomics and local over-representation.

5.4. Regliss – RNA locally optimal structures

_Actively developed in 2011._
Software self-assessment: A-2, SO-4, SM-2, EM-2, SDL-4, DA-4, CD-4, MS-4, TPM-4
Software web site: http://bioinfo.lifl.fr/RNA/regliss/

REGLISS is a tool that studies the energy landscape of a given RNA sequence by considering locally optimal structures. Locally optimal structures are thermodynamically stable structures that are maximal for inclusion: they cannot be extended without producing a conflict between base pairs in the secondary structure, or increasing the free energy. The tool generates all locally optimal structures in a given sequence. Moreover, REGLISS can be used to explore the neighborhood of structures through an energy landscape graph.
5.5. RNAspace – a platform for noncoding RNA annotation

*Actively developed in 2011.*

Software self-assessment: A-5, SO-3, SM-3-up4, EM-2-up3, SDL-4, DA-4, CD-4, MS-4, TPM-4


RNAspace is an open source platform born from a national collaborative initiative. Its goal is to develop and integrate functionalities allowing structural and functional noncoding RNA annotation (see Section 6.2): [http://www.rnaspace.org](http://www.rnaspace.org), and it is distributed under the GPL licence. The project has been awarded by the national IBISA label in autumn 2009<sup>1</sup>.

5.6. CGseq – a toolbox for comparative analysis

*Actively maintained in 2011.*


Software website: [http://bioinfo.lifl.fr/CGseq/](http://bioinfo.lifl.fr/CGseq/)

CGseq is a toolbox to identify functional regions in a genomic sequence by comparative analysis using multispecies comparison.

5.7. Biomanycores.org – a community for bioinformatics on manycore processors

*Actively developed in 2011.*

Software self-assessment: A-3-up4, SO-2, SM-2, EM-3, SDL-4-up5, DA-4, CD-4, MS-4, TPM-4


Manycore architectures are an emerging field of research full of promises for parallel bioinformatics. However, the usage of GPUs is not so widespread in the end-user bioinformatics community. The goal of the biomanycores.org project is to gather open-source CUDA and OpenCL parallel codes and to provide easy installation, benchmarking, and interoperability. The last point includes interfaces to popular frameworks such as Biopython, BioPerl and BioJava.

The development of Biomanycores is supported by a national ADT<sup>2</sup> between BONSAI, SYMBIOSE (CRI Rennes) and DOLPHIN (CRI Lille). This ADT started in October 2010 and led to the hiring of J.-F. Berthelot (IJD).

In the first year of the ADT, J.-F. Berthelot redesigned and rewrote almost all the existing code. The code base is now stable. He worked on the documentation and on various software engineering aspects such as continuous integration. The second year of the ADT will focus on integrating more applications and targeting bioinformaticians users.

5.8. Norine – a resource for nonribosomal peptides

*Actively developed in 2011.*

Software self-assessment: A-5, SO-3, SM-3-up4, EM-2-up3, SDL-4, DA-4, CD-4, MS-4, TPM-4


Objective: Norine is a public computational resource that contains a database of NRPs with a web interface and dedicated tools, such as a 2D graph viewer and editor for peptides or comparison of NRPs.

Project management: Norine was created and is maintained by members of Bonsai team, in tight collaboration with members of the ProBioGEM lab, a microbiological laboratory of Lille1 University.

---

<sup>1</sup>IBISA is a French consortium for evaluating and funding national technological platforms in life sciences.

<sup>2</sup>ADT (Action for Technological Development) is an INRIA internal call.
Users community: Since its creation in 2006, Norine has gained a universal recognition as the unique database dedicated to non-ribosomal peptides because of its high quality and manually curated annotations. It is queried from all around the world by biologists or biochemists. It receives more than 3000 queries per month. Norine main users come for 13% from the United States of America, for 12% from the United Kingdom, for 5% from China or for 4% from Germany where renowned biology laboratories work on nonribosomal peptides (NRPs) or on their synthetases.

Improvements: This year, the source code has been reorganised by Laurie Tonon, a SED engineer, to use model view controller software architecture, implemented with Struts2.

5.9. GkArrays – indexing high throughput sequencer reads

*Actively maintained.*


Software web site: [http://crac.gforge.inria.fr/gkarrays/](http://crac.gforge.inria.fr/gkarrays/)

Objective: Gk-Arrays is a C++ library specifically dedicated to indexing reads produced by high-throughput sequencers. This index allows to answer queries centred on reads. It also takes benefits from the input specificity to lower space consumption.

This library is the result of a collaboration with N. Philippe and T. Commes (IGH laboratory, Montpellier), M. Léonard and T. Lecroq (LITIS laboratory, Rouen) and É. Rivals (LIRMM laboratory, Montpellier). We plan to improve our library in the forthcoming months with the help of Master’s students.
CARMEN Team (section vide)
5. Software

5.1. Urban air quality analysis

Participants: Anne Tilloy, Vivien Mallet.

“Urban Air Quality Analysis” carries out data assimilation at urban scale. It merges the outputs of a numerical model (maps of pollutant concentrations) with observations from an air quality monitoring network, in order to produce the so-called analyses, that is, corrected concentration maps. The data assimilation computes the Best Linear Unbiased Estimator (BLUE), with a call to the data assimilation library Verdandi. The error covariance matrices are parameterized for both model simulations and observations. For the model state error covariances, the parameterization primarily relies on the road network. The software handles ADMS output files, for a posteriori analyses or in an operational context.

5.2. Polyphemus

Participants: Vivien Mallet, Pierre Tran, Damien Garaud, Anne Tilloy.

Polyphemus (see the web site http://cerea.enpc.fr/polyphemus/ ) is a modeling system for air quality. As such, it is designed to yield up-to-date simulations in a reliable framework: data assimilation, ensemble forecast and daily forecasts. Its completeness makes it suitable for use in many applications: photochemistry, aerosols, radionuclides, etc. It is able to handle simulations from local to continental scales, with several physical models. It is divided into three main parts:

- libraries that gather data processing tools (SeldonData), physical parameterizations (AtmoData) and postprocessing abilities (AtmoPy);
- programs for physical preprocessing and chemistry-transport models (Polair3D, Castor, two Gaussian models, a Lagrangian model);
- drivers on top of the models in order to implement advanced simulation methods such as data assimilation algorithms.

Figure 1 depicts a typical result produced by Polyphemus. Clime is involved in the overall design of the system and in the development of advanced methods in model coupling, data assimilation and ensemble forecast (through drivers and post-processing).

In 2011, Polyphemus was extended for a better integration with the data assimilation library Verdandi. A first (unstable) version of Polyphemus with a complete overhaul of the input/output operations and of the configuration files was provided to the developers. The derivative of Polyphemus that is used at IRSN was used for the first time in a crisis context in order to simulate the transport of radionuclides during the Fukushima nuclear disaster.

5.3. Data assimilation library: Verdandi

Participants: Kévin Charpentier, Marc Fragu [MACS], Vivien Mallet, Dominique Chapelle [MACS], Philippe Moireau [MACS], Sergiy Zhuk, Anne Tilloy.

The leading idea is to develop a data assimilation library intended to be generic, at least for high-dimensional systems. Data assimilation methods, developed and used by several teams at INRIA, are generic enough to be coded independently of the system to which they are applied. Therefore these methods can be put together in a library aiming at:

- making easier the application of methods to a great number of problems,
- making the developments perennial and sharing them,
- improving the broadcast of data assimilation works.
An object-oriented language (C++) has been chosen for the core of the library. A high-level interface to Python is automatically built. The design raised many questions, related to high dimensional scientific computing, the limits of the object contents and their interfaces. The chosen object-oriented design is mainly based on three class hierarchies: the methods, the observation managers and the models. Several base facilities have also been included, for message exchanges between the objects, output saves, logging capabilities, computing with sparse matrices.

In 2011, versions 0.9, 1.0 and 1.1 of Verdandi were released. These versions are advanced enough to be used by the data assimilation community. Compared to previous versions, the additions are: 4D-Var, ensemble Kalman filter, redesigned perturbation managers, sequential aggregation, improvements in the documentation and an improved support of Windows.
5. Software

5.1. Spiking neural networks simulation
Participants: Mohamed-Ghaïth Kaabi, Dominique Martinez.

A spiking neuron is usually modeled as a differential equation describing the evolution over time of its membrane potential. Each time the voltage reaches a given threshold, a spike is sent to other neurons depending on the connectivity. A spiking neural network is then described as a system of coupled differential equations. For the simulation of such a network we have written two simulation engines: (i) Mvaspike based on an event-driven approach and (ii) sirene based on a time-driven approach.

- **Mvaspike**: The event-driven simulation engine was developed in C++ and is available on [http://mvaspike.gforge.inria.fr](http://mvaspike.gforge.inria.fr). Mvaspike is a general event-driven purpose tool aimed at modeling and simulating large, complex networks of biological neural networks. It allows to achieve good performance in the simulation phase while maintaining a high level of flexibility and programmability in the modeling phase. A large class of spiking neurons can be used ranging from standard leaky integrate-and-fire neurons to more abstract neurons, e.g. defined as complex finite state machines.

- **Sirene**: The time-driven simulator engine was written in C and is available on [http://sirene.gforge.inria.fr](http://sirene.gforge.inria.fr). It has been developed for the simulation of biologically detailed models of neurons —such as conductance-based neurons— and synapses. Its high flexibility allows the user to implement easily any type of neuronal or synaptic model and use the appropriate numerical integration routine (e.g. Runge-Kutta at given order).

5.2. DANA: Implementation of computational neuroscience mechanisms
Participants: Nicolas Rougier, Mathieu Lefort, Wahiba Taouali.

Computational neuroscience is a vast domain of research going from the very precise modeling of a single spiking neuron, taking into account ion channels and/or dendrites spatial geometry up to the modeling of very large assemblies of simplified neurons that are able to give account of complex cognitive functions. DANA attempts to address this latter modeling activity by offering a Python computing framework for the design of very large assemblies of neurons using numerical and distributed computations. However, there does not exist something as a unified model of neuron: if the formal neuron has been established some sixty years ago, there exists today a myriad of different neuron models that can be used within an architecture. Some of them are very close to the original definition while some others tend to refine it by providing extra parameters or variables to the model in order to take into account the great variability of biological neurons. DANA makes the assumption that a neuron is essentially a set of numerical values that can vary over time due to the influence of other neurons and learning. DANA aims at providing a constrained and consistent Python framework that guarantee this definition to be enforced anywhere in the model, i.e., no symbol, no homunculus, no central executive.

5.3. ENAS: Event Neural Assembly Simulation
Participants: Frédéric Alexandre, Axel Hutt, Nicolas Rougier, Thierry Viéville.
EnaS (that stands for “Event Neural Assembly Simulation”) is a middleware implementing our last numerical and theoretical developments, allowing to simulate and analyze so called “event neural assemblies”. The recent achievements include (in collaboration with the Neuromathcomp EPI): spike trains statistical analysis via Gibbs distributions, spiking network programming for exact event’s sequence restitution, discrete neural field parameters algorithmic adjustments and time-constrained event-based network simulation reconciling clock and event based simulation methods. It has been designed as plug-in for our simulators (e.g. DANA or Mvaspike) as other existing simulators (via the NeuralEnsemble meta-simulation platform) and additional modules for computations with neural unit assembly on standard platforms (e.g. Python or the Scilab platform).

5.4. OpenViBE

Participants: Laurent Bougrain, Baptiste Payan.

OpenViBE is a C++ open-source software devoted to the design, test and use of Brain-Computer Interfaces. The OpenViBE platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. Key features of the platform are its modularity, its high-performance, its portability, its multiple-users facilities and its connection with high-end/Virtual Reality displays. The “designer” of the platform enables to build complete scenarios based on existing software modules using a dedicated graphical language and a simple Graphical User Interface (GUI). This software is available on the INRIA Forge under the terms of the LGPL-V2 license. The development of OpenVibe is done in association with the INRIA research team BUNRAKU for the national INRIA project: ADT LOIC (cf. § 7.2).

5.5. CLONES: Closed-Loop Neural Simulations

Participant: Thomas Voegtlin.

The goal of this work is to provide an easy-to-use framework for closed-loop simulations, where interactions between the brain and body of an agent are simulated.

We developed an interface between the Sofa physics engine, (http://www.sofa-framework.org) and the Brian neural simulator (http://www.briansimulator.org). The interface consists in a Sofa plugin and a Python module for Brian. Sofa and Brian use different system processes, and communicate via shared memory. Synchronization between processes is achieved through semaphores.

As a demonstration of this interface, a physical model of undulatory locomotion in the nematode *c. elegans* was implemented, based on the PhD work of Jordan H. Boyle.

CLONES was presented at the Python in Neuroscience Workshop [18].

5.6. GINNet-DynNet: Decision-making platform

Participants: Laurent Bougrain, Marie Tonnelier.

GINNet (Graphical Interface for Neural Networks) is a decision-aid platform written in Java, intended to make neural network teaching, use and evaluation easier, by offering various parametrizations and several data pre-treatments. GINNet is based upon a local library for dynamic neural network developments called DynNet. DynNet (Dynamic Networks) is an object-oriented library, written in Java and containing base elements to build neural networks with dynamic architecture such as Optimal Cell Damage and Growing Neural Gas. Classical models are also already available (multi-layer Perceptron, Kohonen self-organizing maps, ...). Variable selection methods and aggregation methods (bagging, boosting, arcing) are implemented too.

The characteristics of GINNet are the following: Portable (100% Java), accessible (model creation in few clicks), complete platform (data importation and pre-treatments, parametrization of every models, result and performance visualization). The characteristics of DynNet are the following: Portable (100% Java), extensible (generic), independent from GINNet, persistent (results are saved in HML), rich (several models are already implemented), documented.
This platform is composed of several parts:

1. Data manipulation: Selection (variables, patterns), descriptive analysis (stat., PCA..), detection of missing, redundant data.
2. Corpus manipulation: Variable recoding, permutation, splitting (learning, validation, test sets).
6. Results: Error curves, confusion matrix, confidence interval.

DynNet and GINNet are free softwares, registrated to the APP and distributed under CeCILL license, Java 1.4 compatible (http://ginnet.gforge.inria.fr). GINNet is available as an applet. For further information, see http://gforge.inria.fr/projects/ginnet (news, documentations, forums, bug tracking, feature requests, new releases...).
5. Software

5.1. Software

5.1.1. FES muscle modeling in opensim framework

**Participants:** Mitsuhiro Hayashibe, Philippe Fraisse, Emel Demircan, Oussama Khatib (INRIA Equipe Associee, Stanford Univ.).

In FES, movement synthesis and control are still challenging tasks due to the complexity of whole body dynamics computation and the nonlinearity of stimulated muscle dynamics. An efficient movement synthesis means that criteria can be defined and evaluated through an accurate numeric simulation. We perform the implementation of muscle model representing the electrically stimulated muscle into the OpenSim framework which has whole body musculoskeletal geometry. We would like to develop the FES simulator using Stanford Operational Space Whole-Body Controller which allows the real-time motion generation with virtual FES and finally we aim at the development of motion correction controller to find the appropriate FES signals against a disabled motor function.

5.1.2. Further development of gom2n software - a toolchain to simulate and investigate selective stimulation strategies for FES

**Participants:** Guillaume Jourdain, Pawel Maciejasz, Jeremy Laforet, Christine Azevedo Coste, David Guiraud.

Concurrently with the experiments on selective stimulation of nerve fibres, performed on earthworms (see section 6.1.6), also the gom2n toolchain developed previously by our team was further developed. Main objective of this work was to be able to simulate similar behaviour of nerve fibres, as observed during electrical stimulation of the giant nerve fibres of earthworms, and therefore to be able to compare computational and experimental results. Main improvements which has been implemented in the new version of the gom2n toolchain are:

- improved and more intuitive users interface
- possibility to perform concurrently multiple simulations for various stimulation parameters, as well as various diameters and locations of nerve fibres within the nerve.

Further work is however still needed to adapt electrical properties of simulated fibres, since electrical properties of the earthworm’s giant nerve fibres are different that properties of mammalian nerve fibres.”

5.1.3. RdP to VHDL tool

**Participants:** Gregory Angles, David Andreu, Thierry Gil.

Our SENIS (Stimulation Electrique Neurale dIStribuee) based FES architecture relies on distributed stimulation units (DSU) which are interconnected by means of a 2-wire based network. A DSU is a complex digital system since its embeds among others a dedicated processor (micro-machine with a specific reduced instruction set), a monitoring module and a 3-layer protocol stack. To face the complexity of the unit’s digital part and to ease its prototyping on programmable digital devices (e.g. FPGA), we developed an approach for high level hardware component programming (HILECOP). To support the modularity and the reusability of sub-parts of complex hardware systems, the HILECOP methodology is based on components. An HILECOP component has: a Petri Net (PN) based behavior, a set of functions whose execution is controlled by the PN, and a set of variables and signals. Its interface contains places and transitions from which its PN model can be inter-connected as well as signals it exports or imports. The interconnection of those components, from a behavioral point out view, consists in the interconnection of places and/or transitions according to well-defined mechanisms: interconnection by means of oriented arcs or by means of the “merging” operator (existing for both places and transitions).
The development of an Eclipse-based version of HILECOP has been achieved. This new version of HILECOP has been registered (new deposit) in September 2011, at the French Agence de Protection des Programmes (APP) with the IDDN.FR.001.380008.000.S.P.2011.000.31235.

It will be accessible to the academic community at the beginning of 2012.

5.1.4. SENISManager

Participants: Robin Passama, David Andreu.

We developed a specific software environment called SENISManager allowing to remotely manage and control a network of DSUs, i.e. the distributed FES architecture. SENISManager performs self-detection of the architecture being deployed (Fig. 1; left). This environment allows the manipulation of micro-programs from their edition to their remote control (Fig. 1; right). It also allows the programming of control sequences executed by an external controller in charge of automatically piloting a stimulator.

This new version of SENIS Manager has been registered (updated deposit) in September 2011, at the French Agence de Protection des Programmes (APP), with the IDDN.FR.001.320011.001.S.P.2009.000.31500.
4. Software

4.1. PyGMAlion

PyGMAlion (Plant Growth Model Analysis, Identification and Optimization) has become the leading development project in the group. The objective is on one hand to provide modelers with mathematical and statistical tools for model analysis, and on the other hand to capitalize in the same software the different methods developed in the group. The basic idea is that provided the modeler writes his dynamic system of plant growth in a simple frame (defining model state variables, state function, parameters, external inputs, and model observations) then some parameter estimation methods are available, as well as sensitivity analysis, evaluation of criteria for model selection and data assimilation.
5. Software

5.1. CelDyn

Participants: Nikolai Bessonov, Vitaly Volpert [correspondant].

Software "Celdyn" is developed in order to model cell population dynamics for biological applications. Cells are represented either as soft spheres or they can have more complex structure. Cells can divide, move, interact with each other or with the surrounding medium. Different cell types can be introduced. When cells divide, the types of daughter cells are specified. A user interface is developed.
3. Software

3.1. M1cg1

- Participant: J. Ch. Gilbert.
- Version: 1.2.
- Programming language: Fortran 77.
- Solves a convex quadratic optimization problem and builds a preconditioning matrix, 1 download in 2011.
- See also the web page [http://www-roc.inria.fr/~gilbert/modulopt/optimization-routines/m1cg1/m1cg1.html](http://www-roc.inria.fr/~gilbert/modulopt/optimization-routines/m1cg1/m1cg1.html).

3.2. M1qn3

- Participants: J. Ch. Gilbert, C. Lemaréchal.
- Version: 3.3.
- Programming language: Fortran 77.
- Solves a very large scale differentiable optimization problem, 45 downloads in 2011.
- See also the web page [http://www-roc.inria.fr/~gilbert/modulopt/optimization-routines/m1qn3/m1qn3.html](http://www-roc.inria.fr/~gilbert/modulopt/optimization-routines/m1qn3/m1qn3.html).

3.3. PHlab

- Participant: J. Ch. Gilbert.
- Version: 0.1.
- Programming language: Matlab.
- Solves a stochastic linear optimization problem defined on a scenario tree by the Progressive Hedging algorithm [12].

3.4. Sklml

Participants: Quentin Carbonneaux, François Clément, Pierre Weis.

Easy coarse grain parallelization.

See also the web page [http://sklml.inria.fr/](http://sklml.inria.fr/).

- Version: 1.0+pl1
- Programming language: OCaml

3.5. SQPlab

- Participant: J. Ch. Gilbert.
- Version: 0.4.5.
- Programming language: Matlab.
- See also the web page [http://www-roc.inria.fr/~gilbert/modulopt/optimization-routines/sqp/qp.html](http://www-roc.inria.fr/~gilbert/modulopt/optimization-routines/sqp/qp.html).

3.6. LifeV

Participant: Michel Kern.
Finite element library with emphasis on life and environmental sciences. LifeV is the joint collaboration between École Polytechnique Fédérale de Lausanne (Switzerland), Politecnico di Milano (Italy), Inria (France) and Emory University (U.S.A.).

- Version 2.0
- Programming language: C++

3.7. SOPRANO_scenarios

- Participant: A. Chiche, J. Ch. Gilbert, M. Porcheron
- Version: 0.1.
- Programming language: C++.
- Solves the medium-term electricity planning problem defined on a scenario tree by the Progressive Hedging algorithm.
5. Software

5.1. DenseMotion software - Estimation of 2D dense motion fields

Participants: Thomas Corpetti, Patrick Héas, Etienne Mémin.

This code allows the computation from two consecutive images of a dense motion field. The estimator is expressed as a global energy function minimization. The code enables the choice of different data model and different regularization functional depending on the targeted application. Generic motion estimator for video sequences or dedicated motion estimator for fluid flows can be specified. This estimator allows in addition the users to specify additional correlation based matching measurements. It enables also the inclusion of a temporal smoothing prior relying on a velocity vorticity formulation of the Navier-Stoke equation for Fluid motion analysis applications. The different variants of this code correspond to research studies that have been published in IEEE transaction on Pattern Analysis and machine Intelligence, Experiments in Fluids, IEEE transaction on Image Processing, IEEE transaction on Geo-Science end Remote Sensing. The binary of this code can be freely downloaded on the FLUID web site http://fluid.irisa.fr.

5.2. 2DLayeredMotion software - Estimation of 2D independent mesoscale layered atmospheric motion fields

Participants: Patrick Héas, Etienne Mémin.

This software enables to estimate a stack of 2D horizontal wind fields corresponding to a mesoscale dynamics of atmospheric pressure layers. This estimator is formulated as the minimization of a global energy function. It relies on a vertical decomposition of the atmosphere into pressure layers. This estimator uses pressure data and classification clouds maps and top of clouds pressure maps (or infra-red images). All these images are routinely supplied by the EUMETSAT consortium which handles the Meteosat and MSG satellite data distribution. The energy function relies on a data model built from the integration of the mass conservation on each layer. The estimator also includes a simplified and filtered shallow water dynamical model as temporal smoother and second-order div-curl spatial regularizer. The estimator may also incorporate correlation-based vector fields as additional observations. These correlation vectors are also routinely provided by the Eumetsat consortium. This code corresponds to research studies published in IEEE transaction on Geo-Science and Remote Sensing. It can be freely downloaded on the FLUID web site http://fluid.irisa.fr.

5.3. 3DLayeredMotion software - Estimation of 3D interconnected layered atmospheric motion fields

Participants: Patrick Héas, Etienne Mémin.

This software extends the previous 2D version. It allows (for the first time to our knowledge) the recovery of 3D wind fields from satellite image sequences. As with the previous techniques, the atmosphere is decomposed into a stack of pressure layers. The estimation relies also on pressure data and classification clouds maps and top of clouds pressure maps. In order to recover the 3D missing velocity information, physical knowledge on 3D mass exchanges between layers has been introduced in the data model. The corresponding data model appears to be a generalization of the previous data model constructed from a vertical integration of the continuity equation. This research study has been recently accepted for publication in IEEE trans. on Geo-Science and Remote Sensing. A detailed description of the technique can be found in an Inria research report. The binary of this code can be freely downloaded on the FLUID web site http://fluid.irisa.fr.
5.4. Low-Order-Motion - Estimation of low order representation of fluid motion

Participants: Anne Cuzol, Etienne Mémin.

This code enables the estimation of a low order representation of a fluid motion field from two consecutive images. The fluid motion representation is obtained using a discretization of the vorticity and divergence maps through regularized Dirac measure. The irrotational and solenoidal components of the motion fields are expressed as linear combinations of basis functions obtained through the Biot-Savart law. The coefficient values and the basis function parameters are obtained as the minimizer of a functional relying on an intensity variation model obtained from an integrated version of the mass conservation principle of fluid mechanics. Different versions of this estimation are available. The code which includes a Matlab user interface can be downloaded on the FLUID web site http://fluid.irisa.fr. This program corresponds to a research study that has been published in the International Journal on computer Vision.
5. Software

5.1. Deformable Registration Software

**Participants:** Nikos Paragios [Correspondent], Ben Glocker, Aristeidis Sotiras, Nikos Komodakis.

DROP is a deformable registration platform in C++ for the medical imaging community (publicly available at [http://www.mrf-registration.net](http://www.mrf-registration.net)) developed mainly at Ecole Centrale, Technical University of Munich and University of Crete. This is the first publicly available platform which contains most of the existing metrics to perform registration under the same concept. The platform is used for clinical research from approximately 3,000 users worldwide.

5.2. Fast Primal Dual Strategies for Optimization of Markov Random Fields

**Participants:** Nikos Komodakis [Correspondent], Nikos Paragios, George Tziritas.

FASTPD is an optimization platform in C++ for the computer vision and medical imaging community (publicly available at [http://www.csd.uoc.gr/~komod/FastPD/](http://www.csd.uoc.gr/~komod/FastPD/)) developed mainly at Ecole Centrale and University of Crete. This is the most efficient publicly available platform in terms of a compromise of computational efficiency and ability to converge to a good minimum for the optimization of generic MRFs. The platform is used from approximately 1,500 users worldwide.

5.3. imaGe-based Procedural Modeling Using Shape Grammars

**Participants:** Olivier Teboul [Correspondent], Iasonas Kokkinos, Panagiotis Koutsourakis, Loic Simon, Nikos Paragios.

GRAPE is a generic image parsing library based on re-inforcement learning. It can handle grammars (binary-split, four-color, Hausmannian) and image-based rewards (Gaussian mixtures, Randomized Forests) of varying complexity while being modular and computationally efficient both in terms of grammar and image rewards. The platform is used from approximately 500 users worldwide.

5.4. Texture Analysis Using Modulation Features and Generative Models

**Participants:** Iasonas Kokkinos [Correspondent], Georgios Evangelopoulos.

TEXMED is a front-end for texture analysis and edge detection platform in Matlab that relies on Gabor filtering and image demodulation (publicly available at [http://cvsp.cs.ntua.gr/software/texture/](http://cvsp.cs.ntua.gr/software/texture/)). Includes frequency- and time-based definition of Gabor- and other Quadrature-pair filterbanks, demodulation with the Regularized Energy Separation Algorithm and Texture/Edge/Smooth classification based on MDL criterion. The platform is used from approximately 250 users worldwide.
4. Software

4.1. Genetic Network Analyzer (GNA)

Participants: Hidde de Jong [Correspondent], Michel Page, François Rechenmann, Delphine Ropers.

GENETIC NETWORK ANALYZER (GNA) is the implementation of a method for the qualitative modeling and simulation of gene regulatory networks developed in the IBIS project. The input of GNA consists of a model of the regulatory network in the form of a system of piecewise-linear differential equations, supplemented by inequality constraints on the parameters and initial conditions. From this information, GNA generates a state transition graph summarizing the qualitative dynamics of the system. In order to analyze large graphs, GNA allows the user to specify properties of the qualitative dynamics of a network in temporal logic, using high-level query templates, and to verify these properties on the state transition graph by means of standard model-checking tools, either locally installed or accessible through a remote web server. GNA is currently distributed by the company Genostar, but remains freely available for academic research purposes. The current version is GNA 8.3. In comparison with the previously distributed versions, GNA 8.3 has the following additional functionalities. First, it supports the editing and visualization of regulatory networks, in an SBGN-compatible format, and second it semi-automatically generates a prototype model from the network structure, thus accelerating the modeling process. For more information, see http://www-helix.inrialpes.fr/gna.

4.2. WellReader

Participants: Guillaume Baptist, Johannes Geiselmann, Jérôme Izard, Hidde de Jong [Correspondent], Delphine Ropers.

WELLREADER is a program for the analysis of gene expression data obtained by means of fluorescent and luminescent reporter genes. WELLREADER reads data files in an XML format or in a format produced by microplate readers, and allows the user to detect outliers, perform background corrections and spline fits, compute promoter activities and protein concentrations, and compare expression profiles across different conditions. WELLREADER has been written in MATLAB and is available under an LGPL licence, both as source code (M files) and compiled code (platform-specific binary files). For more information, see: http://ibis.inrialpes.fr/article957.html.
MACS Project-Team

5. Software

5.1. FELISCE

Participants: Dominique Chapelle, Jérémie Foulon [correspondant], Philippe Moireau, Marina Vidrascu.

FELISCE – standing for “Finite Elements for LIfe SCiences and Engineering” – is a new finite element code which the MACS and REO teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library. https://gforge.inria.fr/projects/felisce/

5.2. HeartLab

Participants: Matthieu Caruel, Radomir Chabiniok, Dominique Chapelle, Alexandre Imperiale, Philippe Moireau [correspondant].

The heartLab software is a library written in Matlab and C (mex functions) designed to perform both simulation and estimation (based on various types of measurements, e.g. images) of the heart mechanical behavior. Started in 2006, it is already quite large (about 60,000 lines), and is used within the CardioSense3D community.

The code relies on OpenFEM for the finite element computations, and the implementation was performed with a particular concern for modularity, since modeling and estimation use the same finite element operators. This modularity also allows to couple the code with other FEM solvers, such as LifeV and Mistral developed in the Reo team-project. In particular, we are now able to include perfusion and electrical coupling with LifeV using PVM, and fluid-structure interaction using Mistral.

We also included geometric data and tools in the code to define heart anatomical models compatible with the simulation requirements in terms of mesh quality, fiber direction data defined within each element, and referencing necessary for handling boundary conditions and estimation, in particular. These geometries are analytical or come from computerized tomography (CT) or magnetic resonance (MR) image data of humans or animals.

We recently incorporated numerous non-linear data assimilation observation operators based on medical imaging post-processing to be able to now perform estimation with a large variety of medical imaging modalities.

The Library is now 64 bits compatible with the help of the Cesare Corrado from Reo.

5.3. MITCNL

Participants: Dominique Chapelle, Marina Vidrascu [correspondant].

The package MITCNL is a set of subroutines that implements the triangular MITC3, MITC6 and quadrilateral MITC4 and MITC9 shell elements for large displacements [14]. We use it as a basis for new developments of shell elements, in particular within Modulef. It can be easily interfaced with most finite element codes as well. We also license this package to some of our partners for use with their own codes.

5.4. MODULEF

Participant: Marina Vidrascu [correspondant].
Most of the software developed in our team is integrated in the Modulef library. Modulef is designed to provide building blocks for effective and reliable software development in finite element analysis. Well-adapted rigorous data structures and ease of integration (for new methods or algorithms) are some of its key advantages. Until 1998, Modulef was distributed by the Simulog company within a club structure (for a membership fee). In order to encourage its dissemination, its status was then changed to make it freely available. It can be downloaded at no charge from the INRIA-Rocquencourt web site (http://www-rocq.inria.fr/modulef/).

5.5. OpenFEM: a Finite Element Toolbox for Matlab and Scilab

Participants: Dominique Chapelle, Philippe Moireau [correspondant].

OpenFEM (http://www.openfem.net) is an opensource finite element toolbox for linear and nonlinear structural mechanics within the Matlab and Scilab matrix computing environments. This software is developed in a collaboration between Macs and the SDTools company¹. Performing finite element analyses within a matrix computing environment is of considerable interest, in particular as regards the ease of new developments, integration of external software, portability, post-processing, etc.

This Library is the core of the finite element computations of HeartLab where a specific version have been developed with the help of Cesare Corrado from Reo.

5.6. SHELDDON

Participants: Dominique Chapelle, Marina Vidrascu [correspondant].

SHELDDON (SHELls and structural Dynamics with DOmain decomposition in Nonlinear analysis) is a finite element library based on the Modulef package which contains shell elements, nonlinear procedures and PVM subroutines used in domain decomposition or coupling methods.

5.7. Verdandi

Participants: Dominique Chapelle, Marc Fragu [correspondant], Vivien Mallet, Philippe Moireau.

Verdandi is an opensource (LGPL) software library aiming at providing assimilation data methods and related tools. Mainly targeted at large systems arising from the discretization of PDEs, it is intentionally devised as generic, which allows for applications in a wide range of problems (biology and medicine, environment, image processing...). See also the web page http://verdandi.gforge.inria.fr/, with a complete documentation in English. The first stable version (1.0) was released in June and contains most of the major data assimilation algorithms of both variational and sequential types. Moreover, some specific developments are performed with particular regard to cardiac modeling applications, as Verdandi is partly funded by – and distributed within – the euHeart project.

- ACM: Mathematical software
- AMS: System theory; control
- Software benefit: Verdandi est la seule bibliothèque d’assimilation de données générique.
- License: LGPL (2.1 or any later version)
- Type of human computer interaction: Ligne de commande et fichiers de configuration
- OS/Middelware: Linux, MacOS ou Windows
- Required library or software: Seldon (LGPL, http://seldon.sourceforge.net/)
- Documentation: Chaque fonction est documentée, grâce à Doxygen. Il y a aussi un guide d’utilisation (en cours de rédaction actuellement). Toute la documentation est en anglais.

¹ http://www.sdtools.com
5. Software

5.1. SPECFEM3D

The MAGIQUE-3D project is based (in part) on existing software packages, which are already validated, portable and robust. The SPECFEM3D software package, developed by Dimitri Komatitsch and his colleagues in collaboration with Jeroen Tromp and his colleagues at the California Institute of Technology and at Princeton University (USA), and which is still actively maintained by Dimitri Komatitsch and his colleagues, allows the precise modeling of seismic wave propagation in complex three-dimensional geological models. Phenomena such as anisotropy, attenuation (i.e., anelasticity), fluid-solid interfaces, rotation, self-gravitation, as well as crustal and mantle models can be taken into account. The software is written in Fortran95 with MPI message-passing on parallel machines. It won the Gordon Bell Prize for best performance of the Supercomputing’2003 conference. In 2006, Dimitri Komatitsch established a new collaboration with the Barcelona Supercomputing Center (Spain) to work on further optimizing the source code to prepare it for very large runs on future petaflops machines to solve either direct or inverse problems in seismology. Optimizations have focused on improving load balancing, reducing the number of cache misses and switching from blocking to non-blocking MPI communications to improve performance on very large systems. Because of its flexibility and portability, the code has been run successfully on a large number of platforms and is used by more than 150 academic institutions around the world. In November 2008 this software package was again among the six finalists of the prestigious Gordon Bell Prize of the SuperComputing’2008 conference in the USA [50] for a calculation performed in parallel on 150,000 processor cores, reaching a sustained performance level of 0.16 petaflops.

5.2. Hou10ni

This software, written in FORTRAN 90, simulates the propagation of acoustic waves in heterogeneous 2D and 3D media. It is based on an Interior Penalty Discontinuous Galerkin Method (IPDGM). The 2D version of the code has been implemented in the Reverse Time Migration (RTM) software of TOTAL in the framework of the PhD. thesis of Caroline Baldassari and the 3D version should be implemented soon. The 2D code allows for the use of meshes composed of cells of various order (p-adaptivity in space). For the time discretization, we used the local time stepping strategy described at section 3.2, item High-Order Schemes in Space and Time which permits not only the use of different time-step, but also to adapt the order of the time-discretization to the order of each cells (h/p-adaptivity in time). These functionalities will be soon implemented in the 3D code. The main competitors of Hou10ni are codes based on Finite Differences, Spectral Element Method or other Discontinuous Galerkin Methods (such as the ADER schemes). During her PhD. thesis, Caroline Baldassari compared the solution obtained by Hou10ni to the solution obtained by a Finite Difference Method and by a Spectral Element Method (SPECFEM). To evaluate the accuracy of the solutions, we have compared it to analytical solutions provided by the codes Gar6more (see below). The results of these comparisons is: a) that Hou10ni outperforms the Finite Difference Methods both in terms of accuracy and of computational burden and b) that its performances are similar to Spectral Element Methods. Since Hou10ni allows for the use of meshes based on tetraedrons, which are more appropriate to mesh complex topographies, and for the p-adaptivity, we decided to implement it in the RTM code of TOTAL. Of course, we also used these comparisons to validate the code. Now, it remains to compare the performances of Hou10ni to the ADER schemes.

5.3. Gar6more3D

Participants: Julien Diaz [correspondant], Abdelaâziz Ezziani.
This code computes the analytical solution of problems of waves propagation in two layered 3D media such as:
- acoustic/acoustic-
- acoustic/elastodynamic-
- acoustic/porous-
- porous/porous,
based on the Cagniard-de Hoop method.

See also the web page [http://web.univ-pau.fr/~jdiaz1/softwares.html](http://web.univ-pau.fr/~jdiaz1/softwares.html).

The main objective of this code is to provide reference solutions in order to validate numerical codes. They have been already used by J. Tromp and C. Morency to validate their code of poroelastic wave propagation [67]. They are freely distributed under a CECILL licence and can be downloaded on the website [http://web.univ-pau.fr/~jdiaz1/softwares.html](http://web.univ-pau.fr/~jdiaz1/softwares.html). As far as we know, the main competitor of this code is EX2DELDEL ([available on http://www.spice-rtn.org](http://www.spice-rtn.org)), but this code only deals with 2D acoustic or elastic media. Our codes seem to be the only one able to deal with bilayered poroelastic media and to handle the three dimensional cases.

- **ACM:** J.2
- **AMS:** 34B27 35L05 35L15 74F10 74J05
- **Programming language:** Fortran 90
5. Software

5.1. Inria Bioscience Resources

Participants: Olivier Collin [correspondant], Frédéric Cazals, Mireille Régnier, Marie-France Sagot, Hélène Touzet, Hidde de Jong, David Sherman, Marie-Dominique Devignes, Dominique Lavenier.

Inria Bioscience Resources is a portal designed to improve the visibility of bioinformatics tools and resources developed by Inria teams. This portal will help the community of biologists and bioinformatics understand the variety of bioinformatics projects in Inria, test the different applications, and contact project-teams. Eight project-teams participate in the development of this portal. Inria Bioscience Resources is developed in an Inria Technology Development Action (ADT).

5.2. Magus: Collaborative Genome Annotation

Participants: David James Sherman [correspondant], Pascal Durrens, Natalia Golenetskaya, Florian Lajus, Tiphaine Martin.

As part of our contribution the Génolevures Consortium, we have developed over the past few years an efficient set of tools for web-based collaborative annotation of eukaryote genomes. The MAGUS genome annotation system integrates genome sequences and sequences features, in silico analyses, and views of external data resources into a familiar user interface requiring only a Web navigator. MAGUS implements the annotation workflows and enforces curation standards to guarantee consistency and integrity. As a novel feature the system provides a workflow for simultaneous annotation of related genomes through the use of protein families identified by in silico analyses; this has resulted in a three-fold increase in curation speed, compared to one-at-a-time curation of individual genes. This allows us to maintain Génolevures standards of high-quality manual annotation while efficiently using the time of our volunteer curators.

MAGUS is built on: a standard sequence feature database, the Stein lab generic genome browser [55], various biomedical ontologies (http://obo.sf.net), and a web interface implementing a representational state transfer (REST) architecture [35].

For more information see magus.gforge.inria.fr, the MAGUS Gforge web site. MAGUS is developed in an Inria Technology Development Action (ADT).

5.3. YAGA: Yeast Genome Annotation

Participants: Pascal Durrens, Tiphaine Martin [correspondant].

With the arrival of new generations of sequencers, laboratories, at a lower cost, can be sequenced groups of genomes. You can no longer manually annotate these genomes. The YAGA software’s objective is to syntactically annotate a raw sequence (genetic element: gene, CDS, tRNA, centromere, gap, ...) and functionally as well as generate EMBL files for publication. The annotation takes into account data from comparative genomics, such as protein family profiles.

After determining the constraints of the annotation, the YAGA software can automatically annotate de novo all genomes from their raw sequences. The predictors used by the YAGA software can also take into account the data RNAseq to reinforce the prediction of genes. The current settings of the software are intended for annotation of the genomes of yeast, but the software is adaptable for all types of species.

5.4. BioRica: Multi-scale Stochastic Modeling

Participants: David James Sherman [correspondant], Rodrigo Assar Cuevas, Alice Garcia.
BioRica is a high-level modeling framework integrating discrete and continuous multi-scale dynamics within the same semantics field. A model in BioRica node is hierarchically composed of nodes, which may be existing models. Individual nodes can be of two types:

- Discrete nodes are composed of states, and transitions described by constrained events, which can be non deterministic. This captures a range of existing discrete formalisms (Petri nets, finite automata, etc.). Stochastic behavior can be added by associating the likelihood that an event fires when activated. Markov chains or Markov decision processes can be concisely described. Timed behavior is added by defining the delay between an event’s activation and the moment that its transition occurs.

- Continuous nodes are described by ODE systems, potentially a hybrid system whose internal state flows continuously while having discrete jumps.

The system has been implemented as a distributable software package

The BioRica compiler reads a specification for hierarchical model and compiles it into an executable simulator. The modeling language is a stochastic extension to the AltaRica Dataflow language, inspired by work of Antoine Rauzy. Input parsers for SBML 2 version 4 are currently being validated. The compiled code uses the Python runtime environment and can be run stand-alone on most systems [ 36 ].

For more information see biorica.gforge.inria.fr , the BioRica Gforge web site. BioRica was developed as an Inria Technology Development Action (ADT).

5.5. Pathtastic: Inference of whole-genome metabolic models

**Participants:** David James Sherman [correspondant], Pascal Durrens, Nicolás Loira, Anna Zhukova.

Pathtastic is a software tool for inferring whole-genome metabolic models for eukaryote cell factories. It is based on metabolic scaffolds, abstract descriptions of reactions and pathways on which inferred reactions are hung are are eventually connected by an interactive mapping and specialization process. Scaffold fragments can be repeatedly used to build specialized subnetworks of the complete model.

Pathtastic uses a consensus procedure to infer reactions from complementary genome comparisons, and an algebra for assisted manual editing of pathways.

For more information see pathtastic.gforge.inria.fr , the Pathtastic Gforge web site.

5.6. Génolevures On Line: Comparative Genomics of Yeasts

**Participants:** David James Sherman, Pascal Durrens [correspondant], Natalia Golenetskaya, Tiphaine Martin.

The Génolevures online database provides tools and data for exploring the annotated genome sequences of more than 20 genomes, determined and manually annotated by the Génolevures Consortium to facilitate comparative genomic studies of hemiascomycetous yeasts. Data are presented with a focus on relations between genes and genomes: conservation of genes and gene families, speciation, chromosomal reorganization and synteny. The Génolevures site includes an area for specific studies by members of its international community.

Génolevures online uses the MAGUS system for genome navigation, with project-specific extensions developed by David Sherman, Pascal Durrens, and Tiphaine Martin. An advanced query system for data mining in Génolevures is being developed by Natalia Golenetskaya. The contents of the knowledge base are expanded and maintained by the CNRS through GDR 2354 Génolevures. Technical support for Génolevures On Line is provided the CNRS through UMR 5800 LaBRI.

For more information see genolevures.org , the Génolevures web site.
MASAIE Project-Team (section vide)
MODEMIC Team

5. Software

5.1. VITELBIO

Participants: Jérôme Harmand, Alain Rapaport.

VITELBIO (VIrtual TELluric BIOreactors) is a simulation tool for studying networks of interconnected chemostats with the objective of mimicking microbial activities in soil. The software, developed with the help of ITK Company, is accessible on a server from any web navigator and make use of Flex for the user interface and Octave for the numerical integration. An important effort has been made for obtaining a pleasant and easy interface that is appealing for microbiologists: the network can be drawn graphically on the screen and simulation results can be easily compared between (virtual) experiments, superposing trajectories curves.

This software is used by several researchers, from LBE (INRA Narbonne), UMR Eco & Sols (Montpellier), UREP (INRA Theix), Biomeco (Paris-Grignon), UMR EGC (Paris-Grignon).... and also as a teaching support. See the web page http://sites.google.com/site/vitelbio/
5. Software

5.1. Adaptive Grid Refinement

Participants: Laurent Debreu, Marc Honnorat.

AGRIF (Adaptive Grid Refinement In Fortran, [71]) is a Fortran 90 package for the integration of full adaptive mesh refinement (AMR) features within a multidimensional finite difference model written in Fortran. Its main objective is to simplify the integration of AMR potentialities within an existing model with minimal changes. Capabilities of this package include the management of an arbitrary number of grids, horizontal and/or vertical refinements, dynamic regridding, parallelization of the grids interactions on distributed memory computers. AGRIF requires the model to be discretized on a structured grid, like it is typically done in ocean or atmosphere modelling. As an example, AGRIF is currently used in the following ocean models: MARS (a coastal model developed at IFREMER-France), ROMS (a regional model developed jointly at Rutgers and UCLA universities), OPA-NEMO ocean modelling system (a general circulation model used by the French and European scientific community) and HYCOM (a regional model developed jointly by University of Miami and the French Navy).

In 2011, a new contract has been signed with IFREMER to optimize parallel capabilities of the software. AGRIF is licensed under a GNU (GPL) license and can be downloaded at its web site (http://ljk.imag.fr/MOISE/AGRIF/index.html).

5.2. DatIce

Participants: Bénédicte Lemieux-Dudon, Habib Toye Mahamadou Kele.

The Datice code ([76], [77]) is designed to estimate consistent chronologies of several deep ice cores (i.e., depth-age relationships of the ice matrix and trapped gas). A cost function derived from Bayes theorem puts in competition the chronological constraints brought by heterogeneous observations (stratigraphic links between cores, gas and ice age markers, delta-depth markers, etc.), and the background dating scenarios simulated with glaciological models (fimn densification and ice flow models). The minimization of the cost function provides optimal estimations of three key quantities from which dating scenarios can be derived: the past accumulation rate, the close-off depth which is the depth where the gas is trapped into ice, and the total thinning function. Uncertainties of the analysed dating scenarios (key quantities and chronologies) are assessed on the basis of the Bayesian formulation. This approach is innovative because:

- it relies on data assimilation techniques to calculate ice core chronologies and uncertainties;
- it applies to a large number of heterogeneous observations;
- it ensures consistency between the chronologies of several cores and the consistency between the gas and ice age scales.

The code has been used in several recent publications (see [68], [87] for example).

5.3. SDM toolbox

Participant: Antoine Rousseau.
The computation of the wind at small scale and the estimation of its uncertainties is of particular importance for applications such as wind energy resource estimation. To this aim, we develop a new method based on the combination of an existing numerical weather prediction model providing a coarse prediction, and a Lagrangian Stochastic Model adapted from a pdf method introduced by S.B. Pope for turbulent flows. This Stochastic Downscaling Method (SDM http://sdm.gforge.inria.fr/ ) is thus aimed to be used as a refinement toolbox of large-scale numerical models. SDM requires a specific modelling of the turbulence closure, and involves various simulation techniques whose combination is totally new (such as Poisson solvers, optimal transportation mass algorithm, original Euler scheme for confined Langevin stochastic processes, and stochastic particle methods). In 2011, we worked on the comparison of the SDM model (endowed with a physical geostrophic forcing and a wall log law) with simulations obtained with a LES method (Méso-NH code) for the atmospheric boundary layer (from 0 to 750 meters in the vertical direction), in the neutral case, see [ 58 ].

5.4. CompModSA package

Alexandre Janon is a contributor of the package CompModSA - Sensitivity Analysis for Complex Computer Models (see http://cran.r-project.org/web/packages/CompModSA/index.html ). This package is useful for conducting sensitivity analysis of complex computer codes when model evaluations are somewhat expensive (e.g. take longer than a couple of seconds to run) but a reasonable number (50 or more) of model evaluations can be obtained at sampled input values.

5.5. NEMO-TAM

Tangent and adjoint models for the NEMO platform of the oceanic modelling that have been developed by the MOISE team have been published now under Cecill license and distributed by the NEMO consortium.
4. Software

4.1. Virtual Retina: A Large-Scale Simulator of Biological Retina

Participants: Bruno Cessac, Hassan Nasser, Pierre Kornprobst [correspondent], Adrien Wohrer [Group for Neural Theory - ENS].

Virtual Retina is a simulation software developed by Adrien Wohrer during his PhD [74], [73] that allows large-scale simulations of biologically-plausible retinas.

Virtual Retina has a variety of biological features implemented such as (i) spatio-temporal linear filter implementing the basic center/surround organization of retinal filtering, (ii) non-linear contrast gain control mechanism providing instantaneous adaptation to the local level of contrast; (iii) spike generation by one or several layers of ganglion cells paving the visual field.

Virtual Retina is under INRIA CeCill C open-source licence, so that one can download it, install it and run it on your own sequences. Virtual Retina also offers a web service (v 2.1), so that you may test directly the main software on user’s own data, without any installation. This webservice was developed in collaboration with Nicolas Debeissat (engineer, 2002).

Virtual Retina continues its evolution thanks to the work done in our team by Bruno Cessac and Hassan Nasser who are interested in the analysis of the collective behavior of ganglion cells responses (see Section 5.1.1).

To take this collective behavior into account, Virtual Retina needs to be extended since in its current version, ganglion cells are independent. Other evolutions of Virtual Retina are also investigated by external partners (see, e.g., [68]).

- IDDN number: IDDN.FR.001.210034.000.S.P.2007.000.31235
- Version: v 2.2.2 (September 2011)
- Link: http://www-sop.inria.fr/neuromathcomp/public/software/virtualretina

4.2. ABFilter: A Simulator Of V1 Simple and Complex Cells

Participants: Pierre Kornprobst [correspondent], Maria-Jose Escobar [Electronics Engineering Department of the Universidad Técnica Federico Santa María, Valparaíso, Chile.], Adrien Wohrer [Group for Neural Theory - ENS].

ABFilter is a C++-library that allows the implementation of spatiotemporal filtering in video sequences using filters proposed by Adelson and Bergen in [62].

Simple cells in V1 are characterized by linear receptive fields where the neuron response is a weighted linear combination of the input stimulus inside its receptive field. By combining two simple cells in a linear manner it is possible to get direction-selective cells. The direction-selectivity refers to the property of a neuron to respond to the direction of the stimulus motion. The way to model this selectivity is by obtaining receptive fields oriented in space and time. Some characteristics of V1 complex cells can be explained using a nonlinear combination of V1 simple cells as it has been proposed by, e.g., Adelson and Bergen in [62]. Implementing these cells properly is a difficult problem and this library offers the possibility to easily implement a V1 layer which can serve as an input to subsequent cortical areas such as MT (see, e.g., the architecture developed in [65]).

The ABFilter library is under a CeCill-C open-source license.

- IDDN.FR.001.280017.000.S.P.2011.000.31235
- Version: v 1.0 (May 2011)
- Download: http://www-sop.inria.fr/neuromathcomp/public/software/abfilter-1.0.tar.gz
4.3. MotionLib: A Neural-Fields Model for Motion Estimation

Participants: Pierre Kornprobst [correspondant], Emilien Tlapale.

MotionLib implements the neural field model of motion estimation described in [52], using the Python programming language.

Motion integration is the core of the model: It implements a two-layer model with feedbacks that selects and diffuses motion signals. The main structure has been implemented here, allowing contributors to enrich this model easily. Several tools are also provided to visualize and analyze the distributed velocity fields obtained by this approach.

- License IDDN.FR.001.210029.000.S.P.2011.000.31235
- Version: v1.0 (October 2011)
- Download: http://www-sop.inria.fr/neuromathcomp/public/software/motion.zip

4.4. Event neural assembly Simulation

Participants: Frederic Alexandre [INRIA Cortex Nancy], Bruno Cessac [correspondent], Rodrigo Cofre Torres, Jeremy Fix [INRIA Cortex Nancy], Olivier Rochel [INRIA Cortex Nancy], Sélim Kraria, Olivier Marre, Hassan Nasser, Horacio Rostro-Gonzalez, Vivien Robinet, Thierry Viéville [INRIA Cortex Nancy], Juan-Carlos Vasquez.

Enas is a library providing numerical tools for the simulation of neural networks and the analysis of spike trains either coming from neural simulators or from biological experiments.

It is designed mainly as

- An existing simulator plug-in (e.g. MVASpike or other simulators via the NeuralEnsemble meta-simulation platform),
- Additional modules for computations with neural unit assembly on standard platforms (e.g. Python, Matlab or the Scilab platform).
- Original modules for the analysis of spike train statistics intended to be used by the neuroscientists community.

Achievements include:

- Spike trains statistical analysis via Gibbs distributions. They are based on the estimation of a parametric Gibbs potential optimaly characterizing the statistics of empirical spike trains (by minimisation of the Kullback-Leibler divergence between the empirical measure and the Gibbs measure). From this, classical statistical indicators such as firing rate, correlations, higher order moments and statistical entropy are obtained. Also, the form of the Gibbs potential provides essential informations on the underlying neural network and its structure. This method does not only allows us to estimate the spikes statistics but also to compare different models, thus answering such questions about the neural code as: are correlations (or time synchrony or a given set of spike patterns,. . . ) significant with respect to rate coding?
- Spiking network programing for exact event’s sequence restitution;
- Discrete neural field parameters algorithmic adjustments and time-constrained event-based network simulation reconciling clock and event based simulation methods.

Compared to existing libraries Enas offers new computational methods taking into account time constraints in neural networks (such as memory effects), based on theoretical methods rooted in statistical physics and applied mathematics. The algorithms used are based on linear programming, nonlinear parameter estimations, statistical methods. The C/C++ code has been organized as “bean java” to ease its use by programmers non specialized in advanced object programming. As a consequence the code is distributed in the form of an include source for the lightest and the most universal integration into users codes. The standard algorithms are based on the best free libraries in the domain such as gsl http://www.gnu.org/software/gsl .
Event neural assembly simulation is developed in gForge. It is under CeCILL C licence
APP logiciel Enas: IDDN.FR.OO1.360008.000.S.P.2009.000.10600.

Its development as a friendly software designed for the neuroscience community is our next purpose (ADT proposal).

Website: http://enas.gforge.inria.fr/
5. Software

5.1. Zebre

**Participant:** Thierry Dumont [correspondant].

Thierry Dumont is currently developing a toolbox to solve stiff reaction diffusion equations using splitting methods, together with refined numerical schemes for ODEs (RADO 5).

5.2. OptimChemo

**Participants:** Violaine Louvet [correspondant], Emmanuel Grenier.

OptimChemo is a user-friendly software designed to study numerically the effect of multiple chemotherapies on simple models of tumour growth and to optimize chemotherapy schedules.
5. Software

5.1. Mayavi

**Participant:** Gaël Varoquaux [Correspondant].

Mayavi is the most used scientific 3D visualization Python software (http://mayavi.sourceforge.net/). It has been developed by Prabhu Ramachandran (IIT Bombay) and Gaël Varoquaux (PARIETAL, INRIA Saclay). Mayavi can be used as a visualization tool, through interactive command line or as a library. It is distributed under Linux through Ubuntu, Debian, Fedora and Mandriva, as well as in PythonXY and EPD Python scientific distributions. Mayavi is used by several software platforms, such as PDE solvers (fipy, sfepy), molecule visualization tools (http://pyrx.scripps.edu) and brain connectivity analysis tools (connectomeViewer).

See also the web page http://mayavi.sourceforge.net/ and the following paper http://hal.inria.fr/inria-00528985/en.

- **Version:** 3.4.0

5.2. Nipy

**Participants:** Bertrand Thirion [correspondant], Virgile Fritsch, Gaël Varoquaux.

Nipy is an open-source Python library for neuroimaging data analysis, developed mainly at Berkeley, Stanford, MIT and Neurospin. It is open to any contributors and aims at developing code and tools sharing. Some parts of the library are completely developed by Parietal and LNAO (CEA, DSV, Neurospin). It is devoted to algorithmic solutions for various issues in neuroimaging data analysis. All the nipy project is freely available, under BSD licence. It is available in NeuroDebian.

See also the web page http://nipy.org.

- **Version:** 0.2

5.3. MedINRIA

**Participants:** Pierre Fillard [correspondant], Sergio Medina, Viviana Siless.

MedINRIA is a free collection of softwares developed within the ASCLEPIOS, ATHENA and VISAGES research projects. It aims at providing to clinicians state-of-the-art algorithms dedicated to medical image processing and visualization. Efforts have been made to simplify the user interface, while keeping high-level algorithms. MedINRIA is available for Microsoft windows XP/Vista, Linux Fedora Core, MacOSX, and is fully multithreaded.

See also the web page http://med.inria.fr/.

- **Version:** 2.0

5.4. Scikit learn

**Participants:** Bertrand Thirion [correspondant], Gaël Varoquaux, Alexandre Gramfort, Fabian Pedregosa, Virgile Fritsch.
Scikit-learn is open-source a machine learning toolkit written in Python/C that provides generic tools to learn information for the classification of various kinds of data, such as images or texts. It is tightly associated to the scientific Python software suite (numpy/scipy) for which it aims at providing a complementary toolkit for machine learning (classification, clustering, dimension reduction, regression). There is an important focus on code quality (API consistency, code readability, tests, documentation and examples), and on efficiency, as the scikit-learn compares favorably to state-of-the-art modules developed in R in terms of computation time or memory requirements. Scikit-learn is currently developed by about 30 contributors, but the core developer team has been with the Parietal INRIA team at Saclay-Île-de- France since January 2010. The scikit-learn has recently become the reference machine learning library in Python.

- Version: 0.9
- Programming language: Python, C/Cython
5. Software

5.1. LiFE-V library

Participants: Miguel Ángel Fernández Varela [correspondant], Jean-Frédéric Gerbeau.

LiFE-V\(^2\) is a finite element library providing implementations of state of the art mathematical and numerical methods. It serves both as a research and production library. LiFE-V is the joint collaboration between three institutions: Ecole Polytechnique Fédérale de Lausanne (CMCS) in Switzerland, Politecnico di Milano (MOX) in Italy and INRIA (REO) in France. It is a free software under LGPL license.

5.2. Mistral library

Participants: Cristóbal Bertoglio Beltran, Jean-Frédéric Gerbeau [correspondant], Vincent Martin, Joaquín-Alejandro Mura Mardones.

Mistral is a finite element library which implements in particular fluid-structure interaction algorithms (ALE and Fictitious domain formulations), fluid surface flow (ALE) and incompressible magnetohydrodynamics equations. Mistral results from a collaboration between INRIA and ENPC (CERMICS).

5.3. FELiScE

Participants: Grégory Arbia, Cesare Corrado, Miguel Ángel Fernández Varela, Justine Fouchet-Incaux, David Froger, Jean-Frédéric Gerbeau [correspondant], Damiano Lombardi, Elisa Schenone, Saverio Smaldone.

FELiScE – standing for “Finite Elements for Life Sciences and Engineering” – is a new finite element code which the MACS and REO project-teams have decided to jointly develop in order to build up on their respective experiences concerning finite element simulations. One specific objective of this code is to provide in a unified software environment all the state-of-the-art tools needed to perform simulations of the complex cardiovascular models considered in the two teams – namely involving fluid and solid mechanics, electrophysiology, and the various associated coupling phenomena. FELISCE is written in C++, and may be later released as an opensource library. https://gforge.inria.fr/projects/felisce/

\(^{2}\) http://www.lifev.org/
5. Software

5.1. SOFA

SOFA, the Simulation Open Framework Architecture, is an international, multi-institution, collaborative initiative, aimed at developing a flexible and open source framework for interactive simulations. This will eventually establish new grounds for a widely usable standard system for long-term research and product prototyping, ultimately shared by many academic and industrial sites. Over the last two years, the SOFA framework has evolved from an informal collaborative work between the Sim Group at CIMIT, the Alcove, Asclepios and Evasion teams at INRIA into a more structured development project. By proposing a unique architecture allowing the integration of the multiple competencies required for the development of a medical training system, we believe it will be possible to accelerate and foster research activities in the field of interactive medical simulation. The main objectives of the SOFA framework are:

- Simplify the development of medical simulation systems by improving interoperability
- Evaluate and validate new algorithms
- Accelerate the prototyping of simulation systems by promoting component reusability
- Promote collaboration between research groups
- Facilitate technology transfer between research and industry

Our activities around the SOFA framework will be twofold. We will remain one of the leading teams contributing to the design of SOFA, the development of its architecture and its distribution to research groups and industrial partners. In addition, we will use SOFA as a core element of most of our simulations, as a mean to facilitate the integration of results from partners of the national initiative, and to simplify the development of prototypes of simulation systems. For the past few years, there have been a few attempts at designing software toolkits for medical simulation. Examples include [41], GiPSi [30], SPORE [40] or SSTML [27]. These different solutions aim at the same goal: providing an answer (usually Open Source) to the various challenges of medical simulation research and development. Although our aim is similar, we propose a different approach, through a very modular and flexible software framework, while minimizing the impact of this flexibility on the computation overhead. To achieve these objectives, we have developed a new architecture that implements a series of innovative concepts. Also, by developing the SOFA framework collaboratively with scientific experts in the different areas of medical simulation, we believe we can provide state-of-the-art solutions that are generically applicable, yet computationally efficient. The following sections describe in more details our approach to the development of this framework, from a technical standpoint and from the perspective of a collaborative work.

5.1.1. SOFA architecture

Medical simulation relies on a variety of interacting physics-based models, such as rigid structures (e.g. bones), deformable structures (e.g. soft-tissues) and fluids. It also involves anatomical representations through geometrical models, used for visual rendering, collision detection or meshes that will support various computational models. Finally, interactions between these different models need to be efficient, accurate and capable of handling a variety of representations. In some instances, a hierarchy also exists between the various anatomical structures, and needs to be taken into account in the description of the simulated environment. The design of the SOFA architecture, by supporting these various requirements, brings the flexibility needed for academic research. Yet, its very efficient implementation makes it also suitable for professional applications and potentially for product development. This architecture relies on several innovative concepts, in particular the notion of multi-model representation. In SOFA, most simulation components (deformable models, collision models, medical devices, etc.) can have several representations, connected through a mechanism called mapping. Each
representation is optimized for a particular task (e.g. collision detection, visualization) while at the same time improving interoperability by creating a clear separation between the functional aspects of the simulation components. As a consequence, it is possible to have models of very different nature interact together, for instance rigid bodies, deformable objects, and fluids. This is an essential aspect of SOFA, as it will help the integration of new research components. This modular design also facilitates the rapid prototyping of simulation systems, allowing various combinations of algorithms to be tested and compared against each other. At a finer level of granularity, we also propose a decomposition of physical models (i.e. any model that behaves according to the laws of physics) into a set of basic components. In the case of (bio)mechanical models, which are computationally expensive, many strategies have been used to improve computation times or to reduce the complexity of the original model: linear elastic models have often been used instead of more complex non-linear representations, mass-spring methods as an alternative to finite element methods, etc. Each of these simplifications induces drawbacks, yet the importance of these drawbacks depends largely on the context in which they are applied. It becomes then very difficult to choose which particular method is most likely to provide the best results for a given simulation. To address this issue in SOFA we have introduced a finer level of granularity which permits to independently test and compare each component, such as time integration schemes, to see the change in performance or robustness of the simulation, or to test different constitutive models. These changes can be made in a matter of seconds, without having to recompile any of the code, by simply editing an XML file.

5.1.2. Current Results

Version 1.0 RC1 of SOFA was released in December 2011. More than 87,000 downloads of SOFA have been counted as of December 2011. More than 70 researchers, students, engineers have contributed at various degrees to SOFA, for a total of about 700,000 lines of code. Currently, thanks to its advanced architecture, SOFA allows to:

- Create complex and evolving simulations by combining new algorithms with existing algorithms
- Modify most parameters of the simulation by simply editing a XML file
- Build complex models from simpler ones using a scene-graph description
- Efficiently simulate the dynamics of interacting objects using abstract equation solvers
- Reuse and easily compare a variety of available methods
- Transparently parallelize complex computations using semantics based on data dependencies
- Use new generations of GPUs through the CUDA API to greatly improve computation times
Various results and information can be obtained on the SOFA website at http://www.sofa-framework.org. Most of the current results are generic and only aim at validating the different aspects of the SOFA framework. Developments of complex medical simulations have recently started, in particular in the areas of ophthalmic surgery and interventional radiology. We have also started a collaboration with a few companies (Digital Trainers, Didhaptics, B.K.) which are in the process of developing medical applications based on SOFA.

Figure 4. Animation of a chain combining a FEM model, a mass-spring model, a FFD grid, and a rigid body. This example is a perfect illustration of the flexibility of SOFA. Not only several algorithms for rigid or deformable bodies can be part of the same simulation, but they can also interact in a physically correct manner. No constraints between links were pre-defined, instead we relied on collision detection and stiff contact forces to handle the contacts. Using implicit integrator handling dynamically-created groups of interacting objects resulted in a stable simulation.
5. Software

5.1. H2OLab

Participants: Jocelyne Erhel [correspondant], Aurélien Le Gentil, Géraldine Pichot, Baptiste Poirriez, Nadir Soualem.

The software platform H2OLab is devoted to stochastic simulations of groundwater flow and contaminant transport in highly heterogeneous porous and fractured geological media. It contains software and a database which are interfaced through the web portal H2OWEB. The platform H2OLab is an essential tool for the dissemination of scientific results. Currently, software and database are shared by the partners of the Micas project (see 8.1.2 ). Software integrated in the platform and registered at APP are GW-UTIL, GW-NUM, PARADIS, MP-FRAC.

See also the web page http://h2olab.inria.fr.

5.2. GW-UTIL

Participants: Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant], Baptiste Poirriez, Nadir Soualem.

The software GW-UTIL allows to discretize PDE for flow and transport in aquifers and to deal with stochastic models. It contains a set of utilitary modules for geometry, input, output, random numbers, visualization, parallel computing, numerical algorithms, etc. A package is devoted to launch applications.

See also the web page http://h2olab.inria.fr.

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

5.3. GW-NUM

Participants: Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant], Baptiste Poirriez, Nadir Soualem.

The software GW-NUM is a set of generic modules to discretize PDE of flow and transport in 2D computational domains in order to deal with stochastic models. Methods for flow simulations are either Finite Volume on structured meshes or Mixed Finite Element with unstructured meshes. Method for transport simulations is a particle tracker for advection and a random walker for diffusion. Uncertainty Quantification method is Monte-Carlo. For flow computations, the involved linear system is solved by external software devoted to sparse matrices.

See also the web page http://h2olab.inria.fr.

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

5.4. MP-FRAC

Participants: Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant], Baptiste Poirriez, Nadir Soualem.
The software MP-FRAC aims at modelling and simulating numerically flow in a fractured aquifer. The physical domain is a network of fractures, either deterministic or stochastic, with a permeability field either deterministic or stochastic. The software computes the velocity field in the aquifer, by assuming that the medium is saturated and that flow is steady-state. Physical equations are stochastic PDEs, handled by a Monte-Carlo method. This non intrusive approach generates a set of random samples, which are used for simulations. Then, the software analyzes statistically the flow in the stochastic case. The objective is to characterize hydraulic properties in Discrete Fracture Networks. The software MP-FRAC handles a simulation corresponding to one sample, whereas Monte-Carlo method is implemented in a generic way by the software GW-NUM. The software is specific of the physical model (Discrete Fracture Network) and of the application (steady-state flow). Generic numerical methods to discretize PDE are implemented in the software GW-NUM. See also the web page http://h2olab.inria.fr.

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

5.5. PARADIS

Participants: Jocelyne Erhel, Aurélien Le Gentil, Géraldine Pichot [correspondant], Baptiste Poirriez, Nadir Soualem.

The software PARADIS aims at modelling and simulating numerically flow in a porous aquifer and transport by convection-diffusion of an inert solute. The porous medium is heterogeneous, with a stochastic or deterministic permeability field. A first step computes the velocity field in the aquifer, by assuming that the medium is saturated and that flow is steady-state. A second step computes the distribution of solute concentration, by assuming a transport by convection and by molecular diffusion. Physical equations are stochastic PDEs, handled by a Monte-Carlo method and discretized by numerical methods. This non intrusive approach generates a set of random samples, which are used for simulations. Then, the software analyzes statistically the flow in the stochastic case. The objectives are to determine asymptotic laws of transport, to characterize pre-asymptotic behavior and to define global laws.

The software PARADIS handles a simulation corresponding to one sample, whereas Monte-Carlo method is implemented in a generic way by the software GW-NUM. The software is specific of the physical model (heterogeneous porous medium) and of the application (steady-state flow then transport with macro-dispersion). Generic numerical methods to discretize PDE are implemented in the software GW-NUM. See also the web page http://h2olab.inria.fr/.

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

5.6. GRT3D

Participants: Édouard Canot, Jocelyne Erhel [correspondant], Souhila Sabit, Nadir Soualem.

Reactive transport modeling has become an essential tool for understanding complex environmental problems. It is an important issue for MoMaS partners (see section 8.1.1), in particular Andra (see section 7.1). We have developed a method coupling transport and chemistry, based on a method of lines such that spatial discretization leads to a semi-discrete system of algebraic differential equations (DAE system). The main advantage is to use a complex DAE solver, which controls simultaneously the timestep and the convergence of Newton algorithm. Another approach, called SIA, is to use a fixed-point method to solve the nonlinear system at each timestep.
The software suite GRT3D has three executable modules:

- SIA1D: Sequential Iterative Approach for 1D domains;
- GDAE1D: Global DAE approach for 1D domains;
- GDAE3D: Global DAE approach for 1D, 2D or 3D domains.

- Version: version 1.0, April 2011
- APP: registered
- Programming language: C

5.7. GPREMS

Participants: Édouard Canot, Jocelyne Erhel [correspondant], Désiré Nuentsa Wakam, Nadir Soualem.

GPREMS implements a robust hybrid solver for large sparse linear systems that combines a Krylov subspace method as accelerator with a Schwarz-based preconditioner. This preconditioner uses an explicit formulation associated to one iteration of the multiplicative Schwarz method. The Newton-basis GMRES, which aims at expressing a good data parallelism between subdomains is used as accelerator.

See also the web page http://www.irisa.fr/sage/.

- Version: version 1.0, May 2008
- APP: registered
- Programming language: C++

5.8. DGMRES

Participants: Jocelyne Erhel [correspondant], Désiré Nuentsa Wakam.

DGMRES implements a preconditioner based on adaptive deflation, which can be used with any preconditioner for the GMRES algorithm. It is distributed with the free software PETSC.

See also the web page http://www.irisa.fr/sage/.

- Version: version 1.0, June 2011
- APP: distributed with PETSC
- Programming language: C

5.9. AGMRES

Participants: Jocelyne Erhel [correspondant], Désiré Nuentsa Wakam.

AGMRES implements an augmented subspace approach, based on adaptive deflation, which can be used with any preconditioner for the GMRES algorithm. It also implements a Newton basis for enhancing parallelism. It will be distributed with the free software PETSC.

See also the web page http://www.irisa.fr/sage/.

- Version: version 1.0, November 2011
- APP: soon distributed with PETSC
- Programming language: C

5.10. PPAT: pseudo-spectrum

Participants: Édouard Canot [corresponding author], Bernard Philippe.
PPAT (Parallel PATH following software) is a parallel code, developed by D. Mezher, W. Najem (University of Saint-Joseph, Beirut, Lebanon) and B. Philippe. This tool can follow the contours of a functional from \( \mathbb{C} \) to \( \mathbb{R}^+ \). The present version is adapted for determining the level curves of the function 
\[
f(z) = \sigma_{\text{min}}(A - zI)
\]
which gives the pseudospectrum of matrix \( A \).

The algorithm is reliable: it does not assume that the curve has a derivative everywhere. The process is proved to terminate even when taking into account roundoff errors. The structure of the code spawns many independent tasks which provide a good efficiency in the parallel runs.

The software can be downloaded under the GPL licence from: http://sourceforge.net/projects/ppat.

5.11. MUESLI: Scientific computing

**Participant:** Édouard Canot [corresponding author].

Doing linear algebra with sparse and dense matrices is somehow difficult in scientific computing. Specific libraries do exist to deal with this area (e.g. BLAS and LAPACK for dense matrices, SPARSKIT for sparse ones) but their use is often awful and tedious, mainly because of the large number of arguments which must be used. Moreover, classical libraries do not provide dynamic allocation. Lastly, the two types of storage (sparse and dense) are so different that the user must know in advance the storage used in order to declare correctly the corresponding numerical arrays.

MUESLI is designed to help in dealing with such structures and it provides the convenience of coding in Fortran with a matrix-oriented syntax; its aim is therefore to speed-up development process and to enhance portability. It is a Fortran 95 library split in two modules: (i) FML (Fortran Muesli Library) contains all necessary material to numerically work with a dynamic array (dynamic in size, type and structure), called \( \text{mfArray} \); (ii) FGL (Fortran Graphics Library) contains graphical routines (some are interactive) which use the \( \text{mfArray} \) objects.

MUESLI includes some parts of the following numerical libraries: Arpack, Slatec, SuiteSparse, Triangle, BLAS and LAPACK.

Linux is the platform which has been used for developing and testing MUESLI. Whereas the FML part (numerical computations) should work on any platform (e.g. Win32, Mac OS X, Unix), the FGL part is intended to be used only with X11 (i.e. under all UNIXes).

Last version of MUESLI is 2.5.2 (29 nov 2011). More information can be found at: http://www.irisa.fr/sage/edouard/canot/muesli/

5.12. CANARD: BEM for surface flows

**Participant:** Édouard Canot [corresponding author].

When dealing with non-linear free-surface flows, mixed Eulerian-Lagrangian methods have numerous advantages, because we can follow marker particles distributed on the free-surface and then compute with accuracy the surface position without the need of interpolation over a grid. Besides, if the liquid velocity is large enough, Navier-Stokes equations can be reduced to a Laplace equation, which is numerically solved by a Boundary Element Method (BEM); this latter method is very fast and efficient because computing occur only on the fluid boundary. This method is applied to the spreading of a liquid drop impacting on a solid wall and to the droplet formation at a nozzle; applications take place, among others, in ink-jet printing processes.

The code used (CANARD) has been developed with Jean-Luc Achard (LEGI, Grenoble) for fifteen years and is used today mainly through collaborations with Carmen Georgescu at UPB (University Polytechnica of Bucarest, Romania), and with Alain Glière (CEA-LETI, Grenoble).
5. Software

5.1. nD-SAFIR: Image denoising software

Participants: Charles Kervrann, Patrick Bouthemy.

The nD-SAFIR software (APP deposit number: IDDN.FR.001.190033.002.S.A.2007.000.21000 / new release 3.0 in 2012) written in C++, JAVA and MATLAB, removes additive Gaussian and non-Gaussian noise in still 2D or 3D images or in a 2D or 3D image sequences (with no motion computation) (see Figure 1 ) [ 4 ]. The method is unsupervised and is based on a pointwise selection of small image patches of fixed size (a data-driven adapted way) in spatial or space-time neighbourhood of each pixel (or voxel). The main idea is to associate with each pixel (or voxel) the weighted sum of intensities within an adaptive 2D or 3D (or 2D or 3D + time) neighbourhood and to use image patches to take into account complex spatial interactions. The neighbourhood size is selected at each spatial or space-time position according to a bias-variance criterion. The algorithm requires no tuning of control parameters (already calibrated with statistical arguments) and no library of image patches. The method has been applied to real noisy images (old photographs, JPEG-coded images, videos, ...) and is exploited in different biomedical application domains (time-lapse fluorescence microscopy, video-microscopy, MRI imagery, X-ray imagery, ultrasound imagery, ...).

Academic licence agreements: Institut Curie, CNRS, ENS Ulm, Oxford university, Weizmann Institute, UCSF San-Francisco, Harvard university, Berkeley university, Stanford university, Princeton university, Georgia-Tech, Kyoto university, IMCB Singapore ...

Partners: J. Boulanger, J. Salamero (UMR 144 CNRS Institut Curie), P. Elbau (RICAM Linz, Austria), J.B. Sibarita (UMR 5091 University of Bordeaux 2)

5.2. Fast2D-SAFIR: Fast denoising of large 2D images

Participant: Charles Kervrann.

The Fast2D-SAFIR software (APP deposit number: IDDN.FR.001.190033.001.S.A.2007.000.21000) written in C++ removes mixed Gaussian-Poisson noise in large 2D images, typically $10^3 \times 10^3$ pixels, in few seconds. The method is unsupervised and is a simplified version of the method related to the SAFIR-nD software. The software dedicated to microarrays image denoising, was licensed to the INNOPSYS company which develops scanners for disease diagnosis and multiple applications (gene expression, genotyping, aCGH, ChiP-chip, microRNA, ...).

5.3. PBED: Patch-based event detection

Participant: Charles Kervrann.

The PBED software written in C++ automatically quantifies in space and time the number of sudden and transient events observed in fluorescence (WF, TIRF) microscopy. The algorithm parameters are calibrated from the comparison of image patches expected to distinguish sudden appearing/vanishing fluorescent spots/particles from other motion behaviors such as lateral movements [ 1 ] and [ 23 ]. Two statistical procedures are proposed respectively to control the number of false alarms (Benjamini-Hochsberg, Bonferonni). The algorithm is mainly used to statistically explore the effect of several biological perturbations on the rate of transient events detected on the pilot biological model (e.g. Langerin-YFP endocytic-recycling trans-membrane protein).

Partners: J. Boulanger, A. Gidon, A. Chessel, B. Cinquin, J. Salamero (UMR 144 CNRS Institut Curie)
5.4. HullkGround: Background subtraction by convex hull estimation

**Participant:** Charles Kervrann.

The HULLKGROUND software (APP deposit number: IDDN.FR.001.400005.000.S.P.2009.000.21000) written in JAVA (plug-in IMAGEJ (http://rsbweb.nih.gov/ij/)) decomposes a fluorescence microscopy image sequence into two dynamic components: i) an image sequence showing mobile objects; ii) an image sequence showing the slightly moving background. Each temporal signal of the sequence is processed individually and analyzed with computational geometry tools. The convex hull is estimated automatically for each pixel and subtracted to the original signal. The method is unsupervised, requires no parameter tuning and is a simplified version of the α shapes-based scale-space method [24].

**Partners:** A. Chessel and J. Salamero (UMR 144 CNRS Institut Curie)

5.5. TubuleJ: Straightening of microtubule cryo-EM projection views

**Participant:** Charles Kervrann.

The TUBULEJ software (APP deposit number: IDDN.FR.001.240023.000.S.P.2011.000.21000) written in JAVA (plug-in IMAGEJ (http://rsbweb.nih.gov/ij/)) is devoted to the analysis of microtubules and helical structures in 2D cryo-electron microscope images. The software straightens curved microtubule images by estimating automatically points locations on the microtubule axis. The estimation of microtubule principal axis relies on microtubule cylindrical shape analyzed in the Fourier domain. A user-friendly interface enables to filter straight fiber images by selecting manually the layer lines of interest in the Fourier domain. This software can be used to generate a set of 2D projection views from a single microtubule projection view and a few parameters of this microtubule structure. These projection views are then back projected, by using the IMOD plug-in (http://rsbweb.nih.gov/ij/), to reconstruct 3D microtubules.

**Partners:** S. Blestel and D. Chrétien (UMR 6026 CNRS University of Rennes 1)
5.6. Cryo-Seg: Segmentation of tomograms in cryo-electron microscopy

Participant: Charles Kervrann.

The CRYO-SEG software written in C++ has been developed to detect microtubule structures and helical structures in 2D cryo-electron microscope images (see Figure 2). Cryo-electron tomography allows 3D observation of biological specimens in their hydrated state. Segmentation is formulated as Maximum A Posteriori estimation problem and exploits image patches to take into account spatial contexts (Markov Random Fields). Because of the contrast anisotropy in the specimen thickness direction, the whole tomogram is segmented section by section, with an automatic update of reference patches. This algorithm has been evaluated on synthetic data and on cryo-electron tomograms of in vitro microtubules. On real data, this segmentation method extracts the most contrasted regions of microtubules, and 3D visualization is improved.

Partners: S. Blestel and D. Chrétien (UMR 6026 CNRS University of Rennes 1)

Figure 2. CRYO-SEG software: Segmentation of 3D microtubules in a cryo-EM tomogram (left) and 2D view (right) (UMR 6026 CNRS University of Rennes 1)
4. Software

4.1. The Matlab System Identification ToolBox (SITB)

**Participant:** Qinghua Zhang.

*This development is made in collaboration with Lennart Ljung (Linköping University, Sweden), Anatoli Juditsky (Joseph Fourier University, France) and Peter Lindskog (NIRA Dynamics, Sweden).*

The System Identification ToolBox (SITB) is one of the main Matlab toolboxes commercialized by The Mathworks. INRIA participates in the development of its extension to the identification of nonlinear systems which is released since 2007. It includes algorithms for both black box and grey box identification of nonlinear dynamic systems. INRIA is mainly responsible for the development of black box identification, with nonlinear autoregressive (NLARX) models and block-oriented (Hammerstein-Wiener) models.

4.2. Inverse Scattering for Transmission Lines (ISTL)

**Participants:** Michel Sorine, Qinghua Zhang.

ISTL is a software for numerical computation of the inverse scattering transform for electrical transmission lines. In addition to the inverse scattering transform, it includes a numerical simulator generating the reflection coefficients of user-specified transmission lines. With the aid of a graphical interface, the user can interactively define the distributed characteristics of a transmission line. This software is mainly for the purpose of demonstrating a numerical solution to the inverse problem of non uniform transmission lines. Its current version is limited to the case of lossless transmission lines. It is registered at Agence pour la Protection des Programmes (APP) under the number IDDN.FR.001.120003.000.S.P.2010.000.30705.

4.3. CGAO: Contrôle Glycémique Assisté par Ordinateur

**Participants:** Alexandre Guerrini, Michel Sorine.

This development is made in collaboration with Pierre Kalfon (Chartres Hospital) and Gaëtan Roudillon (LK2).

This software developed with LK2 and Hospital Louis Pasteur (Chartres) provides efficient monitoring and control tools that will help physicians and nursing staff to avoid hyperglycaemia and hypoglycaemia episodes in Intensive Care Units. It is used in a large clinical study, CGAO-REA. Commercialization will be done by LK2.

The software is designed to assist physicians to deal with a variant of the classical Stability/Precision dilemma of control theory met during blood-glucose control. It has been tested in the ICU of Chartres and, since November 2009, it is used in a large scale study launched by the SFAR (French Society of Anesthesia and Intensive Care) involving 62 ICUs and including 6422 patients.

More than 3500 patients have been included in CGAO-REA.

4.4. LARY_CR: Software package for the Analysis of Cardio Vascular and Respiratory Rhythms

**Participants:** Claire Médigue, Serge Steer.
LARY_CR is a software package dedicated to the study of cardiovascular and respiratory rhythms [77]. It presents signal processing methods, from events detection on raw signals to the variability analysis of the resulting time series. The events detection concerns the heart beat recognition on the electrocardiogram, defining the RR time series, the maxima and minima on the arterial blood pressure defining the systolic and diastolic time series. These detections are followed by the resampling of the time series then their analyse. This analyse uses temporal and time frequency methods: Fourier Transform, spectral gain between the cardiac and blood pressure series, Smooth Pseudo Wigner-Ville Distribution, Complex DeModulation, temporal method of the cardiovascular Sequences. The objective of this software is to provide some tools for studying the autonomic nervous system, acting in particular in the baroreflex loop; its functioning is reflected by the cardiovascular variabilities and their relationships with the other physiological signals, especially the respiratory activity. Today LARY_CR is used only internally, in the framework of our clinical collaborations.
4. Software

4.1. TEOS: Tranus Exploration and Optimization Software

Participants: Anthony Tschirhard, Mathieu Vadon, Elise Arnaud, Emmanuel Prados.

The TEOS software offers a set of tools to help the calibration of the land use and transport integrated model TRANUS. It uses some exploration and optimization procedures of the relevant parameters.
5. Software

5.1. Main softwares

Participants: Olivier Collin [correspondant], Dominique Lavenier, François Coste, Olivier Sallou, Romaric Sabas, Guillaume Rizk, Andres Burgos.

We highlight here 3 softwares of the team which received considerable care this year, in particular to improve their ergonomy and diffusion. In the following sections, all softwares of the team will be described, classified according to their applicative domain.

5.1.1. Biomaj : Data synchronization and processing workflow

BioMAJ (BIOlogie Mise A Jour) is a workflow engine dedicated to data synchronization and processing. The Software automates the update cycle and the supervision of the locally mirrored databank repository. Thanks to the funding of INRIA’s ADT, the BioMAJ software has been ergonomically improved and is diffusion enhanced. It is now part of a Linux distribution (Debian-med). The tool is now used on many bioinformatics core facilities in France and Europe. It is used as an infrastructure tool but also as a key component of new resources. For example the AnnotQTL tool relies heavily on BioMAJ. Another example is popgenie, an integrative explorer of the Populus genome in Sweden has been built on top of BioMAJ.

[Web site: http://biomaj.genouest.org]

5.1.2. GASSST: Short reader mapper for large genomic dataset

GASSST is a short read mapper allowing very large genomic dataset to be processed. It takes as input raw data (reads) coming from next generation sequencing machines and map them over full genomes. In 2011, the GASSST software has been tuned to meet industrial requirements and transferred to the GenomeQuest Company. A specific license agreement has been set up between INRIA and GenomeQuest for integrating GASSST into the GenomeQuest NGS tool suite.

web site: http://www.irisa.fr/symbiose/projects/gassst/

5.1.3. Protomata learner: fine characterization of protein families

Protomata-Learner V2.0 is a tool to infer weighted automata for the characterization of (structural or functional) families of proteins from a sample of (unaligned) sequences belonging to the family. Protomata-Learner has been completely rewritten thanks to the ADT "Suite logicielle pour la modélisation de familles protéiques par automates": based on a better formalisation and thanks to the implementation of efficient weighting techniques, this new version is significantly faster and gives better results. Special care has been given to the integration of the different programs to propose an easy-to-use suite.

Protomata-Learner has been tested and improved on real use-case thanks to collaborations established in Lepidolf and Pelican ANR projects. New scanning algorithms (Forward scores) and procedures for choosing automatically the best set of parameters have been developed. New signatures for the studied families of proteins have been established and are used for the predictions of candidates by our partners.

[Web site: http://tools.genouest.org/tools/protomata/]
5.2. Bioinformatics community tools

**Participants:** Olivier Collin [contact], Olivier Sallou, Charles Deltel, François Moreews, Anthony Bre-tegaudeau, Delphine Naquin, Aurélien Roult, Romaric Sabas, Claudia Hériveau.

- **BioMAJ** See first section above.
- **GRISBI** The GRISBI project is aiming to set up a grid infrastructure devoted to the Bioinformatics community. This infrastructure is built upon the resources available on different bioinformatics facilities through gLite middleware. [Web site: http://www.grisbio.fr]
- **Mobyle.net** In partnership with other bioinformatics platforms, GenOuest is setting up a distributed network of bioinformatics resources built upon web portals based on the Mobyle platform. [Web site: http://mobyle.net.rpbs.univ-paris-diderot.fr:8080/]
- **MetaData platform** Seqcrawler is an indexing platform for biological meta data and sequences, providing a google like web interface. It can scale from single computers to the cloud. [Web site: http://seqcrawler.sourceforge.net/]
- **DrMotifs** DrMotifs is a new software resource aiming at the integration of different software commonly used in pattern search and discovery. This resource will also integrate new software elaborated by the Symbiose team. [Web site: http://www.drmotifs.org] [Blog site: http://drmotifs.genouest.org]

5.3. Parallel softwares

**Participants:** Dominique Lavenier [contact], Charles Deltel, Erwan Drezen, Guillaume Chapuis, Guillaume Rizk.

- **PLAST:** **intensive bank sequence comparison.** PLAST is a parallel version of BLAST-like software targeting multiple parallel hardware such as FPGA accelerator or GPU boards. [web site: http://www.irisa.fr/symbiose/projects/plast/]
- **SLICEE** (Service Layer for Intensive Computation Execution Environment) is part of the BioWIC project. This software proposes (1) to abstract the calls to the cluster scheduler by handling command submission; (2) to take care of exploiting the data parallelism with data specific methods; (3) to manage data using a cache references mechanism and route data between tasks. [Web site: http://vapor.gforge.inria.fr/]
- **QTL-map** is a GPU parallel version of the QTLMap Software developed in cooperation with INRA [web site: http://www.inra.fr/qtlmap]

5.4. Softwares for Next Generation Sequencing data

**Participants:** Dominique Lavenier [contact], Pierre Peterlongo, Guillaume Rizk, Rayan Chikhi.

- **GASSST:** **short reads mapper.** See first section above.
- **kisSnp and kisSplice : variant identification without the use of a reference genome.** kisSnp is a tool to find single nucleotide polymorphisms (SNP) by comparing two sets of raw NGS reads. [web site: http://alcovna.genouest.org/kissnp/]
  - KisSplice finds alternative splicings but also short insertions, deletions and duplications, SNPs and sequencing errors in one or two RNA-seq sets, without assembly nor mapping on a reference genome. [web site: http://alcovna.genouest.org/kissplice/]
- **Blastree:** is a tool for computing intensive approximate pattern matching in a string graph. [web site: http://alcovna.genouest.org/blastree/]
- **Mapsembler:** **targeted assembly software.** Mapsembler takes as input a set of NGS raw reads and a set of input sequences (starters). It first determines if each starter is read-coherent, e.g. whether the reads confirm the presence of each starter in the original sequence. Then for each read-coherent starter, Mapsembler outputs its sequence neighborhood as a linear sequence or as a graph, depending on the user choice. [web site: http://alcovna.genouest.org/mapsembler/]
5.5. Genome structure

Participants: Jacques Nicolas [contact], Catherine Belleannée, Pierre Peterlongo, Raoul Vorc’h, Anthony Bretaudneau, Olivier Sallou.

- **CRISPI: CRISPR identification.** CRISPI is a user-friendly web interface with many graphical tools and facilities allowing extracting CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats), finding out CRISPR in personal sequences or calculating sequence similarity with spacers. [web site: http://crispi.genouest.org]

- **Logol** is a language and a tool to define biological patterns to look for in one or more sequences (dna/rna/proteins). Patterns can be complex: the tool allows the use of variables to look for repetitions for example, the use of gaps and morphisms (reverse word complement for example), etc. [web site: http://www.genouest.org/spip.php?article758]

5.6. Protein sequence and structure

Participants: Rumen Andonov [contact], François Coste, Andres Burgos, Pavel Senin.

- **A_purva: Scoring similarities between proteins.** A_purva is a Contact Map Overlap maximization (CMO) solver. Given two protein structures represented by two contact maps, A_purva computes the amino-acid alignment which maximizes the number of common contacts. [web site: http://apurva.genouest.org]

- **Protomata learner: fine characterization of protein families** See first section above.

5.7. Systems biology

Participants: Anne Siegel [contact], Michel Le Borgne, Françoise Mereews, Anthony Bretaudneau.

- **Bioquali: confront knowledge-based regulatory models with data.** Bioquali tests the consistency between an interaction graph and transcriptomic data. It outputs nodes in the network whose variation cannot be globally explained by the other available observations. [web site: http://bioquali.genouest.org  Cytoscape java web start]
4. Software

4.1. V-Plants

Participants: Frédéric Boudon, Christophe Godin [coordinator], Yann Guédon, Christophe Pradal [software architect], Daniel Barbeau, Thomas Cokelaer, David Da Silva, Jean-Baptiste Durand, Pascal Ferraro, Eric Moscardi.

Computer algorithms and tools developed by the Virtual Plants team are integrated in a common software suite V-Plants, dedicated to the modeling and analysis of plant development at different scales (e.g. cellular tissue, whole plant, stand). The VPlants packages are integrated in OpenAlea as Python components. Several components are distributed and usable through the visual programming environment (see figure 2):

- Multi-scale geometric modeling and visualization. VPlants.PlantGL is a geometric library which provides a set of graphical tools and algorithms for 3D plant modeling at different scales [8]. It is used by many other components to represent the geometry of biological shapes from 3D meristems, plant architectures to plant populations. VPlants.PlantGL is built around a scene-graph data structure and provides efficient algorithms and original geometrical shapes (parametric surfaces, dedicated envelopes), that are useful for plant modeling.

- Statistical sequence and tree analysis. Different statistical packages (i.e. VPlants.StatTool, VPlants.SequenceAnalysis, VPlants.TreeMatching and VPlants.TreeAnalysis) are now available in OpenAlea. They provide different models and algorithms for plant architecture analysis and simulation.

- Meristem functioning and development. A first set of components has been created in the last 4-years period to model meristem development in OpenAlea. These tools are currently being integrated thoroughly in the platform so that modelers and biologists can use them, and reuse components easily (for meristem 3D reconstruction, cell tracking, statistical analysis of tissues, creating and manipulating atlases, creating or loading models of growth that can further be run on digitized structures, etc).

- Standard data structure for plants. A new implementation of the MTG formalism for representing and manipulating multiscale plant architecture has been developed. It provides a central data-structure to represent plants in a generic way in OpenAlea. This implementation is available through the packages OpenAlea.MTG. These components make it possible to share plant representations between users and fosters the interoperability of new models.

- Simulation system. The study of plant development requires increasingly powerful modeling tools to help understand and simulate the growth and functioning of plants. In the last decade, the formalism of L-systems has emerged as a major paradigm for modeling plant development. Previous implementations of this formalism were made based on static languages, i.e. languages that require explicit definition of variable types before using them. These languages are often efficient but involve quite a lot of syntactic overhead, thus restricting the flexibility of use for modelers. We developed L-Py [26] an adaptation of L-systems to the Python language (basis of OpenAlea). Thanks to its dynamic typing property, syntax is simple, code execution is made easy and introspection property of the language makes it possible to parameterize and manipulate simply complex models. Independent L-systems can be composed to build-up more complex modular models. MTG structures (that are a common way to represent plants at several scales) can be translated back and forth into L-system data-structure and thus make it easy to reuse in L-systems tools for the analysis of plant architecture based on MTGs. Extensions to integrate multiscale dynamic models are currently being developed in collaboration with P. Prusinkiewicz and his team from the University of Calgary. A paper presenting L-Py has been submitted to Frontiers in Technical Advances in Plant Sciences.
4.2. OpenAlea

**Participants:** Frédéric Boudon, Christophe Godin, Yann Guédon, Christophe Pradal [coordinator], Daniel Barbeau, Thomas Cokelaer, Christian Fournier, Eric Moscardi.

*This research theme is supported by an INRIA ADT Grant and by a RTRA Grant.*

---

Figure 1. V-Plants components of the OpenAlea platform: simulating plant processes at different scales. Top Left: Reconstruction of a virtual meristem, analysis and simulation of the auxin fluxes inside the meristem. Top Right: Reconstruction of a virtual apple tree from digitized data. Bottom: Simulation of an ecosystem (A beech “Fagus Sylvatica L.” trees forest) with a multi level approaches. On the left, explicit representation of the crown volumes that serves as input to generate the detailed representation, on the right.

Figure 2. OpenAlea.Visualea: Visual programming interface. The package manager shows the available components. The components can be interconnected on a workspace to form a data-flow. The python interpreter allows low level interaction with the system.
**OpenAlea** [9] is an open source and collaborative software project primarily dedicated to the plant research community. It is designed as a component framework to dynamically glue together models from different plant research labs, and to enhance re-usability of existing models in the plant research community.

The architecture of OpenAlea is based on a component architecture. It provides a set of standard components (OpenAlea.Stdlib), a package manager to dynamically add and retrieve new components, and a port graph data-structure to compose models by interconnecting components into a data-flow.

Visualea provides a visual programming environment, used by scientists to build new model interactively by connecting available components together through an easy-to-use graphical user interface.

In 2011, one major release was done: Openalea 1.0. The following progresses were accomplished:

1. Develop and extend OpenAlea and Visualea:
   - The standard library of components has been extended with useful scientific packages such as a flexible data plotting package (Openalea.Pylab), 2D and 3D image manipulation (Openalea.Image) and linear algebra operations (Openalea.Numpy).
   - Several models of computation have been implemented on the data-flow data-structure to enable discrete event simulation and control flow inside OpenAlea.

2. Animation and diffusion
   - The first OpenAlea Workshop have been held in Montpellier and has been attended by more than 60 scientists. A scientific board has been defined to manage the development and diffusion of OpenAlea. It is composed by 12 scientists.
   - StandAlone binary installers have been released on Windows and Mac to ease the installation of a large number of packages without relying on a web server. A Ubuntu repository has been set up on Launchpad.
   - A continuous integration server has been set up (link [http://vp-continuous.cirad.fr](http://vp-continuous.cirad.fr)) to test the reliability of all the components after every commit.
   - The OpenAlea project is hosted at the Inria gforge. The web site is visited by more than 300 unique visitor each month; 470000 web pages have been visited and the different available components of OpenAlea have been downloaded more than 500,000 times during the last two years. OpenAlea is the first project at Inria Gforge in term of number of downloads and of page views.

### 4.3. Alinea

**Participants:** Christian Fournier, Christophe Pradal, Frédéric Boudon, Christophe Godin.

Other participants: Bruno Andrieu, Michael Chelle, Gaëtan Louarn, Benoît de Solan, Mariem Abichou, Liqi Han, Elmer Ccopa-Rivera, Frederic Baret, Rafaële Casa, Youcef Mammeri, Didier Combes, Camille Chambon, Romain Barillot, Pierre Huynh, Jean-Christophe Soulie, Delphine Luquet.

The aim of this Action Ciblée Incitative of INRA is to constitute a consortium of modelers from INRA around the OpenAlea platform, and to integrate various ecophysiological models of simulation in OpenAlea (radiative transfer, interaction between plant and pest, circulation of hydric fluxes, and dispersion). The project includes 3 INRA teams and the INRIA Virtual Plants project.

Different components have been integrated into the OpenAlea platform:

- **Alinea.Adel** is a module to simulate the 3D architectural development of gramineous crops.
- **Alinea.Caribu** is a modeling suite for lighting 3D virtual scenes, especially designed for the illumination of virtual plant canopies such as virtual crop fields. It uses a special algorithm, the nested radiosity, that allows for a precise estimation of light absorption at the level of small canopy elements.
- Alinea.TopVine is a component to reconstruct grapevine canopy structure.
- Ecomeristem is a crop growth, eco-physiological model that was designed for rice (model plant for cereals) to account for plant morphogenesis and its plasticity depending on genetic potential and sensitivity to the environment (water, temperature, radiation).
- Alinea.Nema is a module used for modeling of nitrogen dynamics between leaves.
- MAAppleT is a FSPM model of an apple tree taking into account stochastic models for the topological development, a biomechanical model for branch bending, physiological laws as well as light interception.
- M2A3PC is a generic model to simulate spread of a pathogen on a growing plant like vine/powdery mildew and apple tree/apple scab.

In 2011, several research group from INRA and CIRAD have worked together on reconstruction and simulation of plant development for different species of gramineous such as rice, wheat, maize and other species like vine, rose or apple tree.
5. Software

5.1. Vistal

Participant: Alexandre Abadie.

VistaL is a software platform of 3D and 3D+T image analysis allowing the development of generic algorithms used in different contexts (rigid and non-rigid registration, segmentation, statistical modelling, calibration of free-hand 3D ultrasound system and so on, diffusion tensor image processing, tractography). This software platform is composed of generic C++ template classes (Image3D, Image4D, Lattice and so on) and a set of 3D/3D+T image processing libraries. VistaL is a multi-operating system environment (Windows, Linux/Unix...). A web site presenting the project has been developed, precompiled packages and the SDK are now available. VistaL APP registration number is: IDDN.FR.001.200014.S.P.2000.000.21000. See also the web page [http://vistal.gforge.inria.fr](http://vistal.gforge.inria.fr).

![VistaL results screenshots](image)

(a) (b) (c)

**Figure 1. Some ViSTAL results screenshots:**
- a) The ViSTAL Logo,
- b) ViSTAL Brain surface and sulci modelisation,
- c) The ROI3D Extraction view

- Keywords: medical image processing, image analysis, registration, segmentation, denoising
- Software benefit: New methodological image processing, some GPU based algorithms, easy to use C++ library
- APP: IDDN.FR.001.200014.S.P.2000.000.21000
- License: Licence Propriétaire
- Type of human computer interaction: C++ API and less complete Python API
- OS/Middleware: Windows, Mac et Linux.
- Required library or software: CMake (GPL) - ITK (BSD) - VTK (BSD) - Boost (BSD) - Libxml++ (LGPL) - CppUnit (LGPL)
- Programming language: C/C++, Python
- Documentation: Documentation Doxygen, documentation utilisateur.
5.2. Vistal-Tools
Participant: Alexandre Abadie.

The Vistal-Tools are a set of command line binaries based on the VisTaL library. These programs allow users to perform batch mode processing as well as scripting complex processing workflows. The most popular Vistal-Tools are NLMEANS (perform a NLMEANS filtering of 3D or 4D volumes), Registration (encapsulate the most common rigid registration algorithms), Tractography (track fibers from a DTI volume), etc.

5.3. Online applications
Participant: Alexandre Abadie.

Online applications offers a web service for testing the tools developped by the members of the VISAGES team: denoising based on Non Local Mean algorithm (3D and 2D) (NLMEAN), 3D rigid registration, brain symmetry plan estimation. This application support the main formats used in medical imaging data: Nifti-1, Analyze7.5, Mha, GIS. The applications are available at this url http://www.irisa.fr/visages/benchmarks. More than 2000 processes have been benchmarked to date using this service.

5.4. CLARCS: C++ Library for Automated Registration and Comparison of Surfaces
Participants: Alexandre Abadie, Sylvain Prima.

In collaboration with Benoit Combes, within the 3D-MORPHINE ARC project (http://3dmorphine.inria.fr), we conceived and implemented a C++ library (named CLARCS) for the automated analysis and comparison of surfaces. One of the primary goal of this library is to allow the assessment and quantification of morphological differences of free-form surfaces from medical or paleoanthropological data.

- APP: IDDN.FR.001.130002.000.S.P.2011.000.21000
- Programming language: C++

CLARCS was presented at the MeshMed MICCAI workshop (http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html) [27] and is to be distributed through a dedicated website (http://clarcs.inria.fr).

We also developed a surface viewer (named ‘Surface’).

- APP: IDDN.FR.001.110019.000.S.P.2011.000.21000
- Programming language: C++, Python

5.5. SUBANA: SUrface-BAsed Neuronavigation on Atlas for TMS
Participant: Sylvain Prima.

In collaboration with Charles Garraud (http://www.syneika.com), Benoit Combes and Pierre Hellier (http://serpico.rennes.inria.fr), we developed a software for i) the automated surface reconstruction of the face and skull cap from sparsely acquired points and ii) the automated nonlinear registration of free-form surfaces. The latter step is implemented using the CLARCS library (http://clarcs.inria.fr). The primary goal of this software is the surface-based neuronavigation for transcranial magnetic stimulation. The method was presented at the MeshMed MICCAI workshop (http://www2.imm.dtu.dk/projects/MeshMed/2011/index.html) [30].

- APP: IDDN.FR.001.440010.000.S.P.2010.000.31230
- Patent: was granted, but the reference number is unknown
- Programming language: C++

5.6. Shanoir
Participants: Guillaume Renard, Alexandre Abadie, Bernard Gibaud, Christian Barillot.
Shanoir (Sharing NeurOImaging Resources) is an open source neuroinformatics platform designed to share, archive, search and visualize neuroimaging data. It provides a user-friendly secure web access and offers an intuitive workflow to facilitate the collecting and retrieving of neuroimaging data from multiple sources and a wizard to make the completion of metadata easy. Shanoir comes along many features such as anonymization of data, support for multi-centres clinical studies on subjects or group of subjects.

Shanoir APP registration number is: IDDN.FR.001.520021.000.S.P.2008.000.31230

See also the web page: http://www.shanoir.org

- Keywords: neuroimaging, ontology, sharing neuroimage
- Software benefit: full featured neuroimaging management system with additional web services
- APP: IDDN.FR.001.200014.S.P.2000.000.21000
- License: Licence QPL
- Type of human computer interaction: Online web application, web service (SOAP messages based)
- OS/Middleware: Windows, Mac and Linux.
- Required library or software: Java 1.6, JBoss server, JBoss Seam, JSF, JPA Hibernate, EJB, Richfaces, Faceless, Ajax4JSF, Dcm4ctk, Dcm4chee.
- Programming language: Java
- Documentation: see the website

5.7. QtShanoir

Participants: Alexandre Abadie, Olivier Commowick, Guillaume Renard.

QtShanoir is a C++ Qt based library for querying data from a Shanoir server. For those who don’t know what is shanoir, see the shanoir website at http://shanoir.org. QtShanoir uses the soap based webservice provided by a shanoir server to get and display studies, patients, data with their associated metadata. In QtShanoir, you will find a set of Qt widgets (inherited from a QWidget object) that you can embed in your Qt application.

An APP registration is in progress and the library has been released in October under the LGPL license. See http://qtshanoir.gforge.inria.fr.

- Keywords: medical imaging, dicom
- Software benefit: offers a great solution to query a Shanoir server. Can be easily re-used in larger Qt applications
- License: no defined licence for the moment
- Type of human computer interaction: C++ library
- OS/Middleware: Linux, Windows and Mac
- Required library or software: Qt
- Programming language: C++
- Documentation: http://qtshanoir.gforge.inria.fr/html

5.8. QtDcm

Participant: Alexandre Abadie.

QtDcm is a C++ library implementing a widget that can be re-used with the Qt development framework. With this new widget, it is now easy to view the content of a Dicom CD-Rom, to manage dicom Query/Retrieve from a PACS and to convert downloaded data in the nifti format (easy to use medical image format). QtDcm APP registration number (2010) is: IDDN.FR.001.490036.000.S.P.2010.000.31230 A new APP registration is in progress and the library has been released in October under the LGPL license. See http://qtdcm.gforge.inria.fr.
See also the web page [https://www.irisa.fr/visages/members/aabadie/demos](https://www.irisa.fr/visages/members/aabadie/demos)

- **Keywords**: medical imaging, dicom
- **Software benefit**: offers a great solution to query medical images storage server (Dicom PACS). Can be easily reused in larger Qt applications
- **APP**: IDDN.FR.001.490036.000.S.P.2010.000.31230
- **License**: no defined licence for the moment
- **Type of human computer interaction**: C++ library
- **OS/Middleware**: Linux, Windows and Mac
- **Required library or software**: Qt, Dcmtk, dcm2nii (optional)
- **Programming language**: C++
- **Documentation**: [http://qtdcm.gforge.inria.fr/html](http://qtdcm.gforge.inria.fr/html)

### 5.9. AutoMRI

**Participant**: Camille Maumet.

autoMRI is an SPM-based set of tools to study structural and functional MRI data. This software is currently made up of three modules: autofMRI, autoVBM and autoROI. autofMRI produces statistical maps of activations and deactivations at the group or the subject level based on functional MRI data. It can deal with block or event-related designs and is highly configurable in order to fit to a wide range of needs. autoVBM performs between-group voxel-based morphometric analysis in order to outline regions of grey (or white) matter volume reduction and increase. To further study a morphometric or a functional analysis, regions of interest analysis can be performed with autoROI. This module also provides the user with laterality indexes.

- **Keywords**: fMRI, MRI, SPM, automation
- **Software benefit**: Automatic MRI data analysis based on SPM. Once the parameters are set, the analysis can be run without human interaction.
- **APP**: Coming soon
- **License**: Ceccil
- **Type of human computer interaction**: Matlab function (script, no GUI)
- **OS/Middleware**: Linux/Windows
- **Required library or software**: Matlab, SPM, SPM toolboxes : Marsbar, LI-toolbox, NS
- **Programming language**: Matlab
- **Documentation**: Available

### 5.10. Medinria

**Participants**: Alexandre Abadie, Clément Philipot, Olivier Commowick.

Medinria is a national INRIA project shared between 4 INRIA teams (Asclepios, Athena, Parietal and Visages). It aims at creating an easily extensible platform for the distribution of research algorithms developed at INRIA for medical image processing. This project has been funded by the D2T (ADT MedINRIA-NT) for two years, starting from late 2010. The Visages team participates in the development of the common core architecture and features of the software as well as in the development of specific plugins for the team’s algorithm. Medinria is currently being packaged for the main distribution platforms and will be released in the first two weeks of January 2012.
Keywords: medical imaging, diffusion imaging, registration, filtering, user-friendly interface
Software benefit: user-friendly interface to cutting-edge research tools for research clinicians. Straightforward to add functionalities through plugins.
License: core: BSD, plugins: choice of each team.
Type of human computer interaction: Qt-based GUI
OS/Middleware: Windows, Mac et Linux.
Required library or software : Qt, DTK, ITK, VTK.
Programming language: C++

5.11. EMPROS
Participant: Elise Bannier.

EMPROS stands for "Event Related Emotional Prosody Recognition fMRI Task". This software implements a paradigm, i.e., a sequence of stimuli to be proposed to a subject, in order to study the perception of emotions with functional MRI. The subject hears meaningless but emotionally charged pseudo-words or onomatopoeias and selects the evoked emotion among 5 emotions (joy, fear, sadness, anger, neutral) by pushing a button. The response of the subject is registered while a BOLD fMRI acquisition images his/her brain. This paradigm aims at detecting the cortical areas involved in emotional perception.

This software will be distributed as open source code.

APP: IDDN.FR. : APP registration in progress.
Patent: under application
Programming language: E-Basic
Programming software: E-Prime v2.0

5.12. IOGAT
Participant: Elise Bannier.

IOGAT stands for "Iowa Gambling Task for Event Related fMRI". This software implements a paradigm, i.e., a sequence of stimuli to be proposed to a subject, in order to study the decision making process with functional MRI. The subject is presented with 4 decks of cards. Each deck is associated with a gain or a loss of money in a non random way: 2 of them are advantageous to the subject whereas the other 2 are disadvantageous. The subject is asked to pick up cards, choosing freely the deck he/she picks up the card from, so as to maximize his/her gains. While the subject performs this task, his/her brain is imaged with a BOLD fMRI acquisition. This paradigm is designed to localize the cortical areas involved in the decision making process.

This software will be distributed as open source code.

APP: IDDN.FR. : APP registration in progress.
License: the software is being licensed to CHU Besancon
Patent: under application
Programming language: E-Basic
Programming software: E-Prime v2.0
ACES Project-Team (section vide)
5. Software

5.1. Introduction

We intend to develop a number of software to evaluate and validate our solutions. We will complete our development by experimentation, benchmarks and deployment in multi-paradigm platforms. We list our actual software that we intend to continue and to extend in the ADAM project-team.

5.2. CALICO

Participants: Laurence Duchien, Antonio de Almeida Souza Neto, Anne-Françoise Le Meur.

Modern software is characterized by a need for constant and rapid evolution, such as in the mobile domain. To facilitate the development and the rapid evolution of complex systems, software engineering approaches have been proposed, such as software architecture and agile software development. However, current solutions offer poor support to enable the development of a reliable system.

In this context we propose CALICO, an agile development framework for the design and evolution of safe component-based and service-oriented software. The agile software development relies on an iterative and incremental development cycle that allows the architect to iterate between the design of the architecture and the debug of the software in its execution context. At each iteration, the architect can evolve its software and check the consistency of its evolution through the execution of static and dynamic analysis tools. Thus, during the design and the evolution of the system, the architect can use a set of metamodels to specify the structure of the architecture and its various quality of services requirement. During the deployment, CALICO instantiates the system on the target runtime platform from the models specified and keeps them synchronized with the software during its execution. Through this means, the architect has a conceptual view which allows him to reason on the critical software properties during its evolution. Moreover, in order to check these evolutions, CALICO provides a unifying framework which allows reuse of many static analysis tools of software architectures and dynamic debugging tools, that were scattered in different existing platforms. Thus, each change can be statically analyzed on the conceptual view before being propagated to the software system. Dynamic analysis are based on data values available during the execution only. The capture of these values is done through automatic instrumentation of the software system.

Globally, CALICO enables reliable evolution even if the underlying platforms does not natively provide this support. The current version handles four component-based and service-oriented platforms. Moreover, the benchmarks that we have performed show that CALICO is usable for the design and development of safe applications up to 10,000 components and services, which corresponds to the maximal load of most runtime platforms. CALICO has been developed in the context of Guillaume Waignier’s PhD thesis [79].

CALICO is an open source software available at http://calico.gforge.inria.fr.

5.3. Fractal

Participants: Christophe Demarey, Philippe Merle [correspondant], Romain Rouvoy, Lionel Seinturier.

Fractal is a modular, extensible and programming language agnostic component model that can be used to design, implement, deploy and reconfigure systems and applications, from operating systems, middleware platforms to graphical user interfaces [53], [55], [67]. Fractal has been designed by both Inria and France Telecom R&D.

Fractal is also a LGPL open source software project hosted by the OW2 international consortium and is available at http://fractal.ow2.org [59].
Philippe Merle is the leader of the OW2 FRACTAL open source project. The ADAM project-team actively contributes to this project, and more specifically on the following modules:

- **AOKell** is an aspect-oriented implementation of the FRACTAL component model [76].
- **Fraclet** is an attribute-oriented programming model enabling the rapid development of FRACTAL components [73].
- **Fractal ADL** is the extensible architecture definition language for FRACTAL associated to an open FRACTAL component-based toolchain.
- **Fraclet Distribution** is the module to produce packaged releases of the FRACTAL project.
- **Fraclet Documentation** is the module to produce the whole documentation of the FRACTAL project.
- **Fraclet Eclipse Plugin** is a plugin to create FRACTAL projects within the Eclipse IDE [57], [58]. This work was supported by an Inria ODL and is contributed to the FUI MIND project.
- **Fraclet Explorer** is a framework to build graphical consoles to introspect and manage FRACTAL components dynamically at runtime.
- **FScript** is a scripting language for both introspection and reconfiguration of FRACTAL software systems.
- **Juliac** is an extensible framework for generating and compiling the code of FRACTAL component-based systems. Juliac is registered with the APP (Agence pour la Protection des Programmes) under reference FR.001.230007.000.S.P.2009.000.10600.
- **Koch** is an implementation of the FRACTAL component model where components have a component-based control membrane.

### 5.4. FraSCAti

**Participants:** Christophe Demarey, Damien Fournier, Rémi Méliisson, Philippe Merle [correspondant], Christophe Munilla, Romain Rouvoy, Lionel Seinturier.

FraSCAti is a runtime platform for the Service Component Architecture (SCA) component framework. SCA is an initiative for unifying Service Oriented Architectures (SOA) and Component-Based Software Engineering (CBSE). SCA is supported by the Open SOA consortium, which includes partners, such as IBM, Oracle, Sun and Iona, and is standardized by the OASIS consortium (see at [http://www.oasis-openesca.org/sc] ).

FraSCAti includes Tinfi, which provides a SCA personality for the FRACTAL component model. Thanks to the openness of this latter model, the necessary code elements (so called controllers and membranes) have been designed and developed to customize FRACTAL and to end up with components owning both a FRACTAL personality and a SCA personality. As far as we know, this result, which has been presented in [75], is original and is the first one to concretely demonstrates that FRACTAL is open and flexible enough to implement different component personalities. Moreover, Tinfi reuses the aspect-oriented concepts defined in FAC [70] for component-based programming and allows integrating smoothly non functional concerns (so called intents and policy sets in SCA terms). FraSCAti and Tinfi have been implemented by reusing modules developed in the context of the FRACTAL project, and among others, the Juliac FRACTAL compiler.

The development of the FraSCAti platform is conducted in the context of some current and past funded projects (ICT FP7 SOA4All Integrated Project, ANR ARPEGE ITEmIS project, FUI EasySOA project, Inria ADT Adapt).

FraSCAti is a LGPL open source software, hosted by the OW2 consortium since November 2008 at [http://frascati.ow2.org](http://frascati.ow2.org). FraSCAti is registered with the APP (Agence pour la Protection des Programmes) under reference FR.001.050017.000.S.P.2010.000.10000.

### 5.5. SPACES

**Participants:** Russel Nzekwa, Daniel Romero [correspondant], Romain Rouvoy, Lionel Seinturier.
SPACES is a context mediation middleware that follows the REpresentational State Transfer (REST) principles \[ 61 \]. The current implementation of SPACES is based on the COSMOS context framework \[ 54 \], \[ 72 \] and the COMANCHE web server \[ 53 \]. Both COSMOS and COMANCHE are based on the FRACTAL component model and use the JULIA implementation of the FRACTAL runtime environment \[ 53 \].

The main features of the current SPACES implementation are presented below:

1. **Ubiquitous connectors**: SPACES defines connectors that encapsulate the distribution concern. These connectors expose the COSMOS context nodes as REST resources with logical associated URLs, and enable interactions between consumers and producers via different communication protocols and the discovery of the available context sources. The current SPACES implementation supports interaction using the HTTP and twitter \[ 65 \] protocols. For discovery, the implementation uses the Service Location Protocol (SLP) \[ 63 \].

2. **Context Representation**: Following the REST principles, SPACES supports multiple representations of the context information: JSON \[ 56 \], XML and Java serialization.

3. **Quality of context (QoC) information**: The QoC properties are incorporated as service attributes in the SLP advertisements of the context information.

4. **Context selection**: The restrictions in terms of QoC of the required context information are expressed as LDAP filters \[ 77 \]. SPACES benefits from the LDAP based queries of SLP to select the context providers.

We use XStream 1.3.13 \[ 51 \] and JSON-lib 2.2.34 \[ 50 \] to serialize context information as XML and JSON documents. For SLP and twitter we employ jSLP 1.0.0 \[ 71 \] and twitter-4j 2.0.6 \[ 80 \].

SPACES is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN 10-5000002-000.

### 5.6. ApplIDE

**Participants**: Laurence Duchien, Christophe Demarey, Clément Quinton [correspondant].

ApplIDE is directly connected to the work of Carlos Parra’s PhD and Ubinov ADT’s work which covers the definition and implementation of a Context-Aware Dynamic Software Product Line (DSPL) named CAPucine. It provides a set of tools for selection of features, metamodel transformation and code generation for mobile applications \[ 40 \]. The current implementation of ApplIDE addresses transformation from CAPucine metamodel towards SCA metamodel, and Spoon EMF metamodel. The transformations were formerly written with Acceleo tool, which is a dedicated language for transformation, enhancing the readability. ApplIDE meta models are based on the Eclipse Modeling Framework. Code generators are all written in Acceleo.

ApplIDE is registered with the APP (Agence pour la Protection des Programmes) under reference IDDN.FR.001.500004.000.S.A.2010.000.10600.
5. Software

5.1. parXXL

Participants: Jens Gustedt, Stéphane Vialle.

parXXL is a library for large scale computation and communication that executes fine grained algorithms (computation and communication are of the same order of magnitude) on coarse grained architectures (clusters, grids, mainframes). Historically, parXXL is the result of a collaboration between INRIA and SUPÉLEC. This library fulfills the requirements of our model PRO, i.e., it uses an alternation of computation and communication steps. It realizes an abstraction layer between the algorithm as it was designed and its realization on different architectures and different modes of communication. The current version of this library has been registered at the APP and is available at http://parxxl.gforge.inria.fr/. It integrates a layer for message passing with MPI, a layer for shared memory with POSIX threads, a layer for out-of-core management with file mapping (system call mmap).

All three different realizations of the communication layers are quite efficient. They let us execute programs that are otherwise unchanged within the three different contexts. Usually, they reach the performance of programs that are directly written for a given context. Generally they outperform programs that are executed in a different context than they were written for, such as MPI programs that are executed on a shared memory mainframe, or such as multi-threaded programs that are executed on a distributed shared memory machine.

5.2. Distem

Participants: Tomasz Buchert, Emmanuel Jeanvoine, Lucas Nussbaum, Luc Sarzyniec.

Distem is a distributed systems emulator. In the context of research on Cloud, P2P, High Performance Computing or Grid systems, it can be used to transform an homogeneous cluster (composed of identical nodes) into an experimental platform where nodes have different performance, and are connected together through a complex network topology, thus facilitating the evaluation or benchmarking of applications targeting such environments.

Distem relies on modern Linux features (LXC, cgroups, cpufreq, iptables, traffic control) to steal resources from applications. At the node level, it provides the ability to introduce heterogeneity by splitting a multi-core node into several virtual nodes of varying number of cores and CPU frequency. At the network level, it allows the user to describe and build virtual network topologies where each link has a given latency, and bandwidth limit.

Distem is controlled through a REST API to ease its integration into experiment scripts, but also provides a Ruby library and a command-line interface.

It has been registered with the APP, and is freely available under the GNU GPL.

More information is available from http://distem.gforge.inria.fr/.

5.3. Wrekavoc

Participants: Jens Gustedt, Lucas Nussbaum, Tomasz Buchert.

Wrekavoc addresses the problem of controlling the heterogeneity of a cluster to provide a configurable environment that allows for reproducible experiments on large sets of configurations using real applications with no emulation of the code. Work on Wrekavoc has stopped: current works are based on the Distem emulator.
5.4. SimGrid

**Participants:** Pierre-Nicolas Clauss, El Mehdi Fekari, Martin Quinson, Lucas Nussbaum, Cristian Rosa, Christophe Thiéry.

The SimGrid framework aims at being a scientific instrument to the evaluation of algorithmic solutions for large-scale distributed experiments. It is the result of a collaboration with Henri Casanova (Univ. of Hawaii, Manoa) and Arnaud Legrand (MESCAL team, INRIA Grenoble-Rhône-Alpes, France). Simulation is a common answer to the grid specific challenges such as scale and heterogeneity. SimGrid is one of the major simulators in the Grid community.

The main strong point of this is its carefully assessed *model validity*. To this end, the simulation kernel relies on a blend of analytical models and coarse-grain discrete event simulation. It proves several orders of magnitude faster than usual packet-level simulators used in the networking community (such as ns2 or GTNetS) while providing a good level of accuracy [43].

The SimGrid framework is currently extremely fast. Independent authors demonstrated its superior scalability over its main concurrence [36], [38]. In addition to the efficiency of the simulation models, this *scalability* is ensured by a layered architecture, with a simulation kernel computing the time taken by *actions* which need to consume *resources* to complete. Another layer of abstraction introduces the notion of processes and network routing between hosts. On top of this come the user interfaces aiming at providing the syntactic sugar easing the tool usage.

Several such user interfaces exist, ensuring the *versatility* of the SimGrid framework by adapting to the user goal: *MSG* helps the study of distributed heuristics. This is the historical interface of SimGrid, and remains the most used one. *SMPI* is a new interface which allows the simulation of MPI programs designed for multi-processor systems on a single computer [4]. *SimDag* eases the study of scheduling heuristics for DAGs of (parallel) tasks, which helps the work on parallel task scheduling. *GRAS* (Grid Reality And Simulation) eases the development of Grid services and infrastructures [39] through a specific interface implemented twice: once on top of the simulator for the comfort of development, and once using regular sockets for live deployments.

SimGrid can be freely downloaded from its *web page* and its user base is rapidly growing. Over the last decade, it grounded the experimental section of more than hundred scientific publications, not counting the ones being co-authored by members of the development team.

5.5. ORWL

**Participant:** Jens Gustedt.

ORWL is a reference implementation of the Ordered Read-Write Lock tools as described in [3]. It implements interfaces for locking and data management that easily allow to have an overlap between communication and computation. The main tool here is the introduction of a “handle” on a local or remote resource that can be used to trigger asynchronous prefetching of control and/or data. Also it implements a second layer of abstraction for the seamless programming of iterative tasks. With that layer iterative algorithms can be implemented that have guarantees for equity and deadlock-freeness.

ORWL is a standalone library that works on shared memory and in distributed settings. The implementation is uniquely based on C99 and POSIX. ORWL has already been registered at the APP. Final tests and benchmarks are on the way to ensure the quality of the implementation before it will be made publicly available.

5.6. P99

**Participant:** Jens Gustedt.

P99 is a toolbox of header files designated to ease programming in C, in particular modern C99. Originally, these macro definitions and tools for programming in C99 have been implemented for ORWL, but now they are separated out into a separate toolbox.
This toolbox allows e.g. the simplified use of variable length argument list for macros and functions, default arguments of functions, compile time code unrolling, scope bound resource management, transparent allocation and initialization. It has been registered at the APP and is available at http://p99.gforge.inria.fr/.
AMAZONES Team

5. Software

5.1. Logos

Participants: Julien Ponge, Stéphane Frénot.

Logos is a development project linked to the LISE ANR grant. It’s goal is to generate execution logs from OSGi services interactions. The main idea is to intercept every service call and generate an entry in a log file. The log file system should be used in the LISE context which is related to legal issues. Generated Logos logs should be: Complete, encoded with a cryptographic algorithm, compact and immutable.

The software is currently used as a Amazones internal test suite. It is fully tested on standard OSGi architectures.

5.2. Logminer

Participants: Julien Ponge, Stéphane Frénot.

LogMiner is a toolbox, written in Scala in current development. The LogMiner framework takes Logos inputs and generates service usage automata. The goal of logminer is to represent application activity in a synthetic way in order to identify behavioral changes while updating the system. When one updates its applications on its environment, the logminer framework enables observation and identifies variations in service usages.

The software is currently under development it integrates a automata generator and a data visualisations modules.

5.3. Eimc

Participants: Zheng Hu, Stéphane Frénot, Bernard Tourancheau [Projet Swing].

Eimc is an architecture for managing sensor dedicated to legacy equipment management. The project aims at designing a dynamic framework that integrates sensors from the surrounding environment and detects new equipments from their physical behavior. For instance, a fridge vibrates when the compressor is working. The frequency of vibrations distinguishes a fridge from washing machine. The framework designs a Complex Based Event processing architecture where we need to focus on the number of manageable equipments, the number of deployed sensors and the number of physical measurements that can be handled.

The project is a joint project with Orange Labs, and a PhD student Zheng Hu. He is co-directed by Stéphane Frénot and Bernard Tourancheau from Amazones and Swing teams.

5.4. Aoraï

Participant: Nicolas Stouls.

Developed at CEA-LIST, Frama-C is an extensible and collaborative platform dedicated to source-code analysis of C software. The Aoraï [49] plug-in for Frama-C [31] provides a method to automatically annotate a C program according to a behavioral property P such that, if the annotations are verified, then we ensure that the program respects P.

The computation process is divided into two steps: the specification generation from the property and the constraints propagation for static simplification. According to the classical invariant verification granularity, observable states of a program correspond to each call or return statements of an operation. Each state of the program is associated to a set of transitions in an internal representation of the property, managed as a Büchi automata. Starting from a super-set of authorized behaviors, some static simplifications can be done in order to generate sufficient pre/post-conditions on each operation.
The classical method to validate generated annotations is to use the Jessie plug-in and the Why tool, using theorem provers.

A new research report [50] as been published and some new developments has been done in order to increase consequently the efficiency of the tool.

5.5. STOP

Participants: François Goichon, Stéphane Frénot, Pierre Parrend.

STOP is a security-oriented program analysis toolkit developed by François Goichon as part of his masters thesis. He was supervised by Stéphane Frénot and Pierre Parrend from FZI, Karlsruhe.

The tool implements a novel static analysis technique called Service-oriented Tainted Object Propagation, described in more detail in the Results section.

5.6. IzPack

Participant: Julien Ponge.

IzPack [47] is a software installer creation framework for the Java platform. Its main differentiator with respect to the other installation solutions is that it generates cross-platform installers that can adapt themselves to the underlying operating system so as to still provide tight integration. It was also designed to be highly customizable and extensible.

IzPack is nearing its 10 years landmark. It is hosted at the Coddehaus [32] Foundation and released under the terms of the Apache Software License version 2.0. Its users community non-exhaustively comprise Spring-Source, JBoss / RedHat, Oracle / Sun Microsystems, the Scala language, XWiki, Terracotta or Silverpeas.

The project was originally created by now INRIA Amazones team member Julien Ponge, who still leads the project. In 2010, it was presented at the Devoxx conference.

5.7. WSNet

Participants: Guillaume Chelius [INRIA D-NET Team, project leader], Antoine Fraboulet, Loïc Lemaître [INRIA SensTools IJD].

WSNet is a modular wireless network simulator. It incorporates the following aspects: (i) accurate simulation of the radio channel: Supports MIMO, multi-interface, multi-channel, etc. (ii) simulation environment: simulation of the interaction between sensors and their environment: measurement and control, simulation of device power consumption. Furthermore, WSNet can be interfaced with the WSim sensor node emulator to form a distributed emulation of a sensor network.

WSNet source code is registered at the Agency For The Protection Of Programs (APP IDDN 06-370013-000), Licence: CeCILL (2). See also the web page http://wsnet.gforge.inria.fr/.

5.8. WSim

Participants: Guillaume Chelius [INRIA D-NET Team], Antoine Fraboulet [Project leader], Loïc Lemaître [INRIA SensTools IJD], Julien Carpentier [INRIA ORSI IJD].

WSim is a platform emulator for embedded systems allowing performance evaluation and programming assistance during the application design stages of distributed wireless sensor networks. WSIM is a simulation tool enabling a rapid and relevant feedback on features and quality of embedded software in constrained systems. Its simulation model allows to interface with other tools like WSNet to build complex simulation environments.

WSim source code is registered at the Agency For The Protection Of Programs (APP IDDN 06-370012-000), Licence: CeCILL (2). See also the web page http://wsim.gforge.inria.fr/.
5.9. Esimu

**Participant:** Antoine Fraboulet.

eSimu is a complete system energy model based on non-intrusive measurements. This model aims at being integrated in fast cycle accurate simulation tools to give energy consumption feedback for embedded systems software programming. Estimations take into account the whole system consumption including peripherals. Experiments on a complex ARM9 platform show that our model estimates are in error by less than 10% from real system consumption, which is precise enough for source code application design, while simulation speed remains fast. eSimu can be used as a standalone tool or in conjunction with WSim.

Licence: CeCILL (2). See also the web page [http://esimu.gforge.inria.fr/](http://esimu.gforge.inria.fr/).

5.10. ABR

**Participants:** Frédéric Le Mouël, Stéphane Frénot.

The Ambient Bundle Repository (ABR) is an OSGi extension, compliant with the Bundle Repository API. Instead of proposing a centralized discovery as the default bundle repository implementation, ABR abstracts different discovery protocols (UPnP, ...) and publishes/subscribes a local repository containing bundles in a device geographically-close environment. ABR implements mobility models to track mobile devices, to warn the user deploying bundles of the remaining presence time of bundles and to anticipate a possible bundle deployment non-ending.

5.11. AxSeL

**Participants:** Amira Ben Hamida, Frédéric Le Mouël, Stéphane Frénot.

While installing and executing applications on mobile devices, the issue of the limit of resources is quickly encountered.

AxSeL (A conteXtual Service Loader) is an OSGi prototype extension that modifies the bundle loading at deployment time for a context-aware service loading at run time. The approach is based on a service graph colouring process. We represent an application as a bi-dimensional dynamic graph with services and bundles dependencies. The colouring decision provides an optimal deployment configuration of the application in a given context. Context listening mechanisms capture changes and propagate recolouring and redeployment processes.

Context elements currently implemented and monitored are the hardware memory and disk sizes. Application currently implemented and tested is a service-oriented PDF viewer that is adapting its display to available device resources [6].

This prototype is a part of the PhD thesis of Amira Ben Hamida [29].

5.12. QuestMonitor

**Participants:** Stéphane Grumbach, Ahmad Ahmad-Kassem, Fuda Ma.

QuestMonitor [28] is a visualization tool that allows to visualize dynamic networks, and monitor the execution of protocols written in the data centric language Netlog. The language allows to specify protocols which in sometimes their behavior, in dynamic networks, are tricky to understand. QuestMonitor allows to monitor all the communication between the nodes, the evolution of the data stores on each node, as well as the execution of the declarative code. It also allows to color the virtual data structures, such as routes, backbones, etc. Together with the code editing facility, it constitutes a good tool for rapid prototyping.

Amazones team aimed at bridging the gap between "high-level" developed architectures that we called the northBound and "low-level" run-time, the southBound. Northbound architecture rely on virtual machines and advanced development languages, whereas southbound architectures rely on micro-kernels and drivers development. Our results are mainly initial studies since we fixed our research team on November 2010, and the end notification arrived on June 2011. Although our time frame was short we managed to gain knowledge in three areas linked to Amazones goals.
5. Software

5.1. Introduction

In order to validate our research results, our research activities encompass the development of related prototypes as surveyed below.

5.2. Emergent Middleware Enablers

Participant: Valérie Issarny [correspondent].

As part of our research work on Emergent Middleware, we have implemented Enablers (or Enabler functionalities) that make part of the overall CONNECT [30] architecture realizing Emergent Middleware in practice.

**Discovery Enabler:** The CONNECT Discovery Enabler is the component of the overall CONNECT architecture that handles discovery of networked systems (NSS), stores their descriptions (NS models), and performs an initial phase of matchmaking to determine which pairs of systems are likely to be able to interoperate. Such pairs are then passed to the Synthesis Enabler so that mediators can be generated. The Discovery Enabler is written in Java and implements several legacy discovery protocols including DPWS and UPnP. Matchmaking is done on the basis of affordances contained in the system description, that is, ontological concepts describing the system’s category. Systems with the same affordance, or affordances standing in a specialization relationship, can be considered for connection. If a system does not provide its affordance, the Discovery Enabler can infer a likely one using text categorization based on the system’s interface description. The Discovery Enabler will soon be available for download under an open source license.

**Synthesis Enabler:** We have implemented (in Java) two approaches to mediator synthesis as part of the CONNECT Synthesis Enabler:

- **Mapping-based mediator synthesis.** This implementation focuses on networked systems that have compatible functionalities but are unable to interact successfully due to mismatching interfaces and/or behaviors. The ontology used in our implementation is encoded so as to make the reasoning more efficient at runtime while considering both subsumption and the union of classes. Based on the interface mapping, a correct-by-construction mediator is generated. In our current implementation, we are leveraging the LTSA (Labeled Transition System Analyser) model checker to generate the parallel composition of the mapping processes and verify that the overall system successfully terminates. In the near future, we will be incorporating our techniques so as to deal with ambiguous mappings, i.e., when an action from one system may semantically be mapped to different actions from the other system.

- **Goal-based abstract mediator synthesis.** This implementation considers the protocols of two networked systems and produces the mediator protocol that allow them to interact so as to satisfy user goals. More specifically, the alphabet of the two protocols are aligned using ontology matching. The aligned protocols as well as the user goal are encoded as a satisfiability problem. The Zot model checker solves this problem (if possible) and produces a possible feasible interaction satisfying user goals.

The synthesis enabler will soon be available for download under an open source license.

5.3. Service-oriented Middleware for Pervasive Computing

Participants: Nikolaos Georgantas [correspondent], Valérie Issarny [correspondent].
In the past years, we have built a strong foundation of service-oriented middleware to support the pervasive computing vision. This specifically takes the form of a family of middlewares, all of which have been released under the open source LGPL license:

- **WSAMI - A Middleware Based on Web Services for Ambient Intelligence:** WSAMI (Web Services for AMbient Intelligence) is based on the Web services architecture and allows for the deployment of services on wireless handheld devices like smartphones and PDAs.
  URL: [http://www-rocq.inria.fr/arles/download/ozone/index.htm](http://www-rocq.inria.fr/arles/download/ozone/index.htm)

- **Ariadne - A Protocol for Scalable Service Discovery in MANETs:** Ariadne enriches WSAMI with the Ariadne service discovery protocol, which has been designed to support decentralized Web service discovery in multi-hop mobile ad hoc networks (MANETs). Ariadne enables small and resource-constrained mobile devices to seek and find complementary, possibly mobile, Web services needed to complete specified tasks in MANETs, while minimizing the traffic generated and tolerating intermittent connectivity.

- **MUSDAC - A Middleware for Service Discovery and Access in Pervasive Networks:** The MULTi-protocol Service Discovery and ACcess (MUSDAC) middleware platform enriches WSAMI so as to enable the discovery and access to services in the pervasive environment, which is viewed as a loose and dynamic composition of independent networks. MUSDAC manages the efficient dissemination of discovery requests between the different networks and relies on specific plug-ins to interact with the various middleware used by the networked services.

- **INMIDIO - An Interoperable Middleware for Ambient Intelligence:** INMIDIO (INteroperable MIddleware for service Discovery and service InteractiOn) dynamically resolves middleware mismatch. More particularly, INMIDIO identifies the interaction middleware and also the discovery protocols that execute on the network and translates the incoming/outgoing messages of one protocol into messages of another, target protocol.

- **COCOA - A Semantic Service Middleware:** COCOA is a comprehensive approach to semantic service description, discovery, composition, adaptation and execution, which enables the integration of heterogeneous services of the pervasive environment into complex user tasks based on their abstract specification. Using COCOA, abstract user tasks are realized by dynamically composing the capabilities of services that are currently available in the environment.
  URL: [http://gforge.inria.fr/projects/amigo/](http://gforge.inria.fr/projects/amigo/)

- **ubiSOAP - A Service Oriented Middleware for Seamless Networking:** ubiSOAP brings multi-radio, multi-network connectivity to services through a comprehensive layered architecture: (i) the multi-radio device management and networking layers together abstract multi-radio connectivity, selecting the optimal communication link to/from nodes, according to quality parameters; (ii) the communication layer allows for SOAP-based point-to-point and group-based interactions in the pervasive network; and (iii) the middleware services layer brings advanced distributed resource management functionalities customized for the pervasive networking environment.
  URL: [http://www.ist-plastic.org](http://www.ist-plastic.org)

### 5.4. Supporting Service Orchestrations over Heterogeneous Interaction Paradigms

**Participant:** Nikolaos Georgantas [correspondent].
Established architectural paradigms enabling open system integration, such as service oriented architecture (SOA) and enterprise service bus (ESB), have provided answers to the essential issue of interoperability in distributed systems. However, realizations of these architectural paradigms fall short when it comes to integrating systems featuring heterogeneous interaction paradigms, such as client/server (CS), publish/subscribe (PS) and tuple space (TS), due to the differing interaction semantics of the latter. Typical solutions constitute in wrapping any system behind RPC-based service interfaces, which results in partial loss of interaction semantics. This can cause suboptimal or even problematic system integration.

Aiming at enabling seamless integration of heterogeneous interaction paradigms, we introduce an interoperability solution based on abstraction and merging of their common high-level semantics, paying special attention to the preservation of semantics. To this end, we propose three abstract connector types for the CS, PS and TS interaction paradigms. We further introduce a higher-level generic application (GA) connector type, which provides an abstract union of the three models, thus preserving their interaction semantics. We express our connector types in terms of application programming interface (API) primitives and related interaction protocol semantics. We then apply our abstractions to rethink a typical SOA- and ESB-based orchestration of heterogeneous distributed systems. Our solution features:

- Extending the BPEL workflow language with GA API primitives in terms of extension activities enabled by the BPEL specification;
- Introducing XSLT transformation between the GA-extended BPEL and the standard BPEL, which consists of encapsulating GA primitives into standard BPEL primitives and enables conveying GA semantics on top of BPEL primitives and subsequently on top of the common bus protocol primitives;
- Providing Java code templates for systematic and highly facilitated building of ESB-embedded binding components;
- Proposing interface description languages in the form of XSDs for systems employing CS-, PS-, TS-, and GA-connectors; and
- Introducing XSLT transformations between native system interface descriptions and GA-based interface descriptions.

We have developed our solution on top of the PEtALS ESB, which provides inherent support for BPEL by embedding the EasyBPEL workflow engine. Our solution considerably facilitates the application developer in designing and executing heterogeneous orchestrations. Furthermore, it is highly extensible, enabling easy integration of support for new middleware platforms. To demonstrate the applicability of our approach, we have implemented an application workflow integrating a JMEDS DPWS Web Service (CS), a JMS system based on Apache ActiveMQ (PS), and a Jini JavaSpaces system (TS). Our software will soon be released under open source license.

5.5. Srijan: Data-driven Macroprogramming for Sensor Networks

**Participant:** Animesh Pathak [correspondent].

Macroprogramming is an application development technique for wireless sensor networks (WSNs) where the developer specifies the behavior of the system, as opposed to that of the constituent nodes. As part of our work in this domain, we are working on Srijan, a toolkit that enables application development for WSNs in a graphical manner using data-driven macroprogramming.

It can be used in various stages of application development, viz.,

1. Specification of application as a task graph,
2. Customization of the auto-generated source files with domain-specific imperative code,
3. Specification of the target system structure,
4. Compilation of the macroprogram into individual customized runtimes for each constituent node of the target system, and finally
5. Deployment of the auto generated node-level code in an over-the-air manner to the nodes in the target system.
The current implementation of Srijan targets both the Sun SPOT sensor nodes and larger nodes with J2SE. Most recently, Srijan also includes rudimentary support for incorporating Web services in the application being designed. The software is released under open source license, and available as an Eclipse plug-in at http://code.google.com/p/srijan-toolkit/.

5.6. Yarta: Middleware for supporting Mobile Social Applications

Participant: Animesh Pathak [correspondent].

With the increased prevalence of advanced mobile devices (the so-called “smart” phones), interest has grown in Mobile Social Ecosystems (MSEs), where users not only access traditional Web-based social networks using their mobile devices, but are also able to use the context information provided by these devices to further enrich their interactions. We are developing a middleware framework for managing mobile social ecosystems, having a multi-layer middleware architecture consisting of modules, which will provide the needed functionalities, including:

- Extraction of social ties from context (both physical and virtual),
- Enforcement of access control to protect social data from arbitrary access,
- A rich set of MSE management functionalities, using which mobile social applications can be developed.

Our middleware adopts a graph-based model for representing social data, where nodes and arcs describe socially relevant entities and their connections. In particular, we exploit the Resource Description Framework (RDF), a basic Semantic Web standard language that allows representing and reasoning about social vocabulary, and creating an interconnected graph of socially relevant information from different sources.

The current implementation of the Yarta middleware targets both desktop/laptop nodes running Java 2 SE, as well as Android smart phones. The software is released under open source license at https://gforge.inria.fr/projects/yarta/.

5.7. iBICOOP: Mobile Data Management in Multi-* Networks

Participant: Valérie Issarny [correspondent].

Building on the lessons learned with the development of pervasive service oriented middleware and of applications using them, we have been developing the custom iBICOOP middleware. iBICOOP specifically aims at assisting the development of advanced mobile, collaborative application services by supporting interactions between mobile users. Target application services in particular include the U-EVENT suite of services for professional events.

Briefly, the iBICOOP middleware addresses the challenges of easily accessing content stored on mobile devices, and consistent data access across multiple mobile devices by targeting both fixed and mobile devices, leveraging their characteristics (e.g., always on and unlimited storage for home/enterprise servers, ad hoc communication link between mobile devices), and by leveraging the capabilities of all available networks (e.g., ad hoc networks, Internet, Telecoms infrastructure networks). It also relies on Web and Telecoms standards to promote interoperability.

The base architecture of the iBICOOP middleware consists of core modules on top of which we can develop applications that may arise in the up-coming multi-device, multi-user world:

- The Communication Manager provides mechanisms to communicate over different available network interfaces of a device — Bluetooth, WiFi, Cellular — and also using different technologies e.g., Web services, HTTP/TCP sockets, ad hoc mode.
- The Security Manager uses well-established techniques of cryptography and secure communication to provide necessary security.
- The Partnership Manager provides device or user information in the form of profiles.
• iBICOOP relies on service location protocols for naming and discovery of nearby services on currently active network interfaces that support IP multicast.

• Besides normal file managing tasks, the Local File Manager gives the user clear cues to the files that have been replicated across multiple devices or shared among different users by using different icons.

The iBICOOP middleware has been licensed by AMBIENTIC (http://www.ambientic.com/), a start-up that specifically develops innovative mobile distributed services on top of the iBICOOP middleware that allows for seamless interaction and content sharing in today’s multi-* networks.

5. Software

5.1. WhatsUp: A Distributed News Recommender

Participants: Antoine Boutet, Davide Frey, Arnaud Jegou, Anne-Marie Kermarrec.
Contact: Antoine Boutet
Licence: Open Source
Presentation: A Distributed News Recommender
Status: Beta version

This work has lead to the development of WhatsUp, a distributed recommendation system aimed to distribute instant news in a large scale dynamic system. WhatsUp has two parts, an embedded application server in order to exchange with others peers in the system and a fully dynamic web interface for displaying news and collecting opinions about what the user reads. Underlying this web-based application lies Beep, a biased epidemic dissemination protocol that delivers news to interested users in a fast manner while limiting spam. Beep is parametrized on the fly to manage the orientation and the amplification of news dissemination. Every user forwards the news of interest to a randomly selected set of users with a preference towards those that have similar interests (orientation). The notion of interest does not rely on any explicit social network or subscription scheme, but rather on an implicit and dynamic overlay capturing the commonalities between users with respect to they are interested in. The size of the set of users to which a news is forwarded depends on the interest of the news (amplification). A centralized version of WhatsUp is already up and running and the decentralized one is still in beta version.

5.2. GossipLib: effective development of gossip-based applications

Participants: Davide Frey, Heverson Ribeiro, Anne-Marie Kermarrec.
Contact: Davide Frey
Licence: Open Source
Presentation: Library for Gossip protocols
Status: released version 0.7alpha

GossipLib is a library consisting of a set of JAVA classes aimed to facilitate the development of gossip-based application in a large-scale setting. It provides developers with a set of support classes that constitute a solid starting point for building any gossip-based application. GossipLib is designed to facilitate code reuse and testing of distributed application and as thus also provides the implementation of a number of standard gossip protocols that may be used out of the box or extended to build more complex protocols and applications. These include for example the peer-sampling protocols for overlay management.

GossipLib also provides facility for the configuration and deployment of applications as final-product but also as research prototype in environments like PlanetLab, clusters, network emulators, and even as event-based simulation. The code developed with GossipLib can be run both as a real application and in simulation simply by changing one line in a configuration file.

5.3. YALPS

Participants: Davide Frey, Heverson Ribeiro, Anne-Marie Kermarrec.
Contact: Davide Frey
Licence: Open Source
Presentation: Library for Gossip protocols
Status: released version 0.3alpha
YALPS is an open-source Java library designed to facilitate the development, deployment, and testing of distributed applications. Applications written using YALPS can be run both in simulation and in real-world mode without changing a line of code or even recompiling the sources. A simple change in a configuration file will load the application in the proper environment. A number of feature make YALPS useful both for the design and evaluation of research prototypes and for the development of applications to be released to the public. Specifically, YALPS makes it possible to run the same application as a simulation or in a real deployment without a single change in the code. Applications communicate by means of application-defined messages which are then routed either through UDP/TCP or through YALPS’s simulation infrastructure. In both cases, YALPS’s communication layer offers features for testing and evaluating distributed protocols and applications. Communication channels can be tuned to incorporate message losses or to constrain their outgoing bandwidth. Finally, YALPS includes facilities to support operation in the presence of NATs and firewalls using relaying and NAT-traversal techniques.

The work has been done in collaboration with Maxime Monod (EPFL).

5.4. HEAP: Heterogeneity-aware gossip protocol.

Participants: Davide Frey, Arnaud Jegou, Anne-Marie Kermarrec.

Contact: Davide Frey
Licence: Open Source
Presentation: Java Application
Status: release & ongoing development

This work has been done in collaboration with Vivien Quéma (CNRS Grenoble), Maxime Monod and Rachid Guerraoui (EPFL), and has lead to the development of a video streaming platform based on HEAP, Heterogeneity-Aware gossip Protocol. The platform is particularly suited for environment characterized by heterogeneous bandwidth capabilities such as those comprising ADSL edge nodes. HEAP is, in fact, able to dynamically leverage the most capable nodes and increase their contribution to the protocol, while decreasing by the same proportion that of less capable nodes. During the last few months, we have integrated HEAP with the ability to dynamically measure the available bandwidth of nodes, thereby making it independent of the input of the user.
ASCOLA Project-Team

5. Software

5.1. AWED

Participants: Mario Südholt [correspondent], Ismael Mejia.

The model of Aspects With Explicit Distribution (AWED) supports the modularization of crosscutting functionalities of distributed applications. It addresses the problem that common aspect systems do not provide features for distributed programming. It notably features three main aspect abstractions: remote pointcuts, remotely-executed advice, and distributed aspects.

This year a gray-box model for distributed composition has been built and implemented based on the AWED model using the notion of invasive distributed patterns (see Sec. 6.1). Furthermore, the resulting model has been applied to the evolution of grid applications and OpenMRS, an open-source health information system. The AWED system has also been employed in the CESSA project proposal (see Sec. 8.1) as a basis for our work on the secure evolution of service-oriented architectures. Finally, the development of a new, more modular, implementation of the AWED system has started in 2011.

AWED is available at http://awed.gforge.inria.fr.

5.2. ECaesarJ, EJava and EScala

Participants: Jacques Noyé [correspondent], Angel Núñez, Jurgen Van Ham.

ECaesarJ is a language developed in the context of the European project AMPLE, as joint work with the Technische Universität Darmstadt (TUD). The basic objective was to provide support for directly mapping the high-level features defined by a software product line onto implementation-level features, beyond standard feature-oriented programming. But the language has much wider applications. ECaesarJ can actually be seen as a language which smoothly integrates Object-Oriented Programming, Feature-Oriented Programming, Aspect-Oriented Programming, and Event-based Programming.

It is an extension of Java with virtual classes and propagating mixin composition (as its ancestor CaesarJ, developed at TUD), but also declarative events and state machines. Unlike AspectJ, ECaesarJ does not include a class-like concept of aspect. Instead, it deals with pointcuts and pieces of advice as (implicit) events and event handlers, which are standard class members. This makes it possible to use standard inheritance to reuse and refine them. Explicit events can also be used when events must be explicitly triggered as in traditional event-based programming. Finally, in the same way as pointcuts can be composed using logical operators, declarative events can be defined as a composition of other events.

This provides a symmetric version of AOP where virtual classes can be used to deal with structural aspects whereas events can be used to deal with behavioral aspects.

In ECaesarJ, a class can also include, as class members, state transitions. Combining this with virtual classes makes it possible to define, at the programming language level, refinable hierarchical state machines. The combination of state machines and events provides, in particular, effective language support for the State design pattern as well as a form of Event-based AOP.

EJava and EScala are more recent developments of the same ideas applied to Java and Scala, respectively. EJava benefits from Java tooling with an eclipse plugin developed with the Spoofax Language Workbench. Unlike EJava and ECaesarJ, EScala makes it possible to dynamically register and unregister event handlers. It also benefits from a more efficient, compiler-based, implementation. As ECaesarJ, EScala is joint work with TUD.

Prototype implementations of these languages are available through http://ecaesarj.gforge.inria.fr/.
5.3. Entropy

Participants: Jean-Marc Menaud [correspondent], Fabien Hermenier, Adrien Lèbre, Hien Nguyen Van, Rémy Pottier, Thomas Chavrier, Guillaume Le Louët.

Entropy is a virtual machine (VM) manager for clusters. The current prototype acts as an infinite control loop, which performs a globally optimized placement according to cluster resource usage, scheduler objectives and administrative rules.

Relying on an encapsulation of jobs into VMs, Entropy enables the implementation of finer scheduling policies through cluster-wide context switches, i.e., permutations of VMs present in the cluster. It thus supports a more flexible use of cluster resources and frees end-users from the burden of dealing with time estimates.

The major advantage of the Entropy system concerns the cluster-wide context switch process itself. Entropy computes a new viable configuration and an optimized reconfiguration plan. This plan describes the sequences of transitions to perform (i.e. the run, migrate, suspend/resume, stop VM operations) in order to transit from the current situation to the new one. As the cost of each action and the dependencies between them is considered, Entropy reduces the duration of each cluster-wide context switch by performing a minimum number of actions in the most efficient way.

Around this solution, we developed VMScript, a domain-specific language for administration of virtualized grid infrastructures. This language relies on set manipulation and is used to introspect physical and virtual grid architectures, thanks to query expressions, and notably to modify VM placement on machines. VMScript interacts with Entropy and can be used to define administrative placement rules.

In 2011, Entropy has been integrated into a product of a newly founded start-up EasyVirt (see Sec. 7.1). Entropy has also been tested or used by our partners Orange Labs, DGFiP (direction Générale des Finances Publiques), Bull, MACIF, Logica.

Entropy is available under the LGPL license at http://entropy.gforge.inria.fr/.

5.4. FPath and FScript

Participants: Thomas Ledoux [correspondent], Frederico Alvares.

FPath and FScript are two domain-specific languages (DSLs) dealing respectively with the navigation and the dynamic reconfiguration of Fractal architectures. FPath is a DSL for querying Fractal architectures. It is restricted to the introspection of architectures by browsing elements identified by their properties or location in the architecture. This focused domain allows FPath to offer a very concise and readable syntax and ensures correctness properties by construction (e.g. any query terminates in a finite time). FScript is a DSL dedicated to the reconfiguration of Fractal component architectures. It enables reconfiguration scripts to modify a Fractal architecture. Like FPath, FScript guarantees several properties by construction, e.g. termination of scripts by excluding the possibility of infinite loops. Moreover the FScript interpreter supports a transactional model of reconfigurations and the preservation of ACID properties.

An adaptation of FPath/FScript to FraSCAti, a component framework providing runtime support for the Service Component Architecture (SCA), has been developed by the INRIA Adam project-team. In that way, software architects are able to navigate using FPath notation through FraSCAti architectures and to reconfigure them with FScript. We have used this adaptation in our recent work [22] for reconfiguring cloud applications in order to reduce the energy footprint in cloud infrastructures.

FScript and its extensions are available under the LGPL license at http://fractal.ow2.org/fscript.

5.5. WildCAT

Participants: Thomas Ledoux [correspondent], Frederico Alvares.
WildCAT is a generic Java framework for context-aware applications. It permits the monitoring of large-scale applications by allowing developers to easily organize and access resources through a hierarchical organization backed with a powerful SQL-like language to inspect sensors data and to trigger actions upon particular conditions. WildCAT proposes two modes to inspect the resources: a pull mode relies on synchronous communication and a push one relies on asynchronous communication. In the pull mode, developers programmatically get and set attributes. In the push mode, developers register listeners on queries expressed over the events generated by the backend.

WildCAT has been developed by the team in the last years. We have used WildCAT in our recent work [22] for allowing cloud applications to listen events notification fired by the cloud infrastructure (e.g. whenever the pricing policy of cloud resources changes) or to detect changes on the application activity (e.g. to detect whenever the number of requests/s sharply increases/decreases) in order to launch the reconfiguration of cloud applications.

WildCAT is available under GPL v2 at http://wildcat.ow2.org.
5. Software

5.1. The ATL Model Transformation Language

With an eye on the normative work of the OMG (MOF, OCL, QVT, etc.), a new conceptual framework has been developed based on a second generation model transformation language called ATL. Although ATL influenced the OMG standard, the approach is more general as discussed in [8]. In 2004 IBM gave an Eclipse innovation award to the ATL project. In 2007 Eclipse recognized ATL as one central solution for model transformation and promoted it to the M2M project (see Eclipse.org/m2m). There are more than 200 industrial and academic sites using ATL today, and several Ph.D. thesis in the world are based on this work.

In 2011 we started a new evolution phase for ATL. Our mid-term plan is making of ATL the leading solution for building autonomous reactive transformation systems, i.e. transformation networks that can autonomously manage a set of dataflows among the application models.

Following this line, we implemented in [43] a new refinement mode for ATL, to support in-place transformations. This extension allows the dynamic manipulation of models while keeping them connected to runtime applications.

In [31] we presented a lazy execution algorithm for ATL. With it, the elements of the target model are generated only when and if they are accessed. This extension allows to build reactive transformation systems that react to requests of model elements, by triggering the necessary computation. Our lazy version of ATL enable also transformations that generate infinite target models, extending the application space of the model-transformation paradigm.

The latest evolution of the ATL engine is a full reactive version, able to activate the minimal computation for responding to updates or request on the involved models. This engine is studied to scalably support large ATL networks. In this line we also introduced an algorithm for simplifying ATL transformation chains.

5.2. MoDisco (Model Discovery)

MoDisco is an open source Eclipse project that provides a generic and extensible framework dedicated to the elaboration of Model Driven Reverse Engineering (MDRE) solutions. Gathering contributions from both academics and industrials, the goal of the project is to federate common efforts in the model-based transformation of legacy software systems implemented using different technologies (e.g.: Java, COBOL, C). The first principle is to discover models out of legacy artifacts, representing appropriately all the relevant information, to be then used as part of reverse engineering processes for software understanding, evolution or modernization. Targeted scenarios include software (technical or architectural) migration of large legacy systems, but also retro-documentation, refactoring, quality assurance, etc. Within this context, MoDisco has collaborations with the OMG Architecture Driven Modernization (ADM) Task Force, for which the project provides several reference implementations of its standards: Knowledge Discovery Metamodel (KDM), Software Measurement Metamodel (SMM) and Abstract Syntax Tree Metamodel (ASTM).
The MoDisco framework is composed of a set of Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. Thanks to its modular architecture, it allows completely covering the three steps of a standard MDRE approach: 1) Discovery (i.e. extracting a complete model of the source code), 2) Understanding (i.e. browsing and providing views on this model for a given purpose) and 3) Transformation (evolving the model towards a new technology, architecture, etc). More specifically, as part of its Infrastructure layer, MoDisco offers the set of generic (i.e.; legacy technology-independent) reusable components really useful to build the core of MDRE solutions: Discovery Manager and Workflow for MDRE task orchestration, Model Browser for advanced navigation in complex models, model extension and customization capabilities for understanding (e.g.; views definition), etc. As part of its Technologies layer, it provides an advanced support for the Java, JEE and XML technologies, including complete metamodels, corresponding model discoverers, transformations, code generators, customizations, query libraries, etc.

MoDisco (or some of its components) is being used by different partners including other academics, industrials (e.g.; Sodifrance on several of their real modernization projects for their customers) or Eclipse projects (e.g.; Eclipse-MDT Papyrus as developed by CEA). Moreover, the Eclipse-EMFT EMF Facet project has been initiated as a MoDisco spin-off, in order to externalize some features which are not actually specific to reverse engineering problems and thus may be reused in many different contexts (cf. corresponding EMF Facet section).

5.3. EMFtoCSP

URL: http://code.google.com/a/eclipselabs.org/p/emftocsp/

EMFtoCSP is a tool for the verification of precisely defined conceptual models and metamodels. For these models, the definition of the general model structure (using UML or EMF) is supplemented by OCL constraints. The Eclipse Modeling Development Tools (MDT ²) provides mature tool support for such OCL-annotated models with respect to model definition, transformation, and validation.

However, an additional important task that is not supported by Eclipse MDT is the assurance of model quality. A systematical assessment of the correctness of such models is a key issue to ensure the quality of the final application. EMFtoCSP fills this gap by provided support for automated model verification in Eclipse.

Essentially, the EMFtoCSP is a sophisticated bounded model finder that yields instances of the model that conform not only to the structural definition of the model (e.g., the multiplicity constraints), but also to the OCL constraints. Based on this core, several correctness properties can be verified:

1. Satisfiability – is the model able to express our domain? For this check, the minimal number of instances and links can be specified to ensure non-trivial instances.
2. Unsatisfiability – is the model unable to express undesirable states? To verify this, we add further constraints to the model that state undesired conditions. Then we can check if is it impossible to instantiate the amended model.
3. Constraint subsumption – is one constraint already implied by others (and could therefore be removed)?
4. Constraint redundancy – do different constraints express the same fact (and could therefore be removed)?

To solve these search problems, EMFtoCSP translates the EMF/OCL (resp. UML/OCL) model into a problem of constraint logic programming and employs the Eclipse CLP solver ³ to solve it. This way, constraint propagation is exploited to tackle the (generally NP-hard) search.

² http://www.eclipse.org/modeling/mdt/?project=ocl
³ http://eclipseclp.org/
The tool is a continuation of the UMLtoCSP approach [52] developed previously by Jordi Cabot, Robert Clarisó and Daniel Riera. It provides a generic plugin framework for Eclipse to solve OCL-annotated models using constraint logic programming. Apart from already supported Ecore and UML metamodels, further metamodels can be added easily in the future. Similarly, other constraint solving back-ends can be integrated. It is provided under the Eclipse Public License.

5.4. AMW (AtlanMod Model Weaver)

URL: http://wiki.eclipse.org/AMW

AMW is an open source Eclipse project, under the Eclipse Public License (EPL), that provides a generic and extensible tooling dedicated to model weaving. It can be used to establish, represent and manage any kind of links (correspondences) between elements coming from different models (or metamodels). These links are stored in separated weaving models, allowing them to be exchanged and reused in various contexts using different modeling techniques. Thus, weaving models are actually relevant in many concrete application scenarios where several models are involved: (model driven) tool interoperability, mapping definition, transformation specification, traceability, model annotation, model merging, model comparison, etc.

The AMW tooling is composed of several Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. The provided model weaving workbench comes notably with a base weaving editor, reusable matching algorithms and weaving model serialization capabilities. These components are fully generic, so that they can be extended and adapted to any specific weaving metamodel (defining any kind of weaving links). AMW also offers a connector allowing to both use and produce weaving models as part of model-to-model transformations written in ATL.

AMW is being used by more than 40 user sites, including research labs and major companies (NASA, BAE, Versata, Mia-Software, Obeo, etc.). Currently part of the Eclipse-GMT project (to be terminated in the coming months), AMW is now in the process of being migrated into the Eclipse-M2M ATL project.

5.5. AM3 (AtlanMod MegaModel Management)

URL: http://wiki.eclipse.org/index.php/AM3

The AtlanMod Megamodel Management tool offers several functionalities for modeling in the large [3], i.e. for handling several related models (either terminal models, metamodels or transformation models) used as part of a complex modeling project.

The main component in AM3 is a generic megamodel manager that allows the user to browse and manipulate a set of related models. This manager knows the semantic relations between all these models. These relations are often associated to a given weaving model allowing not only navigating the traces between models, but also the traces between model elements. Since the links are stored externally as weaving models, the participating models do not get polluted and may be used as they are. Furthermore it is possible to handle multiple traceability chains going through similar models.

AM3 provides also a textual domain-specific language for model management called MoScript (URL: http://wiki.eclipse.org/MoScript). With MoScript, users can automate model management tasks by means of textual scripts written in an extension of the OCL language. For instance, user may write queries (based on model content, structure, relationships, and behaviour derived through on-the-fly simulation) to retrieve models from model repositories, manipulate them (e.g., by running transformations on sets of models), and store them back in the repository. MoScript also allows to populate and update the megamodel automatically by doing reverse engineer of simple modeling artifact repositories.

The generic tool for megamodel management has been used by different partners for several use cases like operationalization of chains of transformations.
5.6. Virtual EMF (Model Virtualization)

URL:  http://code.google.com/a/eclipselabs.org/p/virtual-emf/

Virtual EMF is an Eclipse plugin built on top of EMF that enables the creation and manipulation of virtual models, i.e., models whose elements do not contain concrete data, but are rather proxies to elements contained in other models. The idea is related to that of model composition, as it aims capturing the (often overlapping) concepts a set of models as one single global model. This is a frequently faced problem as, in complex scenarios, modelers often have to deal with a large number of heterogeneous and interrelated models, and most times the view a specific kind of user requires does not correspond to any of these models, but is a combination of cross-domain information scattered among several ones.

Current composition techniques rely on the materialization of the composed model, an approach that poses some important limitations in terms of (i) efficiency, as they do not scale (the data duplication mechanism they use implies in extra memory usage and time-consuming generation of the composed model), (ii) synchronicity, as updates in the composed model are not propagated to the original ones (or vice-versa), thus losing consistency, or even (iii) interoperability, as in some cases the composed model requires a specific API/tool to be handled.

Virtual EMF allows overcoming the limitations above. A virtual model provides to tools/user the illusion of working with a regular model whereas, in fact, all model access and manipulation requests are transparently redirected to its set of virtualized models. It serves as a centralized and transparent access point to a set of interconnected models, allowing users to easily compose, weave and link them, thus providing the following beneficial properties:

- **Interoperability:** it behaves as a normal model. Therefore, compatibility with existing EMF-based solutions/tools (e.g. models transformations, model editors, ...) is guaranteed;
- **Synchronization:** changes are automatically and transparently propagated between virtual and original models;
- **Scalability:** support for very big models;
  - low memory usage: no data duplication, direct access to original model elements;
  - faster generation time: no need for (time-consuming) information cloning operations (e.g. executing a model transformation);
- **Genericity:** support for several types of inter-model relationships (e.g. merge, association, filter) and extension capabilities for their semantics.

Virtual EMF is available as an open-source project on Eclipse Labs. It has been presented in a talk in EclipseCon Europe 2011 and contributed by the AtlanMod team to the CESAR project.

5.7. EMF Facet

URL:  http://www.eclipse.org/modeling/emft/facet/

EMF Facet is an open source Eclipse project, under the Eclipse Public License (EPL), that provides a generic and extensible framework dedicated to the dynamic and non-intrusive extension of models. It can be used to extend already existing metamodels with additional concepts and properties, the corresponding models being then transparently augmented, reduced or modified accordingly at runtime. Such a metamodel extension is called a facet, and can be specified on top of any metamodel in EMF Ecore. The underlying mechanism is based on the runtime execution of queries on the models corresponding to the faceted metamodels. Facets are notably particularly relevant for obtaining different views on existing models without having to actually alter them with any extra data.
The EMF Facet framework is composed of several Eclipse plugins, and relies on the de-facto standard Eclipse Modeling Framework (EMF) for model handling. The facet definitions are stored as facet models, allowing them to be exchanged and reused in various contexts. The queries can be implemented using any suitable query language (e.g.; ATL, OCL, Java, XPath), as far as the corresponding adaptors exist and are correctly registered within the framework. The proposed tooling includes dedicated editors for creating, editing and saving both facet and query definitions, the implemented support for Java, OCL and ATL queries, a Table Editor for visualizing query results. An advanced support for the model display customization (e.g.; icons, colors, fonts) is also provided as part of the framework.

EMF Facet is currently intensively used in MoDisco for extracting and displaying different specific views from large models of legacy systems. Its extension and customization capabilities are actually integrated into several MoDisco components, such as notably the MoDisco Model Browser. However, different other integration possibilities will be also explored in the future.

5.8. Portolan (Model-Driven Cartography)

URL: http://code.google.com/a/eclipselabs.org/p/portolan/

Processing large amounts of data to extract useful information is an essential task within companies. To help in this task, visualization techniques have been commonly used due to their capacity to present data in synthesized views, easier to understand and manage. However, achieving the right visualization display for a data set is a complex cartography process that involves several transformation steps to adapt the (domain) data to the (visualization) data format expected by visualization tools. With its simple core principles and set of base generic techniques (metamodeling, model transformation, model weaving, etc), Model-Driven Engineering (MDE) provides the relevant support for bridging the gap between data sets and visualization tools and thus for designing and implementing Cartography solutions. The proposed Portolan prototype is a concrete illustration of both a model-based and model-driven Cartography platform. Thus, the objective of Portolan is to facilitate the identification of interoperability solutions between tools by:

1. discovering (at least semi-automatically) maps of given situations in terms of deployed tools and relationships between them;
2. easily navigating and editing these maps;
3. augmenting or specializing them with both manually-entered and computed information;
4. visualizing them, using different customizable ways, in order to facilitate their understanding.

To this aim, the Portolan platform integrates:

- a set of default DSLs like GraphML, KML, Excel;
- visual displays based on Prefuse, Google Maps;
- modeling tools such as ATL language, Ecore modeler, etc.

It includes also an extension mechanism allowing the tool customization for advanced users. This recently developed generic tooling for cartography has already been used during the first action of our collaboration with BNP Paribas, as well as in the context of the IDM++ project.

5.9. The AmmA ToolBox

ATL, AMW, TCS, MoDisco, and AM3 are among the most important Eclipse.org components produced by the AtlanMod team. However there are also other components and a lot of functionalities, examples, and use cases made available and necessary to express solutions to many problems. The whole set of contributions composes the AmmA platform.
5.10. Industrialization strategy for research prototypes

Research labs, as priority innovation providers, are also indirectly key actors of the Software Engineering market. However, even if they already initiate the promotion of many new innovations to the industry, an important collaborative effort is still needed in order to actually transfer the corresponding techniques or technologies from the research lab to the company. Based on the AtlanMod concrete experience with the previously mentioned open source tools/projects, we have extracted a pragmatic approach [37] for transforming the results of scientific experimentation into practical industrial solutions.

While dealing with innovation, this approach is also innovation-driven itself, as the action is actually conducted by the research lab via a technology transfer. Three different partners are directly involved in this process, using open source as the medium for maintaining a constant interaction between all of them:

- **Use Case Provider.** Usually a company big enough to have to face real complex industrial scenarios which need to be solved (at least partially) by applying new innovative principles and techniques;
- **Research Lab.** Usually a group from a research institute (public or private) or university evaluating the scientific relevance of the problems, identifying the research challenges and prototyping possible solutions;
- **Technology Provider.** Usually a small or medium company, with a particular technical expertise on the given domain or Software Engineering field, building and delivering the industrial version of the designed solutions;

From our past and current experience, three main characteristics of this industrialization *business model* can be highlighted:

- **Win-win situation.** Each partner can actually focus on its core activity while also directly benefiting from the results obtained by the others (notably the research lab can continue to do research);
- **Application-driven context.** The end-user need is at the origin of the process, which finally makes the developed solution actually relevant;
- **Iterative process.** The fact of having three distinct partners requires different regular and consecutive exchanges between all of them.
5. Software

5.1. Hubble

Participants: Ludovic Courtès [correspondant], Nicolas Bonichon.

Hubble is implemented in Scheme, using GNU Guile version 2. Details of the simulation, such as keeping track of processor occupation and network usage, are taken care of by SimGrid, a toolkit for the simulation of distributed applications in heterogeneous distributed environments.

The input to Hubble is an XML description of the DAG of build tasks. For each task, a build duration and the size in bytes of the build output are specified. For our evaluation purposes, we collected this data on a production system, the http://hydra.nixos.org/ build farm hosted at the Technical University of Delft. The DAG itself is the snapshot of the Nix Package Collection (Nixpkgs) corresponding to this data. Hubble has its own in-memory representation of the DAG in the form of a purely functional data structure.

The Nixpkgs DAG contains fixed-output nodes, i.e., nodes whose output is known in advance and does not require any computation. These nodes are typically downloads of source code from external web sites. The raw data collected on http://hydra.nixos.org/ specifies a non-zero duration for these nodes, which represents the time it took to perform the download. This duration info is irrelevant in our context, since they don’t require any computation, and Hubble views these nodes as instantaneous.

See also the web page http://hubble.gforge.inria.fr/ .


5.2. namdP2P

Participants: Nicolas Bonichon [correspondant], Olivier Beaumont.

NamdP2P is a distributed implementation of ABF method using NAMD. It is worth noting that NAMD is designed to run on high-end parallel platforms or clusters, but not to run efficiently on instable and distributed platforms.


5.3. Malleable minWCT

Participants: Nicolas Bonichon [correspondant], Olivier Beaumont, Lionel Eyraud-Dubois.

This applet considers the scheduling of malleable tasks with bounded amount of processing resources. The goal is to compute schedules that minimize the weighted completion time of tasks. The applet generates all possible greedy schedules for a given instance and displays only the best ones.

This applet illustrates the complexity of finding an optimal order of tasks [31].

See also the web page http://www.labri.fr/perso/bonichon/malleable


5.4. Bedibe

Participants: Lionel Eyraud-Dubois [correspondant], Przemyslaw Uznanski.

Bedibe (Benchmarking Distributed Bandwidth Estimation) is a software to compare different models for bandwidth estimation on the Internet, and their associated instantiation algorithms. The goal is to ease the development of new models and algorithms, and the comparison with existing solutions. The development of this software is just starting.
See also the web page http://bedibe.gforge.inria.fr/.

5.5. MineWithRounds

Participants: Sofian Maabout [correspondant], Nicolas Hanusse.

This software extracts the Maximal Frequent Itemsets from a transaction data base. It is designed in C++ using OpenMP Library to take full advantage of multicore, multi-cpu machines.
5. Software

5.1. Intrusion Detection

Members of Supélec have developed several intrusion detectors.

Blare implements our approach of illegal information flow detection at the OS level. This implementation is a modification of a standard Linux kernel and it monitors information flows between typical OS containers as files, sockets or IPC. System active entities are processes viewed as black-boxes as we only observe their inputs and outputs. Detection at the OS level is in some cases too coarse-grained to avoid the generation of false positives and to detect attacks targeting the application logic. Even if it remains convenient to define the security policy at the OS-level, sound illegal information flow detection implies an additional detection at the language level. This has led us to implement a detector for Java applications, JBlare, to complement the detection at the OS level. JBlare extends the OS-level one by refining the observation of information flows at the language level.

GNG is an intrusion detection system that correlates different sources (such as different logs) in order to identify attacks against the system. The attack scenarios are defined using the Attack Description Langage (ADeLe) proposed by our team, and are internally translated to attack recognition automats. GNG intends to define time efficient algorithms based on these automatons to recognize complex attack scenarios.

SIDAN (Software Instrumentation for Detecting Attacks on Non-control-data) is a tool that aims to instrument automatically C-language software with assertions whose role is to detect attacks against the software. This tool is implemented as a plugin of the FRAMA-C framework that provides an implementation of static analysis techniques.

5.2. Privacy

GEPETO (GEoPrivacy-Enhancing TOolkit) is an open source software for managing geolocated data (currently in development in cooperation with LAAS). GEPETO can be used to visualize, sanitize, perform inference attacks and measure the utility of a particular geolocated dataset. For each of these actions, a set of different techniques and algorithms can be applied. The global objective of GEPETO is to enable a user to design, tune, experiment, and evaluate various sanitization algorithms and inference attacks as well as visualizing the following results and assessing their utility.

5.3. Reliable Programming

The Prometeus project, part of the Inria Gforge, is a software environment for reliable programming. The basic elements of Prometeus are Eva, a component-based framework and Adam, a set of group communication services. Eva is an implementation of a component model that aims at supporting the development of distributed abstractions and high-level communication protocols. Adam is a library of agreement components, based on the component model implemented by Eva. The central element of the Adam library is GAC (Generic Agreement Component). It implements a generic and adaptive fault-tolerant consensus algorithm that can be customized to cope with the characteristics of the environment. Moreover, thanks to a set of versatile methods, its behavior can be tuned to fit the exact needs of a specific agreement problem. A range of fundamental Adam components are implemented as specializations of this GAC component. The Adam library currently includes the most important components for reliable distributed programming (Group Membership, Atomic Broadcast).
5. Software

5.1. T3devKit testing toolkit and IPv6 test suites

Participants: Anthony Baire, César Viho.

We have built a toolkit for easing executing tests written in the standardized TTCN-3 test specification language. This toolkit is made of a C++ library together with a highly customizable CoDec generator that allows fast development of external components required to execute a test suite. It also provides a framework for representing and manipulating TTCN-3 events so as to ease the production of test reports. The toolkit addresses issues that are not yet covered by ETSI standards while being fully compatible with the existing standard interfaces: TRI (Test Runtime Interfaces) and TCI (Test Control Interfaces). It has been tested with four TTCN-3 environments (IBM, Elvior, Danet and Go4IT) and on three different platforms (Linux, Windows and Cygwin). It is publicly released under the CeCILL-C License. All these tools with associated test suites (for RIPng, DHCPv6 and examples for DNS) are freely available at http://www.irisa.fr/tipi.

5.2. Interoperability Assessment

Participants: Anthony Baire, Nanxing Chen, César Viho.

In previous works we have developed a software toolkit named T3DevKit for easing the development of TTCN-3 tests. The original tool could only run on POSIX systems with the gcc tool-chain. This year we re-factored the build system using the waf build automation tool, which allowed us to integrate with other operating systems and tool-chains (especially MSVC on Windows). This work allowed us to study interoperability issues between tools in the TTCN-3 standard and a poster was presented at the TTCN-3 User Conference 2011. We also presented an introduction tutorial for T3DevKit at this conference [76]. A method to generate and to execute passive interoperability test suites on recorded traces has been proposed in [40].

5.3. Performance and dependability evaluation

Participants: Gerardo Rubino, Bruno Sericola, Bruno Tuffin.

We develop software tools for the evaluation of two classes of models: Markov models and reliability networks. The main objective is to quantify dependability aspects of the behaviors of the modeled systems, but other aspects of the systems can be handled (performance, performability, vulnerability). The tools are specialized libraries implementing numerical, Monte Carlo and Quasi-Monte Carlo algorithms.

One of these libraries has been developed for the Celar (DGA), and its goal is the evaluation of dependability and vulnerability metrics of wide area communication networks (WANs). The algorithms in this library can also evaluate the sensitivities of the implemented dependability measures with respect to the parameters characterizing the behavior of the components of the networks (nodes, lines).

We are also developing tools with the objective of building Markovian models and to compute bounds of asymptotic metrics such as the asymptotic availability of standard metrics of models in equilibrium, loss probabilities, blocking probabilities, mean backlogs,....). A set of functions designed for dependability analysis is being built under the name DependLib.
5. Software

5.1. SOFAT

Participants: Loïc Hélouët [correspondant], Rouwaida Abdallah.

SOFAT is the acronym for Scenario Oracle and Formal Analysis Toolbox. As this name suggests it is a formal analysis toolbox for scenarios. Scenarios are informal descriptions of behaviors of distributed systems. SOFAT allows the edition and analysis of distributed systems specifications described using Message Sequence Charts, a scenario language standardized by the ITU [Z.120]. The main functionalities proposed by SOFAT are the textual edition of Message Sequence Charts, their graphical visualization, the analysis of their formal properties, and their simulation. The analysis of the formal properties of a Message Sequence Chart specification determines if a description is regular, local choice, or globally cooperative. Satisfaction of these properties allow respectively for model-checking of logical formulae in temporal logic, implementation, or comparison of specifications. All these applications are either undecidable problems or unfeasible if the Message Sequence Chart description does not satisfy the corresponding property. The SOFAT toolbox implements most of the theoretical results obtained on Message Sequence Charts this last decade. It is regularly updated and re-distributed. The purpose of this software is twofold:

- Provide a scenario based specification tool for developers of distributed applications
- Serve as a platform for theoretical results on scenarios and partial orders

SOFAT provides several functionalities, that are: syntactical analysis of scenario descriptions, Formal analysis of scenario properties, Interactive Simulation of scenarios when possible, and diagnosis. This year, SOFAT was extended with code synthesis functionalities, allowing to generate communicating automata, promela code, or rest based web services from HMSCs. A new release of the software is expected before the end of the year.

See also the web page http://www.irisa.fr/distribcom/Prototypes/SOFAT/index.html.

- AMS: Order; lattices; ordered algebraic structures
- APP: IDDN.FR.001.080027.000.S.P.2003.00.10600
- Programming language: Java
5. Software

5.1. WSNet

Participant: Guillaume Chelius [correspondant].

WSNet is a wireless sensor network simulator that was designed to offer the following features:

- a modular, flexible and accurate simulation of the radio physical medium;
- support for the simulation of environmental phenomena;
- support for interaction between nodes and their environment (sensor-actuator architecture);
- interconnection with the sensor platform emulator WSim to support the distributed emulation of wireless sensor networks.

WSNet is currently in its second release. The number of WSNet users is still growing and several research works reference the software. Many pointers can be found on the project website. Maintenance and support of the software is handled by the D-NET project but also by several contributors from the CITI laboratory (INSA de Lyon), Orange R&D. The WSNet community is quietly spreading in France as well as abroad.

5.2. WSNet-3

Participant: Guillaume Chelius [correspondant].

Driven by the feedback gathered among WSNet users, we have started the development of the third WSNet release. While still private, the project web page is available. The objectives behind this new development is:

- to ease the simulation of new radio architectures / standards: e.g. MIMO schemes, UWB, multi-interfaces system;
- to ease the writing of new modules through the use of High Level Languages such as Python or Ruby for the development of protocols, etc;
- to ease the debugging and compilation of results during a simulation.

These developments are handled by a core of developers from different affiliations (INSA de Lyon, Orange R&D, INRIA) lead by the D-NET team.

5.3. Sensor Network Tools: drivers, OS and more

Participants: Guillaume Chelius, Eric Fleury [correspondant], Clément Burin des Rosiers, Sandrine Avakian, Guillaume Roche.

As a outcomes of the ANR SensLAB project and the INRIA ADT SensTOOLS and SensAS, several softwares (from low level drivers to OSes) were delivered and made available to the research community. The main goal is to lower the cost of developing/deploying a large scale wireless sensor network application. All software are gathered under the SensLAB web site: http://www.senslab.info/ web page where one can find:

- low C-level drivers to all hardware components;
- ports of the main OS, mainly TinyOS, FreeRTOS and Contiki;
- ports and development of higher level library like routing, localization.
5. Software

5.1. Jolie

Members of Focus have recently developed Jolie [7] (Java Orchestration Language Interpreter Engine, see http://www.jolie-lang.org/). Jolie is a service-oriented programming language. Jolie can be used to program services that interact over the Internet using different communication protocols. Differently from other Web Services programming languages such as WS-BPEL, Jolie is based on a user-friendly C/Java-like syntax (more readable than the verbose XML syntax of WS-BPEL) and, moreover, the language is equipped with a formal operational semantics. This language is used for the proof of concepts developed around Focus activities. For instance, contract theories can be exploited for checking the conformance of a Jolie program with respect to a given contract. A spin-off, called “Italiana Software”, has been launched around Jolie, its general aim is to transfer the expertise in formal methods for Web Services matured in the last few years onto Service Oriented Business Applications. The spin-off is a software producer and consulting company that offers service-oriented solutions (for instance, a “single sign-on” application) based on the Jolie language.

In 2011 the development of Jolie has continued. The main activities have been:

- A new session message-routing mechanism, based on correlation sets has been implemented. This mechanism makes message routing programmable from inside Jolie code.
- A new primitive for (smart) service aggregation.
- A graphical editor.
- An integrated development environment.

Moreover, this year Jolie has been used for teaching, in a 30-hour master course at IT University of Copenhagen, Denmark.

5.2. Others

Below we list some software that has been developed in Focus in previous years but that during 2011 has not substantially changed. Short descriptions of these items can be found in the Focus activity report for 2010.

- PiDuce (see http://www.cs.unibo.it/PiDuce/ ) is a prototype for experimenting Web services technologies, based on theories of process calculi and XML documents and schemas [3].
- IntML is a functional programming language guaranteeing sublinear space bounds for all programs [57].
GANG Project-Team (section vide)
5. Software

5.1. DIET

Participants: Yves Caniou, Eddy Caron [correspondent], Frédéric Desprez, Maurice Djibril Faye, Adrian Muresan, Jonathan Rouzaud-Cornabas.

Huge problems can now be processed over the Internet thanks to Grid and Cloud middleware systems. The use of on-the-shelf applications is needed by scientists of other disciplines. Moreover, the computational power and memory needs of such applications may of course not be met by every workstation. Thus, the RPC paradigm seems to be a good candidate to build Problem Solving Environments on the Grid or Cloud. The aim of the DIET project (http://graal.ens-lyon.fr/DIET) is to develop a set of tools to build computational servers accessible through a GridRPC API.

Moreover, the aim of a middleware system such as DIET is to provide a transparent access to a pool of computational servers. DIET focuses on offering such a service at a very large scale. A client which has a problem to solve should be able to obtain a reference to the server that is best suited for it. DIET is designed to take into account the data location when scheduling jobs. Data are kept as long as possible on (or near to) the computational servers in order to minimize transfer times. This kind of optimization is mandatory when performing job scheduling on a wide-area network. DIET is built upon Server Daemons. The scheduler is scattered across a hierarchy of Local Agents and Master Agents. Applications targeted for the DIET platform are now able to exert a degree of control over the scheduling subsystem via plug-in schedulers [85]. As the applications that are to be deployed on the Grid vary greatly in terms of performance demands, the DIET plug-in scheduler facility permits the application designer to express application needs and features in order that they be taken into account when application tasks are scheduled. These features are invoked at runtime after a user has submitted a service request to the MA, which broadcasts the request to its agent hierarchy.

DIET has been validated on several applications. Example of them have been described in Sections 4.3 through 4.5.

5.1.1. DIET Security

We have worked on extending DIET to include security mechanisms. The first work was to provide authentication of users and components within DIET without breaking DIET distributed architecture. Our security mechanism must also be simple to use by the end users but we need a strong authentication. Recently, we have opted for Kerberos as it provided a Single Sign One that eases the security from the user point of view. Moreover, Kerberos provides strong authentication and works with heterogeneous systems. Work in progress is to integrate Kerberos within DIET. First, it will be used to provide traceability of user’s actions and authentication of all DIET inner components. Then, it will be integrated in an authorization mechanism and other higher level security mechanisms.

5.1.2. GridRPC Data Management API

The GridRPC paradigm is an OGF standard, but the API appeared to lack of precision in order to make a GridRPC code portable to any GridRPC compliant middleware. Additionally required data have to be present on the client side (this can involve a potential transfer from where the data is stored onto the client), and transfers must be performed during the GridRPC call, both degrading performance, and can even make a calculus unfeasible.

Thus the GridRPC community has interests in Data Management within the GridRPC paradigm – Because of previous works performed in the DIET middleware concerning Data Management, Eddy Caron is co-chair of the GridRPC working group.
In consequence, we worked on a Data Management API which has been presented to almost all OGF sessions since OGF’21. Since september 2011, the proposal is an OGF standard, published at http://www.ogf.org/documents/GFD.186.pdf under the title “Data Management API within the GridRPC. Y. Caniou and others, via GRIDRPC-WG”. Some work are still in progress, like 1) the implementation of a library and its integration into GridRPC middleware, in order to publish a proof of concept of both realization and collaboration between two different GridRPC middleware supervising different domain platforms, and 2) a specific OGF document describing some parts of implementation to achieve code portability.

5.1.3. Latest Releases

- November 14th 2011, DIET 2.8 release.
- June 16th 2011, DIET 2.7 release.
- March 7th 2011, DIET 2.6.1 release
- January 14th 2011, DIET 2.6 release

5.2. MUMPS

Participants: Maurice Brémond, Guillaume Joslin, Jean-Yves L’Excellent [correspondent], Mohamed Sid-Lakhdar, Bora Uçar.

MUMPS (for MULTifrontal Massively Parallel Solver, see http://graal.ens-lyon.fr/MUMPS ) is a software package for the solution of large sparse systems of linear equations. The development of MUMPS was initiated by the European project PARASOL (Esprit 4, LTR project 20160, 1996-1999), whose results and developments were public domain. Since then, research and developments have been supported by CERFACS, CNRS, ENS Lyon, INPT-ENSEEIHT-IRIT (main contributor), INRIA, and University of Bordeaux.

MUMPS implements a direct method, the multifrontal method, and is a parallel code capable of exploiting distributed-memory computers; its main originalities are its performance, its numerical robustness and the wide range of functionalities available.

The latest release is MUMPS 4.10.0 (May 2011). Its main new functionalities concern the determinant, the possibility to compute entries of the inverse of a sparse matrix and an option to discard factors. Some memory and performance improvements have also been obtained thanks to specific users’ testcases. This year, we have also worked on generic tools and scripts for experimentation, validation and performance study.


5.3. HLCMi

Participants: Julien Bigot, Cristian Klein, Christian Pérez [correspondent], Vincent Pichon.

HLCMi is an implementation of the HLCM component model defined during the PhD of Julien Bigot. HLCM is a generic extensible component model with respect to component implementations and interaction concerns. Moreover, HLCM is abstract; it is its specialization—such as HLCM/CCM—that define the primitive elements of the model, such as the primitive components and the primitive interactions.

HLCMi is making use of Model-driven Engineering (MDE) methodology to generate a concrete assembly from an high level description. It is based on the Eclipse Modeling Framework (EMF). HLCMi contains 700 Emfatic lines to describe its models and 7000 JAVA lines for utility and model transformation purposes. HLCMi is a general framework that supports several HLCM specialization: HLCM/CCM, HLCM/JAVA, HLCM/C++ (known as L2C) and HLCM/Charm++ (known as Gluon++).

5.4. BitDew

Participants: Gilles Fedak [correspondent], Haiwu He, Bing Tang, José Francisco Saray Villamizar, Mircea Moca, Lu Lu.
BitDEW is an open source middleware implementing a set of distributed services for large scale data management on Desktop Grids and Clouds. BitDEW relies on five abstractions to manage the data: i) replication indicates how many occurrences of a data should be available at the same time on the network, ii) fault-tolerance controls the policy in presence of hardware failures, iii) lifetime is an attribute absolute or relative to the existence of other data, which decides the life cycle of a data in the system, iv) affinity drives movement of data according to dependency rules, v) protocol gives the runtime environment hints about the protocol to distribute the data (http, ftp or bittorrent). Programmers define for every data these simple criteria, and let the BitDEW runtime environment manage operations of data creation, deletion, movement, replication, and fault-tolerance operation.

The current status of the software is the following: BitDEW is open source under the GPLv3 or Cecill licence at the user’s choice, 10 releases were produced in the last two years, and it has been downloaded approximatively 6000 times on the INRIA forge. Known users are Université Paris XI, Université Paris-XIII, University of Florida, Cardiff University and University of Sfax. In term of support, the development of BitDew is partly funded by the INRIA ADT BitDew and by the ANR MapReduce projects. Thanks to this support, we have developed and released the first prototype of the MapReduce programming model for Desktop Grids on top of BitDew. In 2011, 5 versions of the software have been released, including the version 1.0.0 considered as the first stable release of BitDew. Our most current work focuses on providing reliable storage on top of hybrid distributed computing infrastructures.

5.5. XtremWeb

Participants: Gilles Fedak [correspondent], Haiwu He, Bing Tang, Simon Delamare.

XtremWeb is an open source software for Desktop Grid computing, jointly developed by INRIA and IN2P3. XtremWeb allows to build lightweight Desktop Grid by gathering the unused resources of Desktop Computers (CPU, storage, network). Its primary features permit multi-users, multi-applications and cross-domains deployments. XtremWeb turns a set of volatile resources spread over LAN or Internet into a runtime environment executing high throughput applications.

XtremWeb is a highly programmable and customizable middleware which supports a wide range of applications (bag-of tasks, master/worker), computing requirements (data/CPU/network-intensive) and computing infrastructures (clusters, Desktop PCs, multi-Lan) in a manageable, scalable and secure fasion. Known users include LIFL, LIP, LIG, LRI (CS), LAL (physics Orsay), IBBMC (biology), Université Paris-XIII, Université de Guadeloupe, IPF (petroleum), EADS, CEA, University of Wisconsin Madison, University of Tsukuba (Japan), AIST (Australia), UCSD (USA), Université de Tunis, AlmerGrid (NL), Fundecyt (Spain), Hobai (China), HUST (China).

There are two branches of XtremWeb: XtremWeb-HEP is a production version developed by IN2P3. It features many security improvements such as X509 support which allows its usage within the EGEE context. XtremWeb-CH is a research version developed by HES-SO, Geneva, which aims at building an effective Peer-To-Peer system for CPU time consuming applications.

XtremWeb has been supported by national grants (ACI CGP2P) and by major European grants around Grid and Desktop Grid such as FP6 CoreGrid: European Network of Excellence, FP6 Grid4All, and more recently FP7 EDGeS : Enabling Desktop Grid for E-Science and FP7 EDGI: European Desktop Grid Initiative.

On going developments include: providing Quality-of-Service for Desktop Grids (SpeQuIoS), inclusion of the BitDew middleware to distribute data as well as inclusion of virtualization tehnologies.
5. Software

5.1. APMC-CA

Participants: Sylvain Peyronnet [correspondant], Joel Falcou, Pierre Esterie, Khaled Hamidouche, Alexandre Borghi.

The APMC model checker implements the state-of-the-art approximate probabilistic model checking methods. Last year we develop a version of the tool dedicated to the CELL architecture. Clearly, it was very pedagogic, but the conclusion is that the CELL is not adapted to sampling based verification methods.

This year we develop, thanks to the BSP++ framework, a version compatible with SPM/multicores machines, clusters and hybrid architectures. This version outperforms all previous ones, thus showing the interest of both these new architectures and of the BSP++ framework.

5.2. YML

Participants: Serge Petiton [correspondant], Nahid Emad, Maxime Hugues.

Scientific end-users face difficulties to program P2P large scale applications using low level languages and middleware. We provide a high level language and a set of tools designed to develop and execute large coarse grain applications on peer-to-peer systems. Thus, we introduced, developed and experimented the YML for parallel programming on P2P architectures. This work was done in collaboration with the PRiSM laboratory (team of Nahid Emad).

The main contribution of YML is its high level language for scientific end-users to develop parallel programs for P2P platforms. This language integrates two different aspects. The first aspect is a component description language. The second aspect allows to link components together. A coordination language called YvetteML can express graphs of components which represent applications for peer-to-peer systems.

Moreover, we designed a framework to take advantage of the YML language. It is based on two component catalogues and an YML engine. This separation enhances portability of applications and permits real time optimizations. Currently we provide support for the XtremWeb Peer-to-Peer middleware and the OmniRPC grid system. The support for Condor is currently under development and a beta-release will be delivered soon (in this release, we plan to propagate semantic data from the end-users to the middleware). The next development of YML concerns the implementation of a multi-backend scheduler. Therefore, YML will be able to schedule at runtime computing tasks to any global computing platform using any of the targeted middleware.

We experimented YML with basic linear algebra methods on a XtremWeb P2P platform deployed between France and Japan. Recently, we have implemented complex iterative restarted Krylov methods, such as Lanczos-Bisection, GMRES and MERAM methods, using YML with the OmniRPC back-end. The experiments are performed either on the Grid5000 testbed of on a Network of Workstations deployed between Lille, Versailles and Tsukuba in Japan. Demos was proposed on these testbeds from conferences in USA. We recently finished evaluations of the overhead generated using YML, without smart schedulers and with extrapolations due to the lack of smart scheduling strategies inside targeted middleware.

In the context of the FP3C project funded by ANR-JST, we have recently extended YML to support a directive distributed parallel language, XcalableMP http://www.xcalablemp.org/ . This extension is based on the support of the XcalableMP language inside YML components. This allows to develop parallel programs with two programming paradigm and thus two parallelism levels. This work is a part of the project that targets post-Petascale supercomputer that would be composed of heterogeneous and massively parallel hardware.

The software is available at http://yml.prism.uvsq.fr/
5.3. The Scientific Programming InterNet (SPIN)

Participant: Serge Petiton [correspondant].

SPIN (Scientific Programming on the InterNet), is a scalable, integrated and interactive set of tools for scientific computations on distributed and heterogeneous environments. These tools create a collaborative environment allowing the access to remote resources.

The goal of SPIN is to provide the following advantages: Platform independence, Flexible parameterization, Incremental capacity growth, Portability and interoperability, and Web integration. The need to develop a tool such as SPIN was recognized by the GRID community of the researchers in scientific domains, such as linear algebra. Since the P2P arrives as a new programming paradigm, the end-users need to have such tools. It becomes a real need for the scientific community to make possible the development of scientific applications assembling basic components hiding the architecture and the middleware. Another use of SPIN consists in allowing to build an application from predefined components ("building blocks") existing in the system or developed by the developer. The SPIN users community can collaborate in order to make more and more predefined components available to be shared via the Internet in order to develop new more specialized components or new applications combining existing and new components thanks to the SPIN user interface.

SPIN was launched at ASCI CNRS lab in 1998 and is now developed in collaboration with the University of Versailles, PRiSM lab. SPIN is currently under adaptation to incorporate YML, cf. above. Nevertheless, we study another solution based on the Linear Algebra KErnel (LAKE), developed by the Nahid Emad team at the University of Versailles, which would be an alternative to SPIN as a component oriented integration with YML.

5.4. V-DS

Participant: Franck Cappello [correspondant].

This project started officially in September 2004, under the name V-Grid. V-DS stands for Virtualization environment for large-scale Distributed Systems. It is a virtualization software for large scale distributed system emulation. This software allows folding a distributed systems 100 or 1000 times larger than the experimental testbed. V-DS virtualizes distributed systems nodes on PC clusters, providing every virtual node its proper and confined operating system and execution environment. Thus compared to large scale distributed system simulators or emulators (like MicroGrid), V-DS virtualizes and schedules a full software environment for every distributed system node. V-DS research concerns emulation realism and performance.

A first work concerns the definition and implementation of metrics and methodologies to compare the merits of distributed system virtualization tools. Since there is no previous work in this domain, it is important to define what and how to measure in order to qualify a virtualization system relatively to realism and performance. We defined a set of metrics and methodologies in order to evaluate and compared virtualization tools for sequential system. For example a key parameter for the realism is the event timing: in the emulated environment, events should occur with a time consistent with a real environment. An example of key parameter for the performance is the linearity. The performance degradation for every virtual machine should evolve linearly with the increase of the number of virtual machines. We conducted a large set of experiments, comparing several virtualization tools including Vserver, VMware, User Mode Linux, Xen, etc. The result demonstrates that none of them provides both enough isolation and performance. As a consequence, we are currently studying approaches to cope with these limits.

We have made a virtual platform on the GDX cluster with the Vserver virtualization tool. On this platform, we have launched more than 20K virtual machines (VM) with a folding of 100 (100 VM on each physical machine). However, some recent experiments have shown that a too high folding factor may cause a too long execution time because of some problems like swapping. Currently, we are conducting experiments on another platform based on the virtualization tool named Xen which has been strongly improved since 2 years. We expect to get better result with Xen than with Vserver. Recently, we have been using the V-DS version based on Xen to evaluate at large scales three P2P middleware [89].
5.5. PVC: Private Virtual Cluster

Participant: Franck Cappello [correspondant].

Current complexity of Grid technologies, the lack of security of Peer-to-Peer systems and the rigidity of VPN technologies make sharing resources belonging to different institutions still technically difficult.

We propose a new approach called "Instant Grid" (IG), which combines various Grid, P2P and VPN approaches, allowing simple deployment of applications over different administration domains. Three main requirements should be fulfilled to make Instant Grids realistic: simple networking configuration (Firewall and NAT), no degradation of resource security, no need to re-implement existing distributed applications.

Private Virtual Cluster, is a low-level middle-ware that meets Instant Grid requirements. PVC turns dynamically a set of resources belonging to different administration domains into a virtual cluster where existing cluster runtime environments and applications can be run. The major objective of PVC is to establish direct connections between distributed peers. To connect firewall protected nodes in the current implementation, we have integrated three techniques: UPnP, TCP/UDP Hole Punching and a novel technique Traversing-TCP.

One of the major application of PVC is the third generation desktop Grid middleware. Unlike BOINC and XtremWeb (which belong to the second generation of desktop Grid middleware), PVC allows the users to build their Desktop Grid environment and run their favorite batch scheduler, distributed file system, resource monitoring and parallel programming library and runtime software. PVC ensures the connectivity layer and provide a virtual IP network where the user can install and run existing cluster software.

By offering only the connectivity layer, PVC allows to deploy P2P systems with specific applications, like file sharing, distributed computing, distributed storage and archive, video broadcasting, etc.

5.6. OpenWP

Participant: Franck Cappello [correspondant].

Distributed applications can be programmed on the Grid using workflow languages, object oriented approaches (Proactive, IBIS, etc), RPC programming environments (Grid-RPC, DIET), component based environments (generally based on Corba) and parallel programming libraries like MPI.

For high performance computing applications, most of the existing codes are programmed in C, Fortran and Java. These codes have 100,000 to millions of lines. Programmers are not inclined to rewrite then in a "non standard" programming language, like UPC, CoArray Fortran or Global Array. Thus environments like MPI and OpenMPI remain popular even if they require hybrid approaches for programming hierarchical computing infrastructures like cluster of multi-processors equipped with multi-core processors.

Programming applications on the Grid add a novel level in the hierarchy by clustering the cluster of multi-processors. The programmer will face strong difficulties in adapting or programming a new application for these runtime infrastructures featuring a deep hierarchy. Directive based parallel and distributed computing is appealing to reduce the programming difficulty by allowing incremental parallelization and distribution. The programmer add directives on a sequential or parallel code and may check for every inserted directive its correction and performance improvement.

We believe that directive based parallel and distributed computing may play a significant role in the next years for programming High performance parallel computers and Grids. We have started the development of OpenWP. OpenWP is a directive based programming environment and runtime allowing expressing workflows to be executed on Grids. OpenWP is compliant with OpenMP and can be used in conjunction with OpenMP or hybrid parallel programs using MPI + OpenMP.
The OpenWP environment consists in a source to source compiler and a runtime. The OpenWP parser, interprets the user directives and extracts functional blocks from the code. These blocks are inserted in a library distributed on all computing nodes. In the original program, the functional blocks are replaced by RPC calls and calls to synchronization. During the execution, the main program launches non blocking RPC calls to functions on remote nodes and synchronize the execution of remote functions based on the synchronization directives inserted by the programmer in the main code. Compared to OpenMP, OpenWP does not consider a shared memory programming approach. Instead, the source to source compiler insert data movements calls in the main code. Since the data set can be large in Grid application, the OpenWP runtime organize the storage of data sets in a distributed way. Moreover, the parameters and results of RPC calls are passed by reference, using a DHT. Thus, during the execution, parameter and result references are stored in the DHT along with the current position of the datasets. When a remote function is called, the DHT is consulted to obtain the position of the parameter data sets in the system. When a remote function terminates its execution, it stores the result data sets and store a reference to the data set in the DHT.

We are evaluating OpenWP from an industrial application (Amibe), used by the European aerospace company EADS. Amibe is the mesher module of jCAE 1. Amibe generates a mesh from a CAD geometry in three steps. It first creates edges between every patch of the CAD (mesh in one dimension), then generates a surface mesh for every unfolded patch (mesh in two dimensions) and finally adds the third dimension to the mesh by projecting the 2D mesh into the original CAD surfaces. The first and third operation cannot be distributed. However the second step can easily be distributed following a master/worker approach, transferring the mesh1d results to every computing node and launching the distributed execution of the patches.

5.7. Parallel solvers for solving linear systems of equations

**Participant:** Laura Grigori.

In the last several years, there has been significant research effort in the development of fully parallel direct solvers for computing the solution of large unsymmetric sparse linear systems of equations. In this context, we have designed and implemented a parallel symbolic factorization algorithm, which is suitable for general sparse unsymmetric matrices. The symbolic factorization is one of the steps that is sequential and represents a memory bottleneck. The code is intended to be used with very large matrices when because of the memory usage, the sequential algorithm is not suitable. This code is available in the SuperLU_DIST, a widely used software, developed at UC Berkeley and LBNL by Professor James W. Demmel and Dr. Xiaoye S. Li. The algorithm is presented in [77]. The SuperLU_DIST is available at http://crd.lbl.gov/~xiaoye/SuperLU/.

5.8. OpenScop

**Participant:** Cédric Bastoul.

OpenScop is an open specification which defines a file format and a set of data structures to represent a static control part (SCoP for short), i.e., a program part that can be represented in the polyhedral model, an algebraic representation of programs used for automatic parallelization and optimization (used, e.g., in GNU GCC, LLVM, IBM XL or Reservoir Labs R-Stream compilers). The goal of OpenScop is to provide a common interface to various polyhedral compilation tools in order to simplify their interaction.

OpenScop provides a single format for tools that may have different purposes (e.g., as different as code generation and data dependence analysis). We could observe that most available polyhedral compilation tools during the last decade were manipulating the same kind of data (polyhedra, affine functions...) and were actually sharing a part of their input (e.g., iteration domains and context concepts are nearly everywhere). We could also observe that those tools may rely on different internal representations, mostly based on one of the major polyhedral libraries (e.g., Polylib, PPL or isl), and this representation may change over time (e.g., when switching to a more convenient polyhedral library). OpenScop aims at providing a stable, unified format that offers a durable guarantee that a tool can use an output or provide an input to another tool without breaking a compilation chain because of some internal changes in one element of this chain. The other promise of

---

1 project page: http://jcae.sourceforge.net
OpenScop is the ability to assemble or replace the basic blocks of a polyhedral compilation framework at no, or at least low engineering cost. The OpenScop Library (licensed under the 3-clause BSD license) has been developed as an example, yet powerful, implementation of the OpenScop specification.

5.9. CALU for multicore architectures

**Participant:** Laura GRIGORI [correspondant].

The communication avoiding algorithms are implemented in the form of a portable library. In its current form, this library is designed for multicore architectures and uses a hybrid scheduling technique that exploits well the data locality and can adapt to dynamic changes in the machine. The library will be publicly available since February 2012.


- Version: 1.0

5.10. Fast linear system solvers in public domain libraries

**Participant:** Marc Baboulin [correspondant].

Hybrid multicore+GPU architectures are becoming commonly used systems in high performance computing simulations. In this research, we develop linear algebra solvers where we split the computation over multicore and graphics processors, and use particular techniques to reduce the amount of pivoting and communication between the hybrid components. This results in efficient algorithms that take advantage of each computational unit [12]. Our research in randomized algorithms yields to several contributions to propose public domain libraries PLASMA and MAGMA in the area of fast linear system solvers for general and symmetric indefinite systems. These solvers minimize communication by removing the overhead due to pivoting in LU and LDLT factorization.

See also the web page [http://icl.cs.utk.edu/magma/](http://icl.cs.utk.edu/magma/).

5.11. cTuning: Repository and Tools for Collective Characterization and Optimization of Computing Systems

**Participant:** Grigori Fursin [correspondant].

Designing, porting and optimizing applications for rapidly evolving computing systems is often complex, ad-hoc, repetitive, costly and error prone process due to an enormous number of available design and optimization choices combined with the complex interactions between all components. We attempt to solve this fundamental problem based on collective participation of users combined with empirical tuning and machine learning.

We developed cTuning framework that allows to continuously collect various knowledge about application characterization and optimization in the public repository at cTuning.org. With continuously increasing and systematized knowledge about behavior of computer systems, users should be able to obtain scientifically motivated advices about anomalies in the behavior of their applications and possible solutions to effectively balance performance and power consumption or other important characteristics.

Currently, we use cTuning repository to analyze and learn profitable optimizations for various programs, datasets and architectures using machine learning enabled compiler (MILEPOST GCC). Using collected knowledge, we can quickly suggest better optimizations for a previously unseen programs based on their semantic or dynamic features [7].

We believe that such approach will be vital for developing efficient Exascale computing systems. We are currently developing the new extensible cTuning2 framework for automatic performance and power tuning of HPC applications.

For more information, see the web page [http://cTuning.org](http://cTuning.org).
5. Software

5.1. Introduction

We describe in this section the software that we are developing. The first two (MaPHyS and EPSN) will be the main milestones of our project. The other software developments will be conducted in collaboration with academic partners or in collaboration with some industrial partners in the context of their private R&D or production activities. For all these software developments, we will use first the various (very) large parallel platforms available through CERFACS and GENCI in France (CCRT, CINES and IDRIS Computational Centers), and next the high-end parallel platforms that will be available via European and US initiatives or projects such that PRACE.

5.2. MaPHyS

MaPHyS (Massively Parallel Hybrid Solver) is a software package whose prototype was initially developed in the framework of the PhD thesis of Azzam Haidar (CERFACS) and further consolidated thanks to the ANR-CIS Solstice funding. This parallel linear solver couples direct and iterative approaches. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz.

In the framework of the INRIA technologic development actions; 24 man-month engineer (Yohan Lee-Tin-Yien) have been allocated to this software activity for the 2009-2011 period. The initial software prototype has been completely redesigned in order to enable us to easily interface any sparse direct solvers and develop new preconditioning technique. The first public release of the software is planned early 2012. The same software effort has been undertaken for interfacing any graph partitioning tools.

The MaPHyS package is very much a first outcome of the research activity described in Section 3.3. Finally, MaPHyS is a preconditioner that can be used to speed-up the convergence of any Krylov subspace method. We foresee to either embed in MaPHyS some Krylov solvers or to release them as standalone packages, in particular for the block variants that will be some outcome of the studies discussed in Section 3.3.

5.3. EPSN

EPSN (Environment for Computational Steering) is a software environment for the steering of legacy parallel-distributed simulations with simple GUI or more complex (possibly parallel) visualization programs (see Figure 1). In order to make a legacy simulation steerable, the user annotates the sourcecode with the EPSN API. These annotations provide the EPSN environment with two kinds of information: the description of the program structure according to a Hierarchical Task Model (HTM) and the description of the distributed data that will be remotely accessible. EPSN provides a distributed data model, that handles common scientific objects such as parameters, structured grids, particles/atoms and unstructured meshes. It is then possible to dynamically connect EPSN with a client program, that provides a GUI with some visualization & interaction features, as for instance SIMONE (Simulation MONitoring for Epsn). Once a client is connected, it interacts with the simulation via EPSN API. It is possible: 1) to control the execution flow of the remote simulation; 2) to access/modify its data onthefly; and 3) finally to invoke advanced user-defined routines in the simulation. The current version of EPSN is fully based on CORBA for communication on heterogeneous system and VTK/Paraview for visualization. A new release of EPSN, that will be fully based on MPI to handle efficient communication, is currently under development. A prototype is already working.
Figure 1. EPSN: software environment for $M \times N$ computational steering.
EPSN has been supported by the ACI-GRID program (grant number PPL02-03), the ARC RedGRID, the ANR MASSIM (grant number ANR-05-MMSA-0008-03) and the ANR CIS NOSSI (2007). More informations are available on our web site: http://www.labri.fr/projet/epsn. This software is publicaly available at Inria Gforge (http://epsn.gforge.inria.fr).

5.4. MPICPL

MPICPL (MPI CouPLing) is a software library dedicated to the coupling of parallel legacy codes, that are based on the well-known MPI standard. It proposes a lightweight and comprehensive programming interface that simplifies the coupling of several MPI codes (2, 3 or more). MPICPL facilitates the deployment of these codes thanks to the mpicplrun tool and it interconnects them automatically through standard MPI inter-communicators. Moreover, it generates the universe communicator, that merges the world communicators of all coupled-codes. The coupling infrastructure is described by a simple XML file, that is just loaded by the mpicplrun tool. Future releases will incorporate new features for checkpoint/restart and dynamic parallel code connection.

MPICPL was developed by the Inria HiePACS project-team for the purpose of the ANR CIS NOSSI. It uses advanced features of MPI2 standard. The framework is publicaly available at Inria Gforge: http://mpicpl.gforge.inria.fr.

5.5. MONIQA

MONIQA (MONitoring graphic user Interface for Qm/mm Applications) is a GUI specially designed for the monitoring & steering of the QM/MM application in the ANR CIS NOSSI project. It is based on Tulip, a graph visualization software http://tulip.labri.fr), used to display atoms and molecules. It proposes two working modes: offline or online. The offline mode is mainly used to load input files of DL_POLY & Siesta, and to prepare the quantum region for the QM/MM coupling. In online mode, the end-user can monitor & interact with the running QM/MM application thanks to EPSN. It is thus possible to visualize molecular and physical data (distances, angles, charges, energies), and to change simulation parameters on-the-fly, such as the target temperature of the system, thermo or barostat parameters, verbosity of output, ... MONIQA is based on QT4. It was developed specifically for the ANR NOSSI project and is available (restricted access) at Inria Gforge: http://nossi.gforge.inria.fr.

5.6. ScalFMM

ScalFMM (Parallel Fast Multipole Library for Large Scale Simulations) is a software library to simulate N-body interactions using the Fast Multipole Method. ScalFMM is based on the FMB prototype developed by Pierre Fortin during his PhD thesis. In the framework of the INRIA technologic development actions; 24 man-month engineer (Bérenger Bramas) have been allocated to this software activity started in January 2011.

ScalFMM intends to offer all the functionalities needed to perform large parallel simulations while enabling an easy customization of the simulation components: kernels, particles and cells. It works in parallel in a shared/distributed memory model using OpenMP and MPI. The software architecture has been designed with two major objectives: being easy to maintain and easy to understand. The code is extremely documented and the naming convention fully respected. Driven by its user-oriented philosophy, ScalFMM is using CMAKE as a compiler/installer tool. Even if ScalFMM is written in C++ it will support a C and fortran API soon.

The ScalFMM package is very much a first outcome of the research activity described in Section 3.4.
5.7. Other software

These software packages are or will be developed in collaboration with some academic partners (LIP6, LaBRI, CPMOH, IPREM, EPFL) or in collaboration with industrial partners (CEA, TOTAL, EDF) in the context of their private R&D or production activities.

- For the materials physics applications, a lot of development will be done in the context of ANR projects (NOSSI and proposal OPTIDIS, see Section 4.2) in collaboration with LaBRI, CPMOH, IPREM, EPFL and with CEA Saclay and Bruyère-le-Châtel.

- In the context of the PhD thesis of Mathieu Chanaud (collaboration with CEA/CESTA), we have developed a new parallel platform based on a combination of a geometric full multigrid solver and a direct solver (the PaStiX solver developed in the previous ScAlApplix project-team) to solve huge linear systems arising from Maxwell equations discretized with first-order Nédélec elements (see Section 3.3).

- Finally, we contribute to software developments for seismic analysis and imaging and for wave propagation in collaboration with TOTAL (use of GPU technology with CUDA).
5. Software

5.1. RPL P2P

Participants: Emmanuel Baccelli [correspondant], Matthias Philipp.

P2P-RPL is an implementation of draft-ietf-roll-p2p-rpl, providing reactive discovery of point-to-point routes in low power and lossy networks such as wireless sensor networks. The implementation is based on the Contiki operating system. See also the web page http://contiki-p2p-rpl.gforge.inria.fr/.

- Version: 0.4

5.2. OPERA infrastructure

Participants: Cédric Adjih [correspondant], Ichrak Amdouni, Pascale Minet, Ridha Soua.

OPERA-infrastructure is the system support code of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling.

5.3. OPERA perf simul

Participants: Cédric Adjih [correspondant], Ichrak Amdouni.

OPERA-perf-simul is a set of tools for simulation and performance evaluation as well as large scale tests of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling.

5.4. OPERA protocol

Participants: Cédric Adjih [correspondant], Ichrak Amdouni, Pascale Minet, Saoucene Mahfoudh.

OPERA-protocol is the heart of OPERA, the Optimized Protocol for Energy efficient Routing with node Activity scheduling. It includes EOND a neighborhood discovery protocol, EOSTC a protocol byuiding and maintaining a n energy efficient routing tree and SERENA a node coloring algorithm.

5.5. OPERA validation and tools

Participant: Cédric Adjih [correspondant].

OPERA-validation and tools is a set of tools for validation, debugging, analysis and visualization of OPERA protocol, the Optimized Protocol for Energy efficient Routing with node Activity scheduling. It operates either in a real embedded system or in simulation.
5. Software

5.1. Introduction

Most INDES software packages, even the older stable ones that are not described in the following sections are freely available on the Web. In particular, some are available directly from the INRIA Web site:

http://www.inria.fr/centre/sophia/innovation

Most other software packages can be downloaded from the INDES Web site:

http://www-sop.inria.fr/teams/indes

5.2. Functional programming

Participants: Frédéric Boussinot [Inria], Thomas Gazagnaire [Inria], Zhengqin Luo [Inria], Cyprien Nicolas [Inria], Tamara Rezk [Inria], Bernard Serpette [Inria], Manuel Serrano [correspondant].

5.2.1. The Bigloo compiler

The programming environment for the Bigloo compiler [5] is available on the INRIA Web site at the following URL: http://www-sop.inria.fr/teams/indes/fp/Bigloo. The distribution contains an optimizing compiler that delivers native code, JVM bytecode, and .NET CLR bytecode. It contains a debugger, a profiler, and various Bigloo development tools. The distribution also contains several user libraries that enable the implementation of realistic applications.

Bigloo was initially designed for implementing compact stand-alone applications under Unix. Nowadays, it runs harmoniously under Linux and MacOSX. The effort initiated in 2002 for porting it to Microsoft Windows is pursued by external contributors. In addition to the native back-ends, the Bigloo JVM back-end has enabled a new set of applications: Web services, Web browser plug-ins, cross platform development, etc. The new Bigloo .NET CLR back-end that is fully operational since release 2.6e enables a smooth integration of Bigloo programs under the Microsoft .NET environment.

5.2.2. Camloo

Camloo is a caml-light to bigloo compiler, which was developed few years ago to target bigloo 1.6c. New major releases 0.4.x of camloo have been done to support bigloo 3.4 and bigloo 3.5. Camloo make it possible for the user to develop seamlessly a multi-language project, where some files are written in caml-light, in C, and in bigloo. Unlike the previous versions of camloo, 0.4.x versions do not need a modified bigloo compiler to obtain good performance. Currently, the only supported backend for camloo is bigloo/C. We are currently rewriting the runtime of camloo in bigloo to get more portability and to be able to use HOP and camloo together.

5.2.3. The FunLoft language

FunLoft (described in http://www-sop.inria.fr/teams/indes/rp/FunLoft) is a programming language in which the focus is put on safety and multicore.

FunLoft is built on the model of FairThreads which makes concurrent programming simpler than usual preemptive-based techniques by providing a framework with a clear and sound semantics. FunLoft is designed with the following objectives:

- provide a safe language, in which, for example, data-races are impossible.
- control the use of resources (CPU and memory), for example, memory leaks cannot occur in FunLoft programs, which always react in finite time.
- have an efficient implementation which can deal with large numbers of concurrent components.
- benefit from the real parallelism offered by multicore machines.
Networks, Systems and Services, Distributed Computing - Software - Project-Team INDES

A first experimental version of the compiler is available on the Reactive Programming site http://www-sop.inria.fr/teams/indes/rp. Several benchmarks are given, including cellular automata and simulation of colliding particles.

5.3. Web programming

Participants: Gérard Berry [Inria], Cyprien Nicolas [Inria], Manuel Serrano [correspondant].

5.3.1. The HOP web programming environment

HOP is a higher-order language designed for programming interactive web applications such as web agendas, web galleries, music players, etc. It exposes a programming model based on two computation levels. The first one is in charge of executing the logic of an application while the second one is in charge of executing the graphical user interface. HOP separates the logic and the graphical user interface but it packages them together and it supports strong collaboration between the two engines. The two execution flows communicate through function calls and event loops. Both ends can initiate communications.

The HOP programming environment consists in a web broker that intuitively combines in a single architecture a web server and a web proxy. The broker embeds a HOP interpreter for executing server-side code and a HOP client-side compiler for generating the code that will get executed by the client.

An important effort is devoted to providing HOP with a realistic and efficient implementation. The HOP implementation is validated against web applications that are used on a daily-basis. In particular, we have developed HOP applications for authoring and projecting slides, editing calendars, reading RSS streams, or managing blogs.

HOP has won the software open source contest organized by the ACM Multimedia Conference 2007 http://mmc36.informatik.uni-augsburg.de/acmmm2007/. It is released under the GPL license. It is available at http://hop.inria.fr

5.4. Language-based security

Participants: Zhengqin Luo [Inria], Tamara Rezk [correspondant].

5.4.1. CFlow

The prototype compiler “CFlow” takes as input code annotated with information flow security labels for integrity and confidentiality and compiles to F# code that implements cryptography and protocols that satisfy the given security specification.

Cflow has been coded in F#, developed mainly on Linux using mono (as a substitute to .NET), and partially tested under Windows (relying on .NET and Cygwin). The code is distributed under the terms of the CeCILL-B license http://www.msr-inria.inria.fr/projects/sec/cflow/index.html.

5.4.2. FHE type-checker

We have developed a type checker for programs that feature modern cryptographic primitives such as fully homomorphic encryption. The type checker is thought as an extension of the “CFlow” compiler developed last year on the same project. It is implemented in F#. The code is distributed under the terms of the CeCILL-B license http://www.msr-inria.inria.fr/projects/sec/cflow/index.html.

5.4.3. Mashic compiler

The Mashic compiler is applied to mashups with untrusted scripts. The compiler generates mashups with sandboxed scripts, secured by the same origin policy of the browsers. The compiler is written in Bigloo and can be found at http://www.mashic.net.
5.5. Old software

5.5.1. Skribe

SKRIBE is a functional programming language designed for authoring documents, such as Web pages or technical reports. It is built on top of the SCHEME programming language. Its concrete syntax is simple and looks familiar to anyone used to markup languages. Authoring a document with SKRIBE is as simple as with HTML or LaTeX. It is even possible to use it without noticing that it is a programming language because of the conciseness of its original syntax: the ratio tag/text is smaller than with the other markup systems we have tested.

Executing a SKRIBE program with a SKRIBE evaluator produces a target document. It can be HTML files for Web browsers, a LaTeX file for high-quality printed documents, or a set of info pages for on-line documentation.

5.5.2. Scheme2JS

Scm2JS is a Scheme to JavaScript compiler distributed under the GPL license. Even though much effort has been spent on being as close as possible to R5RS, we concentrated mainly on efficiency and interoperability. Usually Scm2JS produces JavaScript code that is comparable (in speed) to hand-written code. In order to achieve this performance, Scm2JS is not completely R5RS compliant. In particular it lacks exact numbers. Interoperability with existing JavaScript code is ensured by a JavaScript-like dot-notation to access JavaScript objects and by a flexible symbol-resolution implementation.

Scm2JS is used on a daily basis within HOP, where it generates the code which is sent to the clients (web-browsers). Scm2JS can be found at http://www-sop.inria.fr/indes/scheme2js.
KERDATA Team

5. Software

5.1. BlobSeer

Contact: Gabriel Antoniu, gabriel.antoniu@inria.fr.

Participants from the KerData team: Alexandra Carpen-Amarie, Diana Moise, Viet-Trung Tran, Alexandu Costan, Gabriel Antoniu, Luc Bougé.

Presentation: BlobSeer is the core software platform for most current projects of the KerData team. It is a data storage service specifically designed to deal with the requirements of large-scale data-intensive distributed applications that abstract data as huge sequences of bytes, called BLOBs (Binary Large OBjects). It provides a versatile versioning interface for manipulating BLOBs that enables reading, writing and appending to them.

BlobSeer offers both scalability and performance with respect to a series of issues typically associated with the data-intensive context: scalable aggregation of storage space from the participating nodes with minimal overhead, ability to store huge data objects, efficient fine-grain access to data subsets, high throughput in spite of heavy access concurrency, as well as fault-tolerance.

Users: Work is currently in progress in several formalized projects (see previous section) to integrate and leverage BlobSeer as a data storage back-end in the reference cloud environments: a) Microsoft Azure; b) the Nimbus cloud toolkit developed at Argonne National Lab (USA); and c) in the OpenNebula IaaS cloud environment developed at UCM (Madrid).

URL: http://blobseer.gforge.inria.fr/

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on INRIA’s forge. Version 1.0 (released late 2010) registered with APP: IDDIN.FR.001.310009.000.S.P000.10700.

5.2. Damaris

Contact: Gabriel Antoniu, gabriel.antoniu@inria.fr.

Participants from the KerData team: Matthieu Dorier, Gabriel Antoniu.

Presentation: Damaris is a middleware for multicore SMP nodes enabling them to efficiently handle data transfers for storage and visualization. The key idea is to dedicate one or a few cores of each SMP node to the application I/O. It is developed within the framework of a collaboration between KerData and the Joint Laboratory for Petascale Computing (JLPC). The current version enables efficient asynchronous I/O, hiding all I/O related overheads such as data compression and post-processing. On-going work is targeting fast direct access to the data from running simulations, and efficient I/O scheduling.

Users: Damaris has been preliminarily evaluated at NCSA (Urbana-Champaign) with the CM1 tornado simulation code. CM1 is one of the target applications of the Blue Waters supercomputer developed by at NCSA/UIUC (USA), in the framework of the INRIA-UIUC Joint Lab (JLPC). Work is currently in progress to use Damaris as an intermediate data layer optimizing simulation/visualization coupling for several HPC scientific applications intended to run on the Blue Waters.

URL: http://damaris.gforge.inria.fr/

License: GNU Lesser General Public License (LGPL) version 3.

Status: This software is available on INRIA’s forge. Registration with APP is in progress.
5.3. Derived software

Derived from BlobSeer, two additional platforms are currently being developed within KerData: 1) Pyramid, a software service for array-oriented active storage developed within the framework of the PhD thesis of Viet-Trung Tran (see Section 6.7); and 2) TomusBlobs, a PaaS-level storage service for Azure clouds developed within the framework of the thesis of Radu Tudoran in relation to the A-Brain project. These platforms have not been publicly released yet.
MADYNES Project-Team

5. Software

5.1. Voip bots

Participants: Mohamed Nassar [contact], Olivier Festor.

VoIPbot is a VoIP security tool created as a demonstrator of how attacks can be launched against VoIP/SIP services and users in a remotely and distributed manner. The environment contains bots that can be remotely managed over an Internet Relay Chat (IRC) channel from a central manager. Our bots are currently able to perform the following tasks:

- send SPAM over IP Telephony (SPIT),
- distributed denial of service through intensive generation of invite messages to a target device,
- active scanning of users through incremental options messages issuance to servers and response analysis,
- cracking through brute-force testing of passwords against an identified user account,
- simple device scanning and fingerprinting,
- target aware device fuzzing.

The tool is developed using the Java programming language. It uses the JAIN-SIP, JMF and PIRCBOT libraries. The tool is distributed under a GPL2 Open Source license. Reports show its use mainly in the testing business so far.

5.2. SecSIP

Participants: Abdelkader Lahmadi [contact], Olivier Festor.

SecSIP is developed by the team to defend SIP-based (The Session Initiation Protocol) services from known vulnerabilities. It presents a proactive point of defense between a SIP-based network of devices (servers, proxies, user agents) and the open Internet. Therefore, all SIP traffic is inspected and analyzed against authored Veto specification before it is forwarded to these devices. When initializing, the SecSIP runtime starts loading and parsing authored Veto blocks to identify different variables, event patterns, operations and actions from each rule. It implements an input and output layer, to capture, inject, send and receive SIP packets from and to the network. Intercepted packets are moved to the SIP Packet parser module. The main function of this module is to extract different fields within a SIP message and trigger events specified within the definition blocks. During each execution cycle when a SIP message arrives, the SecSIP runtime uses a data flow acyclic graph network to find definition matching rules and triggers defined events. The paired events in each operator node are propagated over the graph until a pattern is satisfied. When the pattern is satisfied, the respective rule is fired and the set of actions is executed.

SecSIP is freely available on the Internet and has been demonstrated in various High Security Labs exhibits in 2011.

5.3. NDPMon

Participants: Isabelle Chrisment, Olivier Festor [contact].

\(^2\) http://secsip.gforge.inria.fr/doku.php
The Neighbor Discovery Protocol Monitor (NDPMon) is an IPv6 implementation of the well-known ArpWatch tool. NDPMon monitors the pairing between IPv6 and Ethernet addresses (NDP activities: new station, changed Ethernet address, flip flop...). NDPMon also detects attacks on the NDP protocol, as defined in RFC 3756 (bogon, fake Router Advertisements...). New attacks based on the Neighbor Discovery Protocol and Address Auto-configuration (RFC 2461 and RFC 2462) have been identified and integrated in the tool. An XML file describes the default behavior of the network, with the authorized routers and prefixes, and a second XML document containing the neighbors database is used. This second file can be filled during a learning phase. All NDP activities are logged in the syslog utility, and so the attacks, but these ones are also reported by mail to the administrator. Finally, NDPMon can detect stack vulnerabilities, like the assignment of an Ethernet broadcast address on an interface.

NDPMon comes along with a WEB interface acting as a GUI to display the informations gathered by the tool, and give an overview of all alerts and reports. Thanks to color codes, the WEB interface makes possible for the administrator to have an history of what happened on his network and identify quickly problems. All the XML files used or produced by the daemon (neighbor cache, configuration file and alerts list) are translated in HTML via XSL for better readability. A statistic module is also integrated and gives informations about the discovery of the nodes and their type (MAC manufaturer distribution ...).

The software package and its source code is freely distributed under an opensource license (LGPL). It is implemented in C, and is available through a SourceForge project at http://ndpmon.sf.net. An open source community is now established for the tool which has distributions for several Operating Systems (Linux, FreeBSD, OpenBSD, NetBSD and Mac OS X). It is also integrated in FreeBSD ports at http://www.freebsd.org/cgi/cvsweb.cgi/ports/net-mgmt/ndpmon/. Binary distributions are also available for .deb and .rpm based Linux flavors.

5.4. AA4MM

Participants: Laurent Ciarletta, Julien Siebert [main developer].

This work has been undertaken in a joint Phd Thesis between the Madynes and MAIA Teams. Vincent Chevrier (MAIA team, LORIA) has been the co-advisor of this PhD and correspondant for this software.

AA4MM (Agents and Artefacts for Multi-modeling and Multi-simulation) is a framework for coupling existing and heterogeneous models and simulators in order to model and simulate complex systems. This is the first implementation of the AA4MM meta-model proposed in Julien Siebert’s PhD. It is written in Java and relies upon Java Messaging Services (JMS) for its distributed version.

AA4MM can be downloaded at http://www.loria.fr/~siebertj/aa4mm/index.html.

5.5. MASDYNE

Participants: Laurent Ciarletta, Julien Siebert [main developer].

This work is undertaken in a joint Phd Thesis between the Madynes and MAIA Teams. Vincent Chevrier (MAIA team, LORIA) has been a co-advisor of this PhD and correspondant for this software.

Other contributors to this software are: Tom Leclerc (Madynes), Francois Klein, Christophe Torin, Marcel Lamenu, Guillaume Favre and Amir Toly.

MASDYNE (Multi-Agent Simulator of DYnamic Networks usErs) is a multi-agent simulator for modeling and simulating users behaviors in mobile ad hoc network. This software is part of joint work with MAIA team, on modeling and simulation of ubiquitous networks.

It has been notably coupled with a network simulator (JANE : Java Adhoc Network Development Environment) to advanced behavior capabilities to standard network simulations.
MAESTRO Project-Team (section vide)
MASCOTTE Project-Team

5. Software

5.1. Grph

Participants: Nathann Cohen, David Coudert, Luc Hogie [correspondant], Aurélien Lancin, Grégory Morel, Issam Tahiri.

Around 20,000 lines of code, developed in Java.

The GRPH project takes over Dipergrafs which was introduced in the activity report of 2010. A drastic change in the model of Dipergrafs justified the name change.

The objective of GRPH is to provide researchers and engineers a suitable graph library for graph algorithms experimentation and network simulation. GRPH is mainly a software library, but it also comes with a set of executable files for user interaction and graph format conversion; as such, it can be used autonomously.

Performance and accessibility are the primary targets of the GRPH library. At every stage, it is designed to be efficient in terms of: computation time (use of parallelism, caching, adequate data structures, native code, etc.); memory requirements (use of Java primitives); and portability (it is written in a Java and C). Its model considers mixed graphs composed of (un)directed simple- and hyper-edges. It can handle large dynamic graphs in the order of millions of nodes. GRPH comes with a collection of base graph algorithms which are regularly augmented.

So far, most known users of the GRPH library are part of INRIA and of the FP7 STREP EULER project. GRPH is distributed under the terms of a license defined by its contributors and is available for download. This license allows free usage and access to the source code. See http://www-sop.inria.fr/mascotte/software/grph.

In 2011, GRPH was augmented over Dipergrafs of a number of features suited to its usage within the MASCOTTE research team. These include: addition of numerous graph manipulation methods; introduction of an incidence-list data structure for the representation of graphs; introduction of an adaptive data structure for the representation of sets (based on hash-tables and bit-vectors); integration of implementations of "maximum clique" and "sub-graph isomorphism" algorithms by Christine Solnon (CNRS, INSA Lyon). These sources, written in C, are compiled on-the-fly; integration of implementation of "graph isomorphism" algorithm by Brendan McKay (Australian National University); iteration of implementation of "number of triangles" algorithm by Matthieu Latapy (LIP6); introduction of a bridge to the Mascot/OpenGVE library; introduction of a bridge to the JUNG library; addition of numerous graph algorithms; introduction of a new layer atop GRPH which allows the representation and manipulation of graph as Java objects, like it is done in other libraries such as Mascot, JUNG, etc.; introduction of an efficient mechanism for the definition of graph properties; addition of graph reporting facilities.

On-going works concern the distributed execution of graph algorithms, a bridge to Sage, and the graphical edition of graphs.

5.2. DRMSim

Participants: David Coudert, Luc Hogie [correspondant], Aurélien Lancin, Nicolas Nisse, Issam Tahiri.

Around 45,000 lines, developed in Java, collaboration between MASCOTTE and researchers in LaBRI (95% MASCOTTE).
The expansion of the Internet results in a number of issues: BGP (Border Gateway Protocol) starts to show its limits in terms of the number of routing table entries it can manage. More efficient dynamic routing protocols are thus under investigation. However, because deploying under-development routing protocols on the Internet is not practicable at a large-scale, simulation is a necessary step to validate the properties of a newly proposed routing scheme. Unfortunately, the simulation of routing protocols over large networks poses real challenges due to the limited computational capabilities of computers. Existing simulation tools exhibit limitations in terms of the number of nodes they can handle and of the models they propose. This motivated us to conceive and develop DRMSim (Dynamic Routing Model Simulator): a network simulator which addresses the specific problem of large-scale simulations of routing models.

DRMSim relies on a discrete-event simulation engine. It proposes a general routing model which accommodates any network configuration. Aside to this, it includes specific models for Generalized Linear Preference (GLP), and K-chordal network topologies, as well as implementations of routing protocols, including the routing protocol proposed in [99] and lightweight versions of BGP (Border Gateway Protocol).

Recent developments (in 2011) in the DRMSim simulator include the four following elements:

1. The initial framework was composed of a routing model. It now incorporates a system model and a metric model. In addition, the system model now considers the dynamic evolution of the simulated network. This dynamic behavior includes the maintenance operations on the network infrastructure as well as router failures. This model stores the connectivity of routers and links before their failure is simulated. This information is used for the simulation of the recovery procedure. This model takes as its input parameter the distribution of failure probability for both routers and links.

2. The metric model has been fully rewritten and is now geared towards computational performance and flexibility. Taking measures along a discrete-event simulation can be performed in many ways. DRMSim uses a new approach which consists in a metric model listening to the simulation and system models. The user can define its own metrics. Memory and CPU usages depend on which metrics are defined, to which set of routers/links they are applied, how many measures are taken and their computational complexity. It is possible to restrict the model to a small amount of nodes/links by selectors provided as input parameters. At the cost of memory and CPU usage, metrics measures can be stored as time-ordered sequence of values. To reduce the need of resources, a single global measure for each metric can be computed. Finally, metrics can be computed globally on the set of selected entities (links/routers) but also separately for each entity.

3. DRMSim enables the definition of customized simulation scenarios and stateful simulation campaigns. Commonly, a simulation campaign consists in iterating over the set of combinations of parameter values, calling the simulation function for every combination. These combinations cannot be found randomly nor can they be determined using linear functions. Indeed, most of the time there exist correlations between the parameters involved. Also for performance reasons, the end-user will prefer non-linear (most often logarithmic) evolutions for the values of the parameters. The definition of the set of combinations is strongly linked to the simulated system and the time needed to solve it. DRMSim provides a simulation methodology that describes (programmatically) the way a simulation campaign should be conducted.

The duration of a simulation can be as long as several hours (or days). In the context of a simulation campaign where numerous simulations are executed, it is important that re-starting a simulation campaign that was interrupted does not entail the re-computation of already computed results. In order to do this, DRMSim stores on disk every step of the execution of a simulation campaign.

In a simulation campaign, simulation runs are independent (no simulation depends on the result computed by another simulation). Consequently they can be executed in parallel. Because one simulation is most likely to use large amount of memory and to be multi-threaded, parallelizing the simulation campaign on one single computer is a poor parallelization scheme. Instead, we currently work at enabling the remote parallel execution of several simulation runs, with the same distribution framework that is used in the GRPH library.

4. Finally, DRMSim manipulates graph abstractions, allowing the user to force the use of a library different from the default one, i.e. GRPH.
5.3. Mascopt and openGVE

**Participant:** Michel Syska [correspondant].

Developed in Java.

MASCOPT [98](MASCOTTE Optimization) is a Java library distributed under the terms of the LGPL license which is dedicated to graph and network processing. MASCOPT includes a collection of Java interfaces and classes that implement fundamental data structures and algorithms. The forthcoming public distribution of MASCOPT will appear under the name of the openGVE project, MASCOPT being one implementation of the bridge graph interface (see [http://opengve.inria.fr/bridge-graph-interface/apidocs/fr/inria/opengve/bridge/interfaces/Graph.html](http://opengve.inria.fr/bridge-graph-interface/apidocs/fr/inria/opengve/bridge/interfaces/Graph.html)). The objective is to allow easy integration of different implementations. The applications already written will not be affected. They will have different choices of internal implementation which may lead to better performances for specific issues such as large graphs processing.

The main objective of MASCOPT project is to ease software development in the field of network optimization. Examples of problems include routing, grooming, survivability, and virtual network design. MASCOPT helps implementing a solution to such problems by providing a data model of the network and the demands, classes to handle data and ready to use implementations of existing algorithms or linear programs (e.g. shortest paths or integral multicommodity flow).

A key feature of MASCOPT is to provide a generic linear programming object interface which allows users to program the same way whether the target solver is IBM ILOG CPLEX, GLPK (GNU Linear Programming Kit) or CLP/CBC (accessed through JNI).

MASCOPT has been intensively used in the past within MASCOTTE industrial cooperation programs for experimentation and validation purposes as for example with Alcatel Space Technologies and Orange Labs. Today, the library is used within the framework of the ANR AGAPE to implement FPT algorithms (work done at LIFO).

See also the web page [http://www-sop.inria.fr/mascotte/mascopt/](http://www-sop.inria.fr/mascotte/mascopt/).

5.4. Open Simulation Architecture (OSA)

**Participants:** Olivier Dalle [correspondant], Van Dan Nguyen, Judicaël Ribault.

Developed in Java (80%) and XML, AspectJ, etc. Represent the work of about 8 man/year during the last 6 years.

Component-based modeling has many well-known good properties. One of these properties is the ability to distribute the modeling effort amongst several experts, each having his/her own area of system expertise. Clearly, the less experts have to care about areas of expertise of others, the more efficient they are in modeling sub-systems in their own area. Furthermore, the process of studying complex systems using discrete-event computer simulations involves several areas of non-system expertise, such as discrete-event techniques or experiment planning.

The Open Simulation Architecture (OSA) [97] is designed to enforce a strong separation of the end-user roles and therefore, ensure a successful cooperation of all the experts involved in the process of simulating complex systems.

The OSA architecture is also intended to meet the expectations of a large part of the discrete-event simulation community: it provides an open platform intended to support researchers in a wide range of their simulation activities, and allows the reuse and sharing of system models in the simulation community by means of a flexible and generic component model (Fractal).
Many discrete-event simulators are developed concurrently, but with identical or similar purpose. Another goal of OSA is to favor the reuse and integration of simulation software components and models. To favor reuse, OSA uses a layered approach to combine the modeling, simulation, and related concerns, such as instrumentation or deployment. This ability is demonstrated by the successful integration and reuse of third-party components, such as Scave, the analysis module of Omnet++, or a large number of the James II plugins developed by the University of Rostock. OSA is both a testbed for experimenting new simulation techniques and a tool for real case studies.

OSA is Open Source (LGPL) and is available for download on the INRIA forge server [http://osa.gforge.inria.fr/](http://osa.gforge.inria.fr/).

See also the web page [http://osa.inria.fr/](http://osa.inria.fr/).

### 5.5. SageMath

**Participants:** Nathann Cohen [correspondant], David Coudert, Leonardo Sampaio.

Developed in Python, Cython, and C++. N. Cohen wrote more than 180 patches and N. Cohen, D. Coudert and L. Sampaio reviewed more than 120 others for inclusion in Sage.

Sagemath is a free open-source mathematics software aiming at becoming an alternative to Maple and Matlab. Initially created by William Stein (Professor of mathematics at Washington University), Sagemath is currently developed by more than 180 contributors around the world (mostly researchers). It has currently more than 200 MB of source code and the graph module consists of 40,000 lines. It was initially of interest for Mascotte because of its large library in Combinatorics and Graph Theory. This year, impressive improvements have been made to this library. In particular, N. Cohen contributed a lot into the following: 1) implementation of a generic interface between Sage and existing (Mixed Integer) Linear Program solvers, 2) implementation of exact algorithms for common Polynomial/NP-Complete graph problems, often through the use of Linear Programs, and 3) improving Sage’s documentation by participating to the writing of a french manual on the use of Sage with 10 other french scientists. New patches are in preparation in the group for possible inclusion in Sage.

Sage’s Graph and Linear Programing libraries are currently used by Mascotte members to test algorithms or compare their performances, as well as to prove/disprove theoretical conjectures and for teaching purposes in the Master IFI, stream UBINET.

### 5.6. Utilities

#### 5.6.1. Java4unix

**Participant:** Luc Hogie [correspondant].

More than 5,000 lines, developed in Java.

Java4unix proposes a development and distribution framework which simplifies the use of Java for UNIX software programming/distribution. Until now, Java could hardly be used for the development UNIX applications because invoking Java applications from the UNIX shell must be done through an explicit call to the Java virtual machine and writing simple things in Java often requires long coding. Java4unix aims at filling those two gaps by providing a UNIX installer for java applications, turning them to standard UNIX application and a framework that UNIX programmers may use to manipulate files/text, etc.

Java4unix includes a module which enables the reporting and automatic releasing of Eclipse Java projects. This module was formely separated from Java4unix and was referred to as EPR.

See also the web page [http://www-sop.inria.fr/members/Luc.Hogie/java4unix/](http://www-sop.inria.fr/members/Luc.Hogie/java4unix/).

#### 5.6.2. Jalinopt

**Participants:** Luc Hogie [correspondant], Grégory Morel.
Developed in Java.

Many mathematical and engineering problems can be expressed as linear programs, and doing so facilitates their resolution. Indeed it is generally more convenient to transform a domain-specific problem into a linear-optimizable one (that can be solved by any solver) rather than writing a complex domain-specific algorithm. In the case of graph theory, problems like flows, minimum vertex cover, maximum stable can be conveniently represented via linear programs.

Jalinopt is a Java toolkit for building and solving linear programs. It consists of a straightforward object-oriented model for linear programs, as well as a bridge to most common solvers, including GLPK and CPLEX. Albough Jalinopt is inspired by Mascopt and JavaILP, it provides a significantly different model and an utterly different approach to connecting to the solver. In particular this approach, based in inter-process piping, offers better portability, and the possibility to connect (via SSH) to solvers on remote computers.

See also the web page http://www-sop.inria.fr/members/Luc.Hogie/jalinopt/.

5.6.3. JavaFarm

Participant: Luc Hogie [correspondant].

More than 1,500 lines, developed in Java.

JavaFarm is a middleware enabling the distribution of Java applications across farms of servers. Its workflow basically enables an application to locally aggregate code and data into an object, called job that will migrate to another computer, where it will be computed. When a job completes, its result is transferred back to the caller. Among other features, JavaFarm supports futures (asynchronous job executions), thereby enabling parallelization of the distributed code. The design objectives of JavaFarm are to make distribution and parallelism as transparent and easy as possible.

See also the web page http://www-sop.inria.fr/members/Luc.Hogie/javafarm/.

5.6.4. Mascsim

Participants: Luc Hogie [correspondant], Aurélien Lancin, Issam Tahiri.

Around 12,000 lines, developed in Java.

Mascsim is a distributed discrete event simulator whose main target is to be easy to use.

Unlike most discrete-event simulators, the researcher who is using Mascsim is required to provide only the bare minimum material needed for the simulation: a model for the system, a set of events describing what is going on in the system, as well as a set of metrics of interest.

The simulation process is then entirely automatized.

See also the web page http://www-sop.inria.fr/mascotte/software/mascsim/.

5.6.5. P2PVSim

Participant: Remigiusz Modrzejewski [correspondant].

Around 8,000 lines, developed in Python.

P2PVSim is a simple discrete-event simulator created for analyzing theoretical properties of peer-to-peer live video streaming algorithms. Implemented in Python it was designed with clarity and extensibility in mind from the beginning. It is capable of simulating overlays of a few thousands of peers. Multiple control protocols have been implemented. At the same time, a lot of work was put into the performance and scalability aspects of the software. Currently it is meant for simulating overlays of a few thousand peers running multiple control protocols that have been implemented.
5. Software

5.1. Tools for cluster management and software development

Participant: Olivier Richard [correspondant].

The KA-Tools is a software suite developed by MESCAL for exploitation of clusters and grids. It uses a parallelization technique based on spanning trees with a recursive starting of programs on nodes. Industrial collaborations were carried out with Mandrake, BULL, HP and Microsoft.

5.1.1. KA-Deploy

KA-Deploy is an environment deployment toolkit that provides automated software installation and reconfiguration mechanisms for large clusters and light grids. The main contribution of KA-Deploy toolkit is the introduction of a simple idea, aiming to be a new trend in cluster and grid exploitation: letting users concurrently deploy computing environments tailored exactly to their experimental needs on different sets of nodes. To reach this goal KA-Deploy must cooperate with batch schedulers, like OAR, and use a parallel launcher like Taktuk (see below).

5.1.2. Taktuk

Taktuk is a tool to launch or deploy efficiently parallel applications on large clusters, and simple grids. Efficiency is obtained thanks to the overlap of all independent steps of the deployment. We have shown that this problem is equivalent to the well known problem of the single message broadcast. The performance gap between the cost of a network communication and of a remote execution call enables us to use a work stealing algorithm to realize a near-optimal schedule of remote execution calls. Currently, a complete rewriting based on a high level language (precisely Perl script language) is under progress. The aim is to provide a light and robust implementation. This development is lead by the MOAIS project-team.

5.2. OAR: Batch scheduler for clusters and grids

Participant: Olivier Richard [correspondant].

The OAR project focuses on robust and highly scalable batch scheduling for clusters and grids. Its main objectives are the validation of grid administration tools such as Taktuk, the development of new paradigms for grid scheduling and the experimentation of various scheduling algorithms and policies.

The grid development of OAR has already started with the integration of best effort jobs whose purpose is to take advantage of idle times of the resources. Managing such jobs requires a support of the whole system from the highest level (the scheduler has to know which tasks can be canceled) down to the lowest level (the execution layer has to be able to cancel awkward jobs). The OAR architecture is perfectly suited to such developments thanks to its highly modular architecture. Moreover, this development is used for the CiGri grid middleware project.

The OAR system can also be viewed as a platform for the experimentation of new scheduling algorithms. Current developments focus on the integration of theoretical batch scheduling results into the system so that they can be validated experimentally.

See also the web page http://oar.imag.fr.

5.3. FTA: Failure Trace Archive

Participant: Derrick Kondo [correspondant].
With the increasing functionality, scale, and complexity of distributed systems, resource failures are inevitable. While numerous models and algorithms for dealing with failures exist, the lack of public trace data sets and tools has prevented meaningful comparisons. To facilitate the design, validation, and comparison of fault-tolerant models and algorithms, we led the creation of the Failure Trace Archive (FTA), an on-line public repository of availability traces taken from diverse parallel and distributed systems.

While several archives exist, the FTA differs in several respects. First, it defines a standard format that facilitates the use and comparison of traces. Second, the archive contains traces in that format for over 20 diverse systems over a time span of 10 years. Third, it provides a public toolbox for failure trace interpretation, analysis, and modeling. The FTA was released in November 2009. It has received over 11,000 hits since then. The FTA has had national and international impact. Several published works have already cited and benefited from the traces and tools of the FTA. Simulation toolkits for distributed systems, such as SimGrid (CNRS, France) and GridSim (University of Melbourne, Australia), have incorporated the traces to allow for simulations with failures.

5.4. SimGrid: simulation of distributed applications
Participants: Arnaud Legrand [correspondant], Lucas Schnorr, Pierre Navarro, Sascha Hunold, Laurent Bobelin.

SimGrid is a toolkit that provides core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The specific goal of the project is to facilitate research in the area of distributed and parallel application scheduling on distributed computing platforms ranging from simple network of workstations to Computational Grids.

We have released one new major version (3.6) of SimGrid (June 2011) and two minor versions (June and October 2011). These versions include our current work on visualization, analysis of large scale distributed systems, and extremely scalable simulation. See also the web page http://simgrid.gforge.inria.fr/ .

5.5. TRIV A: interactive trace visualization
Participants: Lucas Schnorr [correspondant], Arnaud Legrand.

TRIV A is an open-source tool used to analyze traces (in the Pajé format) registered during the execution of parallel applications. The tool serves also as a sandbox for the development of new visualization techniques. Some features include: Temporal integration using dynamic time-intervals; Spatial aggregation through hierarchical traces; Scalable visual analysis with squarified treemaps; A Custom Graph Visualization.

See also the web page http://triva.gforge.inria.fr/ .

5.6. $\psi$ and $\psi^2$: perfect simulation of Markov Chain stationary distributions
Participant: Jean-Marc Vincent [correspondant].

$\psi$ and $\psi^2$ are two software tools implementing perfect simulation of Markov Chain stationary distributions using coupling from the past. $\psi$ starts from the transition kernel to derive the simulation program while $\psi^2$ uses a monotone constructive definition of a Markov chain. They are available at http://www-id.imag.fr/Logiciels/psi/ .
MOAIS Project-Team

5. Software

5.1. KAAPI

Participants: Thierry Gautier [correspondant], Vincent Danjean, Pierre Neyron.

KAAPI means Kernel for Adaptative, Asynchronous Parallel and Interactive programming. It is a C++ library that allows to execute multithreaded computation with data flow synchronization between threads. The library is able to schedule fine/medium size grain program on distributed machine. The data flow graph is dynamic (unfold at runtime). Target architectures are clusters of SMP machines. Main features are: * It is based on work-stealing algorithms; * It can run on various processors; * It can run on various architectures (clusters or grids); * It contains non-blocking and scalable algorithms.

See also the web page http://kaapi.gforge.inria.fr.

- ACM: D.1.3
- License: CeCILL
- OS/Middleware: Unix (Linux, MacOSX, ...)
- Programming language: C/C++, Fortran

5.2. OAR

Participants: Pierre Neyron [correspondant MOAIS], Grégory Mounié.

OAR is a batch scheduler developed by Mescal team (correspondant: Olivier Richard). The MOAIS team develops the central automata and the scheduling module that includes successive evolutions and improvements of the policy. OAR is used to schedule jobs both on the CiGri (Grenoble region) and Grid5000 (France) grids. CiGri is a production grid that federates about 500 heterogeneous resources of various Grenoble laboratories to perform computations in physics. MOAIS has also developed the distributed authentication for access to Grid5000.

See also the web page http://oar.imag.fr.

5.3. SOFA

Participant: Bruno Raffin [correspondant].

SOFA is an Open Source framework primarily targeted at real-time simulation, with an emphasis on medical simulation. It is mostly intended for the research community to help develop newer algorithms, but can also be used as an efficient prototyping tool. Based on an advanced software architecture, it allows to: - create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; - modify most parameters of the simulation (deformable behavior, surface representation, solver, constraints, collision algorithm, etc.) by simply editing an xml file; - build complex models from simpler ones using a scene-graph description; - efficiently simulate the dynamics of interacting objects using abstract equation solvers; - reuse and easily compare a variety of available methods.

See also the web page http://www.sofa-framework.org/.

- ACM: J.3
- Programming language: C/C++

5.4. TakTuk - Adaptive large scale remote execution deployment

Participants: Guillaume Huard [correspondant], Pierre Neyron.
TakTuk is a tool for deploying remote execution commands to a potentially large set of remote nodes. It spreads itself using an adaptive algorithm and set up an interconnection network to transport commands and perform I/Os multiplexing/demultiplexing. The TakTuk algorithms dynamically adapt to environment (machine performance and current load, network contention) by using a reactive algorithm that mix local parallelization and work distribution. Characteristics:

- adaptivity: efficient work distribution is achieved even on heterogeneous platforms thanks to an adaptive work-stealing algorithm
- scalability: TakTuk has been tested to perform large size deployments (hundreds of nodes), either on SMPs, regular clusters or clusters of SMPs
- portability: TakTuk is architecture independent (tested on x86, PPC, IA-64) and distinct instances can communicate whatever the machine they’re running on
- configurability: mechanics are configurable (deployment window size, timeouts, ...) and TakTuk outputs can be suppressed/formatted using I/O templates
- auto-propagation: the engine can spread its own code to remote nodes in order to deploy itself
- communication layer: nodes successfully deployed are numbered and perl scripts executed by TakTuk can send multicast communications to other nodes using this logical number
- information redirection: I/O and commands status are multiplexed from/to the root node. http://taktuk.gforge.inria.fr under GNU GPL licence.

5.5. KRASH - Kernel for Reproduction and Analysis of System Heterogeneity

Participants: Guillaume Huard [correspondant], Swann Perarnau.

KRASH is a tool for reproducible generation of system-level CPU load. This tool is intended for use in shared memory machines equipped with multiple CPU cores that are usually exploited concurrently by several users. The objective of KRASH is to enable parallel application developers to validate their resource use strategies on a partially loaded machine by replaying an observed load in concurrence with their application. To reach this objective, KRASH relies on a method for CPU load generation which behaves as realistically as possible: the resulting load is similar to the load that would be produced by concurrent processes run by other users. Nevertheless, contrary to a simple run of a CPU-intensive application, KRASH is not sensitive to system scheduling decisions. The main benefit brought by KRASH is this reproducibility: no matter how many processes are present in the system the load generated by our tool strictly respects a given load profile. This last characteristic proves to be hard to achieve using simple methods because the system scheduler is supposed to share the resources fairly among running processes. http://krash.ligforge.imag.fr under GNU GPL licence.

5.6. Cache Control

Participants: Guillaume Huard [correspondant], Swann Perarnau.

Cache Control is a Linux kernel module enabling user applications to restrict their memory allocations to a subset of the hardware memory cache. This module reserves and exports available physical memory as virtual devices that can be mmap’d to. It gives to calling processes physical memory using only a subset of the cache (similarly to page coloring). It actually creates cache partitions that can be used simultaneously by a process to control how much cache a data structure can use.
5. Software

5.1. SAFDIS

Contact: Jean-Louis Pazat, Jean-Louis.Pazat@irisa.fr
URL: http://www.irisa.fr/myriads/software/folder.2011-12-13.8949308917/
Status: Version 1.0
License: TBD

Presentation: SAFDIS (Self Adaptation for Distributed Services) is a generic framework allowing the self-adaptation of distributed service based applications within a highly volatile context. Compared to other adaptation frameworks, the main advantages of SAFDIS are its genericity, its distributed nature and the focus on SOAs. SAFDIS is in its final implementation and testing phase within the Myriads team and is being used with a real life use case for emergency services.

The current implementation of SAFDIS is based on a Java OSGi implementation. SAFDIS is written in Java and organized into OSGi bundles. SAFDIS is not tight to any specific operating system and work within any JAVA 1.6 platform. An OSGi implementation is needed (such as the Apache Felix http://felix.apache.org or Equinox eclipse.org/equinox implementations). In order to benefit from the reactive adaptation tools, the Jess engine is also needed as an OSGi bundle (http://www.jessrules.com).

5.2. HOCL-tools

Contact: Cédric Tedeschi, Cedric.Tedeschi@irisa.fr
Status: Version 1.0 to be released
License: TBD

Presentation: HOCL (Higher Order Chemical Language) is a chemical programming language based on the chemical metaphor presented before (see Section 3.5). It was developed for several years within the PARIS team. Within HOCL, following the chemical metaphor, computations can be regarded as chemical reactions, and data can be seen as molecules which participate in these reactions. If a certain condition is held, the reaction will be triggered, thus continuing until it gets inert: no more data can satisfy any computing conditions. To realize this program paradigm, a multiset is implemented to act as a chemical tank, containing necessary data and rules. An HOCL program is then composed of two parts: chemical rule definitions (reaction rules) and multiset definition (data). More specifically, HOCL provides the high order: reaction rules are molecules that can be manipulated like any other molecules. In other words, HOCL programs can manipulate other HOCL programs.

An HOCL compiler was developed using java to execute some chemical programs expressed with HOCL. This compiler is based on the translation of HOCL programs to java code. As a support for service coordination and service adaptation (refer to Section 6.4), we recently extended the HOCL compiler with the support of decentralized workflow execution. Works around the implementation of a distributed multiset gave birth to an underlying layer for this compiler, making it able to deploy HOCL programs transparently over large scale platforms. This last part is currently considered to be interfaced with the current HOCL compiler. All these features are planned to be released under the common name of HOCL-tools.

• Version: 1
5.3. XtreemOS

Contact: Yvon Jégou, Yvon.Jégou@inria.fr
Status: Version 3.0
License: GPL-2/BSD depending on software packages composing the system

Presentation: XtreemOS is a Grid Operating system based on Linux with native support for virtual organizations. Three flavours of XtreemOS are developed for individual PCs, clusters and mobile devices (PDA, notebooks and smartphones). XtreemOS has been developed by the XtreemOS consortium. The third public version of XtreemOS has been released in December 2010.

XtreemOS has been presented at the Contrail summer school, Presqu’île de Giens, France in July 2011 (Y. Jégou), SC’11, Seattle, USA (Y. Jégou, Ch. Morin) in November 2011. XtreemOS software is a set of services developed in Java, C++ and C. XtreemOS cluster version leverages KERRIGHED single system image operating system. A permanent testbed composed of computers provided by several XtreemOS partners has been public since fall 2010. In 2011, XtreemOS has been packaged for the OpenSuse Linux distribution. Ready-to-use XtreemOS virtual machine images have been made available for the community.

Active contributors (from Myriads team): Amine Belhaj, Jérôme Gallard, Rémy Garrigue, Yvon Jégou, Christine Morin, Yann Radenac, Pierre Riteau.

5.4. Contrail Virtual Execution Platform (VEP)

Contact: Yvon Jégou, Yvon.Jégou@inria.fr
URL: http://www.contrail-project.eu
Status: Version 1.0
License: BSD

Presentation: Virtual Execution Platform (VEP) is a Contrail service that sits just above IaaS layer at the service provider end of the Contrail cloud federation. The VEP provides a uniform interface for managing the whole lifecycle of elastic applications on the cloud and hides the details of the IaaS layer to the user. VEP applications are described in OVF (Open Virtualization Format) standard format. Resource usage is controlled by CEE (Constrained Execution Environment) rules which can be derived from SLAs (Service Level Agreement). The VEP integrates a monitoring system where the major events about the application, mainly resource usage, are made available to the user.

The VEP service provides a RESTful interface and can be exploited directly by users on top of the provider IaaS. OpenNebula and OCCI-based IaaS interfaces are currently supported.

Active contributors (from Myriads team): Roberto-Gioacchino Cascella, Florian Dudouet, Piyush Harsh, Yvon Jégou, Christine Morin.

5.5. Snooze

Contact: Christine Morin, Christine.Morin@inria.fr
URL: http://www.irisa.fr/myriads/software/snooze/
Status: Version 1.0
License: TBD
Presentation: Snooze [56], is a novel VM management framework for private clouds which is designed to scale across thousands of nodes. Unlike the existing cloud management frameworks, Snooze utilizes a self-organizing hierarchical architecture and performs distributed VM management. Particularly, VM management tasks are performed by multiple managers, with each manager having only a partial view of the system. Moreover, fault-tolerance is provided at all levels of the hierarchy by replication and integrated leader election algorithm. Consequently, the system is able to self-heal and continue its operation despite system component failures. In addition, VM monitoring is integrated into the framework and a generic scheduling engine exists to support advanced (e.g., consolidation) scheduling policies. Last but not least, a RESTful command line interface (CLI) exists to support virtual cluster (VC) definitions and management (i.e., start, shutdown, destroy, suspend, etc.) as well hierarchy visualization and exporting in GraphML format.

Snooze is fully implemented from scratch in Java and currently comprises approximately 15,000 lines of maintainable abstractions-based code. The leader election algorithm is built on top of the Apache ZooKeeper [59] highly available and reliable coordination system. In order to provide a uniform interface to the underlying hypervisors and support transparent VM monitoring and management, Snooze integrates the libvirt virtualization library.

The first Snooze prototype [41] has been developed and evaluated on Grid5000 experimental testbed.

Active contributors (from Myriads team): Eugen Feller, Christine Morin.

5.6. Resilin

Contact: Christine Morin, Christine.Morin@inria.fr
URL: http://www.irisa.fr/myriads/software/resilin/
Status: Version 0.1
License: TBD

Presentation: Resilin [45] is a system to easily create execution platforms over distributed cloud resources for executing MapReduce computations. Resilin implements the Amazon Elastic MapReduce web service API with resources from other clouds than Amazon EC2, such as private and community clouds. Resilin allows users to perform MapReduce computations on other infrastructures than Amazon EC2, and offers more flexibility: users are free to select different types of virtual machines, different operating systems or newer Hadoop versions. Users only have to submit computations to the service through a web service API, and Resilin takes care of provisioning, configuring and managing cloud-based Hadoop execution platforms, potentially using multiple clouds.

Resilin is implemented in the Python language. It uses the boto library in order to interact with EC2-compatible clouds. Resilin has been evaluated on the Grid’5000 experimental testbed. Our comparison with the Amazon Elastic MapReduce service shows similar performance.

Active contributors (from Myriads team): Pierre Riteau, Christine Morin.

5.7. Saline

Contact: Christine Morin, Christine.Morin@inria.fr
URL: https://www.grid5000.fr/mediawiki/index.php/VMdeploy
Status: Version V1.0 (experimental)
License: BSD
Presentation: Saline (formerly called VMdeploy) is a generic framework to deploy and manage encapsulated user jobs in virtual machines (VMs) at grid level by moving them from one site to another transparently for the encapsulated jobs [58]. Moreover, Saline is non-intrusive and can be used with any non-modified Grid resource management systems (RMSs).

Saline deploys and configures a set of VMs according to the user needs. Then, periodically, Saline takes snapshots of the running VMs and saves them on a dedicated node in an efficient way [57]. In addition, Saline checks the status of the running VMs. If something wrong happens i.e. one or more VMs have failed due to a node failure or the arrival of a higher priority job, Saline redeploys the set of VMs from the latest snapshot taken on new available resources provided by the Grid RMS. Thanks to Saline, the redeployment of the snapshot is done in a transparent way from the encapsulated job point of view.

In its current implementation, Saline is programmed in bash and C. It uses Libvirt in order to create, to snapshot, and to restart the VMs. It means that Saline can deploy and manage KVM and XEN VMs or any other VMs usable with Libvirt. In addition, the architecture of Saline is very modular in order to have a clear and easily extensible code.

Active contributors (from Myriads team): Jérôme Gallard, Christine Morin
OASIS Project-Team

5. Software

5.1. ProActive


url: Proactive Parallel Suite

ProActive is a Java library (Source code under AGPL license) for parallel, distributed, and concurrent computing, also featuring mobility and security in a uniform framework. With a reduced set of simple primitives, ProActive provides a comprehensive API to simplify the programming of applications that are distributed on a Local Area Network (LAN), on cluster of workstations, Clouds, or on Internet Grids.

The library is based on an Active Object pattern that is a uniform way to encapsulate:

- a remotely accessible object,
- a thread,
- an actor with its own script,
- a server of incoming requests,
- a mobile and potentially secure agent.

and has an architecture to inter-operate with (de facto) standards such as:

- Web Service exportation (Apache Axis2 and CXF),
- HTTP transport,
- ssh, rsh, RMI/ssh tunnelling,
- Globus: GT2, GT3, GT4, gsi, Unicore, ARC (NorduGrid)
- LSF, PBS, Sun Grid Engine, OAR, Load Leveler

ProActive is only made of standard Java classes, and requires no changes to the Java Virtual Machine, no preprocessing or compiler modification; programmers write standard Java code. Based on a simple Meta-Object Protocol, the library is itself extensible, making the system open for adaptations and optimisations. ProActive currently uses the RMI Java standard library as default portable transport layer, but others such as Ibis or HTTP can be used instead, in an adaptive way.

ProActive is particularly well-adapted for the development of applications distributed over the Internet, thanks to reuse of sequential code, through polymorphism, automatic future-based synchronisations, migration of activities from one virtual machine to another. The underlying programming model is thus innovative compared to, for instance, the well established MPI programming model.

In order to cope with the requirements of large-scale distributed and heterogeneous systems like the Grid, many features have been incorporated into ProActive, including support for many transport and job submission protocols, GCM component support, graphical visualization interface, object migration, distributed and non-functional exception handling, fault-tolerance and checkpointing mechanisms; file transfer capabilities, a job scheduler, a resource manager able to manage various hosting machines, support for JMX and OSGi capabilities, web service object exposition, an SCA personality, etc.

ProActive is a project of the former ObjectWeb, now OW2 Consortium. OW2 is an international consortium fostering the development of open-source middleware for cutting-edge applications: EAI, e-business, clustering, grid computing, managed services and more. For more information, refer to [5] [42] and to the web pages http://www.objectweb.org and http://proactive.inria.fr/ which list several white papers.
ProActive management, distribution, support, and commercialisation is now ensured by the start-up company ActiveEon (http://www.activeeon.com), in the context of a collaboration with INRIA and UNS.

5.2. Vercors platform

Participants: E. Madelaine, R. Halalai, L. Henrio, A. Savu, M. Alexe.

The Vercors tools (http://www-sop.inria.fr/oasis/Vercors) include front-ends for specifying the architecture and behaviour of components in the form of UML diagrams. We translate these high-level specifications into behavioural models in various formats, and we also transform these models using abstractions. In a final step, abstract models are translated into the input format for various verification toolsets. Currently we mainly use the various analysis modules of the CADP toolset.

- Our main effort this year was based on the development of a quite large case-study, two orders of magnitude larger than our previous experiments. This study was the opportunity to develop new methods for encoding our models using a new combination of CADP formalisms, combining compositional approaches, abstraction techniques, and distributed model-checking [22]. The implementation of these methods in the Vercors tools has started.

We have also been conducting experiments towards the next generation of specification formalism editors for VerCors, using the Papyrus UML-based environment.
5. Software

5.1. DiaSuite: a Development Environment for Sense/Compute/Control Applications

Participants: Charles Consel [correspondent], Benjamin Bertran, Ghislain Deffrasnes, Amélie Marzin, Damien Cassou, Julien Bruneau, Emilie Balland.

Despite much progress, developing a pervasive computing application remains a challenge because of a lack of conceptual frameworks and supporting tools. This challenge involves coping with heterogeneous devices, overcoming the intricacies of distributed systems technologies, working out an architecture for the application, encoding it in a program, writing specific code to test the application, and finally deploying it.

DiaSuite is a suite of tools covering the development life-cycle of a pervasive computing application:

- **Defining an application area.** First, an expert defines a catalog of entities, whether hardware or software, that are specific to a target area. These entities serve as building blocks to develop applications in this area. They are gathered in a taxonomy definition, written in the taxonomy layer of the Diaspec language.

- **Designing an application.** Given a taxonomy, the architect can design and structure applications. To do so, the Diaspec language provides an application design layer [33]. This layer is dedicated to an architectural pattern commonly used in the pervasive computing domain [24]. Describing the architecture application allows to further model a pervasive computing system, making explicit its functional decomposition.

- **Implementing an application.** We leverage the taxonomy definition and the architecture description to provide dedicated support to both the entity and the application developers. This support takes the form of a Java programming framework, generated by the Diagen compiler. The generated programming framework precisely guides the developer with respect to the taxonomy definition and the architecture description. It consists of high-level operations to discover entities and interact with both entities and application components. In doing so, it abstracts away from the underlying distributed technologies, providing further separation of concerns.

- **Testing an application.** Diagen generates a simulation support to test pervasive computing applications before their actual deployment. An application is simulated in the Diasim tool, without requiring any code modification. Diasim provides an editor to define simulation scenarios and a 2D-renderer to monitor the simulated application. Furthermore, simulated and actual entities can be mixed. This hybrid simulation enables an application to migrate incrementally to an actual environment.

- **Deploying a system.** Finally, the system administrator deploys the pervasive computing system. To this end, a distributed systems technology is selected. We have developed a back-end that currently targets the following technologies: Web Services, RMI, SIP and OSGI. This targeting is transparent for the application code. The variety of these target technologies demonstrates that our development approach separates concerns into well-defined layers.

This development cycle is summarized in the Figure 1.

See also the web page [http://diasuite.inria.fr](http://diasuite.inria.fr).
5.1.1. DiaSpec: a Domain-Specific Language for Networked Entities

The core of the DIA SUITE development environment is the domain specific language called DIA SPEC and its compiler DIA GEN:

- **DIA SPEC** is composed of two layers:
  - The **Taxonomy Layer** allows the declaration of entities that are relevant to the target application area. An entity consists of sensing capabilities, producing data, and actuating capabilities, providing actions. Accordingly, an entity description declares a data source for each one of its sensing capabilities. As well, an actuating capability corresponds to a set of method declarations. An entity declaration also includes attributes, characterizing properties of entity instances. Entity declarations are organized hierarchically allowing entity classes to inherit attributes, sources and actions. A taxonomy allows separation of concerns in that the expert can focus on the concerns of cataloging area-specific entities. The entity developer is concerned about mapping a taxonomical description into an actual entity, and the application developer concentrates on the application logic.
  - The **Architecture Layer** is based on an architectural pattern commonly used in the pervasive computing domain [24]. It consists of context components fueled by sensing entities. These components process gathered data to make them amenable to the application needs. Context data are then passed to controller components that trigger actions on entities. Using an architecture description enables the key components of an application to be identified, allowing their implementation to evolve with the requirements (e.g., varying light management implementations in a controller component to optimize energy consumption).

- **DIA GEN** is the DIA SPEC compiler that performs both static and runtime verifications over DIA SPEC declarations and produces a dedicated programming framework that guides and eases the implementation of components. The generated framework is independent of the underlying distributed technology. As of today, DIA GEN supports multiple targets: Local, RMI, SIP, Web Services and OSGI.

5.1.2. DiaSim: a Parametrized Simulator for Pervasive Computing Applications

![DiaSUITE Development Cycle](image)
Pervasive computing applications involve both software and integration concerns. This situation is problematic for testing pervasive computing applications because it requires acquiring, testing and interfacing a variety of software and hardware entities. This process can rapidly become costly and time-consuming when the target environment involves many entities.

To ease the testing of pervasive applications, we are developing a simulator for pervasive computing applications: DIA SIM. To cope with widely heterogeneous entities, DIA SIM is parameterized with respect to a DIA SPEC specification describing a target pervasive computing environment. This description is used to generate with DIA GEN both a programming framework to develop the simulation logic and an emulation layer to execute applications. Furthermore, a simulation renderer is coupled to DIA SIM to allow a simulated pervasive system to be visually monitored and debugged. The simulation renderer is illustrated in Figure 2.

5.2. DiaSuiteBox: an Open Service Platform
Participants: Benjamin Bertran [correspondent], Julien Bruneau, Charles Consel, Emilie Balland.

The DiaSuiteBox platform runs an open-ended set of applications leveraging a range of appliances and web services. Our solution consists of a dedicated development environment, a certifying application store, and a lightweight runtime platform. This solution is based on the DIA SUITE project.

The DiaSuiteBox platform can be embedded in a small plug-computer. This box can be easily deployed, runs silently, and has a reduced energy consumption. Thanks to the application store and the developer community, the platform is fed by a full offer of new innovative applications. During the submission process, an application is automatically analyzed and checked in order to be certified. The user is ensured of the behavior of its applications are innocuous and correct beside the provided information. This box relies on several technology standards like UPnP, Bluetooth, USB, etc. As shown in Figure 3, this platform can be easily extended by plugging appliances directly on the box or by connecting devices on the local network.

See also the web page http://diabox.inria.fr.

5.3. Pantagruel: a Visual Domain-Specific Language for Ubiquitous Computing
Participants: Ghislain Deffrasnes [correspondent], Julien Mercadal, Charles Consel.
Figure 3. DiaSuiteBox platform architecture

Figure 4. A screenshot of the Pantagruel graphical editor
Pantagruel aims at easing the description of an orchestration logic between networked entities of a pervasive environment. First, the developer defines a taxonomy of entities that compose the environment. This step provides an abstraction of the entities capabilities and functionalities. Second, the developer defines the orchestration logic in terms of rules. To facilitate its programming, we provide a visual domain-specific language based on the sensor-controller-actuator paradigm. An example of a visual orchestration is given in Figure 4 where a shower automatically runs at the right temperature when someone enters the bathroom and closes the door.

Pantagruel brings a high-level layer intended to complement existing tools in the activity of safe orchestration logic description, allowing novice-programmers to prototype pervasive applications. The Pantagruel compiler generates code compliant with the D1ASUITE toolset. Pantagruel is being completed by tools aimed at verifying safety properties like termination and reachability.

See also the web page http://phoenix.inria.fr/software/pantagruel.
5. Software

5.1. ns-3

**Participant:** Daniel Camara [correspondant].

ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use. ns-3 includes a solid event-driven simulation core as well as an object framework focused on simulation configuration and event tracing, a set of solid 802.11 MAC and PHY models, an IPv4, UDP, and TCP stack and support for nsc (integration of Linux and BSD TCP/IP network stacks).

See also the web page [http://www.nsnam.org](http://www.nsnam.org).

- **Version:** ns-3.7
- **Keywords:** networking event-driven simulation
- **License:** GPL (GPLv2)
- **Type of human computer interaction:** programmation C++/python, No GUI
- **OS/Middleware:** Linux, cygwin, osX
- **Required library or software:** standard C++ library: GPLv2
- **Programming language:** C++, python
- **Documentation:** doxygen

5.2. EphPub

**Participants:** Mohamed Ali Kaafar [correspondant], Claude Castelluccia.

EphPub (Ephemeral Publishing) (previously called EphCom) implements a novel key storage mechanism for time-bounded content, that relies on the caching mechanism of the Domain Name System (DNS). Features of EphPub include: EphPub exploits the fact that DNS servers temporarily cache the response to a recursive DNS query for potential further requests. EphPub provides higher security than Vanish, as it is immune to Sybil attacks. EphPub is easily deployable and does not require any additional infrastructure, such as Distributed Hash Tables. EphPub comes with high usability as it does not require users to install and execute any extra additional software. EphPub lets users define data lifetime with high granularity. We provide EphPub as an Android Application to provide ephemeral exchanged SMS, emails, etc. and as a Firefox or Thunderbird extensions so as to support ephemeral publication of any online document.

For more details about the different software products, see [http://planete.inrialpes.fr/projects/ephemeral-publication/](http://planete.inrialpes.fr/projects/ephemeral-publication/).

- **Version:** v0.1.2-beta
- **ACM:** K.4.1
- **AMS:** 94Axx
- **Keywords:** Ephemeral communications, Right to Forget, Future Internet Architecture, Privacy
- **Software benefit:** We provide a Firefox Extension that easily allows users to manage disappearing emails. We also provide a command-line tool to manage disappearing files.
- **APP:** Under APP deposit internal process
- **License:** GPL
- **Type of human computer interaction:** Firefox extension + Unix Console
- **OS/Middleware:** Firefox under any OS
- **Required library or software:** Python Ext
- **Programming language:** Python
- **Documentation:** No detailed documentation has been released so far. A detailed howto can be consulted however at: [http://code.google.com/p/disappearingdata/source/browse/wiki/EphCOM_Firefox_Extension.wiki?r=77](http://code.google.com/p/disappearingdata/source/browse/wiki/EphCOM_Firefox_Extension.wiki?r=77)
5.3. Username Tester

**Participants:** Claude Castelluccia [correspondant], Mohamed Ali Kaafar, Daniele Perito.

Usernames are ubiquitous on the Internet. Almost every web site uses them to identify its users and, by design, they are unique within each service. In web services that have millions or hundreds of millions of users, it might become difficult to find a username that has not already been taken. For instance, you might have experienced that a specific username you wanted was already taken. This phenomenon drives users to choose increasingly complex and unique usernames.

We built a tool to estimate how unique and linkable usernames are and made it available on this page for you to check. For example, according to our tool, “ladygaga” or “12345678” only carry 24 and 17 bits of entropy, respectively. They are therefore not likely to be unique on the Internet. On the other hand, usernames such as “pdjkwert” or “yourejerky” carry about 40 bits of entropy and are therefore very good identifiers.

Type your username (for example “zorro1982” or “dan.perito”) to discover how unique it is. This tool can help you to select an username that has low entropy and can’t be used to track you on the Internet.

Alternatively, try typing two usernames separated by a space. The tool will give an estimation on whether the two usernames are linkable. The tool is accessible here: [http://planete.inrialpes.fr/projects/how-unique-are-your-usernames/](http://planete.inrialpes.fr/projects/how-unique-are-your-usernames/)

5.4. DroidMonitor

**Participants:** Claude Castelluccia [correspondant], Mohamed Ali Kaafar, Anasthesia Fedane.

In nowadays world the technological progress evolves very quickly. There are more and more new devices, fully equipped with the latest innovations. The question is: do we adopt our main privacy concerns according to these new technologies as quickly as they grow and become widely available for us?…

We developed a novel tool, private data leakage monitoring tool, DroidMonitor. It aims to serve as an educational tool for regular Android Smartphones users to make them aware of existing privacy threats while they are using Location-Based Services. It can be downloaded here: [http://planete.inrialpes.fr/android-privacy/](http://planete.inrialpes.fr/android-privacy/)

5.5. NEPI

**Participants:** Thierry Turletti [correspondant], Alina Quereilhac, Claudio Freire.

NEPI stands for Network Experimentation Programming Interface. NEPI implements a new experiment plane used to perform ns-3 simulations, planetlab and emulation experiments, and, more generally, any experimentation tool used for networking research. Its goal is to make it easier for experimenters to describe the network topology and the configuration parameters, to specify trace collection information, to deploy and monitor experiments, and, finally, collect experiment trace data into a central datastore. NEPI is a python API (with an implementation of that API) to perform all the above-mentioned tasks and allows users to access these features through a simple yet powerful graphical user interface called NEF. During the year 2011 we improved the robustness in the experiment control scheme, and we added support for new experimentation environments. We released and registered a second version of the NEPI software (IDDN.FR.001.06003.001.S.A.2010.000.10600). Details on the improvements made can be found in [48].

See also the web page [http://nepihome.org](http://nepihome.org).

- Version: 1.0
- ACM: C.2.2, C.2.4
- Keywords: networking experimentation
- License: GPL (2)
- Type of human computer interaction: python library, QT GUI
- OS/Middelware: Linux
- Programming language: python
5.6. Reference implementation for SFA Federation of experimental testbeds

Participants: Thierry Parmentelat [correspondant], Baris Metin, Julien Tribino.

We are codeveloping with Princeton University a reference implementation for the Testbed-Federation architecture known as SFA for Slice-based Federation Architecture. During 2011 we have focused on the maturation of the SFA codebase, with several objectives in mind, better interoperability between the PlanetLab world and the EmuLab, a more generic shelter that other testbeds can easily leverage in order to come up with their own SFA-compliant wrapper and support for ‘reservable’ mode, which breaks the usual best-effort PlanetLab model. For more details about this contribution see section

See also the web page http://planet-lab.eu

- Version: myplc-5.0-rc26
- Keywords: networking testbed virtual machines
- License: Various Open Source Licences
- Type of human computer interaction: Web-UI, XMLRPC-based API, Qt-based graphical client
- OS/Middelware: Linux-Fedora
- Required library or software: Fedora-14 for the infrastructure side; the software comes with a complete software suite for the testbed nodes
- Programming languages: primarily python, C, ocaml
- Documentation: most crucial module plcapi is self-documented using a local format & related tool. See e.g. https://www.planet-lab.eu/db/doc/PLCAPI.php
- Codebase: http://git.onelab.eu

5.7. MultiCast Library Version 3

Participant: Vincent Roca [correspondant].

MultiCast Library Version 3 is an implementation of the ALC (Asynchronous Layered Coding) and NORM (NACK-Oriented Reliable Multicast Protocol) content delivery Protocols, and of the FLUTE/ALC file transfer application. This software is an implementation of the large scale content distribution protocols standardized by the RMT (Reliable Multicast Transport) IETF working group and adopted by several standardization organizations, in particular 3GPP for the MBMS (Multimedia Broadcast/Multicast Service), and DVB for the CBMS (Convergence of Broadcast and Mobile Services). Our software is used in operational, commercial environments, essentially in the satellite broadcasting area and for file delivery over the DVB-H system where FLUTE/ALC has become a key component. See http://planete-bcast.inrialpes.fr/ for more information.

5.8. OpenFEC.org: because open, free AL-FEC codes and codecs matter

Participants: Vincent Roca [correspondant], Jonathan Detchart [engineer], Ferdaouss Mattoussi [PhD student].

The goals of the OpenFEC.org http://openfec.org are:

- to share IPR-free, open, AL-FEC codes, to share high performance, ready-to-use, open, free, C-language, software codecs and to share versatile and automated performance evaluation environments.

This project can be useful to users who do not want to know the details of AL-FEC schemes but do need to use one of them in the software they are designing, or by users who want to test new codes or new encoding or decoding techniques, and who do know what they are doing and are looking for, or by users who need to do extensive tests for certain AL-FEC schemes in a given use-case, with a well defined channel model.
5.9. BitHoc

Participants: Chadi Barakat [correspondant], Thierry Turletti, Amir Krifa.

BitHoc (BitTorrent for wireless ad hoc networks) enables content sharing among spontaneous communities of mobile users using wireless multi-hop connections. It is an open source software developed under the GPLv3 licence. A first version of BitHoc has been made public. We want BitHoc to be the real testbed over which we evaluate our solutions for the support and optimization of file sharing in a mobile wireless environment where the existence of an infrastructure is not needed. The proposed BitHoc architecture includes two principal components: a membership management service and a content sharing service. In its current form it is composed of PDAs and smartphones equipped with WIFI adapters and Windows Mobile 6 operating system.

See also the web page http://planete.inria.fr/bithoc

- Version: 1.2
- Keywords: Tracker-less BitTorrent for mobile Ad Hoc networks
- License: GPL (GPLv3)
- Type of human computer interaction: Windows Mobile 6 GUI
- OS/Middleware: Windows Mobile 6
- Required library or software: OpenSSL (http://www.openssl.org/, GPL), C++ Sockets (http://www.alhem.net/Sockets/, GPL)
- Programming languages: C++, C#
- Documentation: doxygen

5.10. TICP

Participant: Chadi Barakat [correspondant].

TICP is a TCP-friendly reliable transport protocol to collect information from a large number of network entities. The protocol does not impose any constraint on the nature of the collected information: availability of network entities, statistics on hosts and routers, quality of reception in a multicast session, weather monitoring, etc. TICP ensures two main things: (i) the information to collect arrives entirely and correctly to the collector where it is stored and forwarded to upper layers, and (ii) the implosion at the collector and the congestion of the network are avoided by controlling the rate of sending probes. The congestion control part of TICP is designed with the main objective to be friendly with applications using TCP. Experimental results show that TICP can achieve better performance than using parallel TCP connections for the data collection. The code of TICP is available upon request, it is an open source software under the GPLv3 licence.

See also the web page http://planete.inria.fr/ticp/

- Version: 1.0
- Keywords: Information Collection, Congestion and Error Control
- License: GPL (GPLv3)
- Type of human computer interaction: XML file
- OS/Middleware: Linux/Unix
- Required library or software: C/C++ Sockets
- Programming languages: C/C++
- Documentation: Text
5.11. Experimentation Software

WisMon
WisMon is a Wireless Statistical Monitoring tool that generates real-time statistics from a unified list of packets, which come from possible different probes. This tool fulfills a gap on the wireless experimental field: it provides physical parameters on realtime for evaluation during the experiment, records the data for further processing and builds a single view of the whole wireless communication channel environment. WisMon is available as open source under the Cecill license, at http://planete.inria.fr/software/WisMon/.

WEX Toolbox
The Wireless Experimentation (WEX) Toolbox aims to set up, run and make easier the analysis of wireless experiments. It is a flexible and scalable open-source set of tools that covers all the experimentation steps, from the definition of the experiment scenario to the storage and analysis of results. Sources and binaries of the WEX Toolbox are available under the GPLv2 licence at https://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/WEXToolkit. WEX Toolbox includes the CrunchXML utility, which aims to make easier the running and the analysis of wireless experimentations. In a nutshell, it implements an efficient synchronization and merging algorithm, which takes XML (or PDML) input trace files generated by multiple probes, and stores only the packets fields that have been marked as relevant by the user in a MySQL database –original pcap traces should be first formatted in XML using wireshark. These operations are done in a smart way to balance the CPU resources between the central server (where the database is created) and the different probes (i.e., PC stations where the capture traces are located). CrunchXML is available under the GNU General Public License v2 at http://twiki-sop.inria.fr/twiki/bin/view/Projets/Planete/CrunchXML.

WiMAX ns-3
This simulation module for the ns-3 network simulator is based on the IEEE 802.16-2004 standard. It implements the PMP topology with TDD mode and aims to provide detailed and standard compliant implementation of the standard, supporting important features including QoS scheduling services, bandwidth management, uplink request/grant scheduling and the OFDM PHY layer. The module is available under the GNU General Public License at http://code.nsnam.org/iamine/ns-3-wimax. It will be included in the official 3.8v release of ns-3.

MonLab
Monitoring Lab is a platform for the emulation and monitoring of traffic in virtual ISP networks. It is supported by the FP7 ECODE project and is available for download at the web page of the tool http://planete.inria.fr/MonLab/ under the terms of the GPL licence. MonLab presents a new approach for the emulation of Internet traffic and for its monitoring across the different routers of the emulated ISP network. In its current version, the traffic is sampled at the packet level in each router of the platform, then monitored at the flow level. We put at the disposal of users real traffic emulation facilities coupled to a set of libraries and tools capable of Cisco NetFlow data export, collection and analysis. Our aim is to enable running and evaluating advanced applications for network wide traffic monitoring and optimization. The development of such applications is out of the scope of this research. We believe that the framework we are proposing can play a significant role in the systematic evaluation and experimentation of these applications’ algorithms. Among the direct candidates figure algorithms for traffic engineering and distributed anomaly detection. Furthermore, methods for placing monitors, sampling traffic, coordinating monitors, and inverting sampling traffic will find in our platform a valuable tool for experimentation.

MobiTrade
MobiTrade is the ns-3 and Android implementation of our solution in [41] for trading content between wireless devices. The application provides a utility driven trading system for efficient content dissemination on top of a disruption tolerant network. While simple tit-for-tat (TFT) mechanisms can force nodes to give one to get one, dealing with the inherent tendency of peers to
take much but give back little, they can quickly lead to deadlocks when some (or most) of interesting content must be somehow fetched across the network. To resolve this, MobiTrade proposes a trading mechanism that allows a node (merchant) to buy, store, and carry content for other nodes (its clients) so that it can later trade it for content it is personally interested in. To exploit this extra degree of freedom, MobiTrade nodes continuously profile the type of content requested and the collaboration level of encountered devices. An appropriate utility function is then used to collect an optimal inventory that maximizes the expected value of stored content for future encounters, matched to the observed mobility patterns, interest patterns, and collaboration levels of encountered nodes. See also http://planete.inria.fr/MobiTrade.
4. Software

4.1. ASPIRE TDT

Participants: Nathalie MITTON, Loïc Schmidt [correspondant], David Simplot-Ryl.

Tag Data Translation (TDT) is an EPCGlobal Inc. standard allowing the translation of identifiers EPC in different representation standard. EPCGlobal standards deal only EPC identifiers. We have extended it to other RFID GS1 and smartcard standards (as ISO 14443 or 15693 and EAN/UPC).


- Version: version 0.5

4.2. ASPIRE ALECC

Participants: Nathalie MITTON [correspondant], David Simplot-Ryl, Lei Zhang.

According to the feedback of several RFID application SMEs. They are more likely to accept a light and efficient ALE scheme which only includes the most-used basic modules defined by EPC standard. They desire that such light scheme can be encapsulated and be flexibly used to establish their own RFID application.

The AspireALECC scheme is encapsulated in jar and aims to supply an easy and efficient framework for developers to realize the most used basic operations defined by the EPC ALECC standard.

- Version: 1.0

4.3. EPC TAG CONVERTER

Participants: Roudy Dagher [correspondant], Nathalie MITTON, Loïc Schmidt, David Simplot-Ryl.

This module is an EPC-compliant module that aims to convert any tag format into an EPC tag understandable by the middleware.

- Version: 1.0

4.4. EPC TAG GENERATOR

Participants: Roudy Dagher [correspondant], Nathalie MITTON, Loïc Schmidt, David Simplot-Ryl.

This module aims to generate tag ids in hexadecimal format in order to validate the EPC grouping and filtering engines of the ALE.


- Version: 1.0

4.5. Eve - TCF

Participants: Arnaud Fontaine, Isabelle Simplot-Ryl [correspondant].

Verification of transitive control flow policies on JavaCard 2.x bytecode. Control flow policies expressed using a DSL language are embedded in JavaCard packages (CAP files) using Eve-TCF convert tool. Control flow policies are then statically verified on-device at loading-time thanks to an embedded verifier (designed for smart cards in Eve-TCF). Eve-TCF (Embedded Verifier for Transitive Control Flow) also contains an off-device (i.e. PC tool) to simulate on-device loading process of JavaCard 2.x platforms with GlobalPlatform 2.x installed.

- Version: v2.0 - 07/09/2011
4.6. GOLIATH 1.0  
**Participants:** Tony Ducrocq, Nathalie MITTON, David Simplot-Ryl [correspondant], Julien Vandaele.  
GOLIATH (Generic Optimized LIghtweight communication stack for Ambient TecHnologies) is a full protocol stack for wireless sensor networks.

4.7. JITS  
**Participants:** Geoffroy Cogniaux, Gilles GRIMAUD [correspondant].  
Initial goal of Java was to allow high level software development on small devices. Eventually it founds success and promotion with software deployment on the Web, and more recently as a solution for huge enterprise servers and massive parallel computing. Today small targets are still supported, but with dedicated (Java-like) APIs and VMs. These specific technologies dramatically restrain the context in which Java applications can be deployed.

JITS focuses on these technologies and on enhancements to allow the use of a real Java Runtime Environment and a Java Virtual Machine everywhere by targeting tiny devices such as SmartCards. These devices usually don’t use a Virtual Machine layer over an OS, but expect the Virtual Machine to be the OS. This is possible thanks to the JVM features which can be presented as a specific hardware abstraction for most of them. See also the web page [http://jits.gforge.inria.fr/](http://jits.gforge.inria.fr/)

- Version: 1

4.8. Light ALE  
**Participants:** Roudy Dagher [correspondant], Nathalie Mitton, Loic Schmidt, David Simplot-Ryl.  
In order to provide minimal inventory services, at interface level, subsets of the Reading and the Logical Reader APIs are implemented:

1. **Immediate mode:** sufficient for user-triggered inventory.
2. **Fixed reader configuration:** only some properties (Power, Session and InitialQ) can be updated.

The ALE engine manages tag grouping and filtering according to EPC standard patterns in input ECSpec objects. A lightweight custom CODEC was developed as well, in order to decode tag IDs using binary format (array of bytes) and in a garbage-free fashion. A filter engine is also made available for software filtering of tags. This leaves the choice for Reader Connectors to choose the best tradeoff between software and hardware filtering. Note that, because of the Java CDC constraint, the ECSpec and ECReports classes and subclasses were written manually despite of automatic generation from XSD files. See also the web page [http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation.Filtering%26Collection/EmbeddedALE](http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation.Filtering%26Collection/EmbeddedALE)

- Version: 1.0

4.9. Light RP  
**Participants:** Roudy Dagher [correspondant], Nathalie Mitton, Loic Schmidt, David Simplot-Ryl.  
This wrapper defines the Reader Protocol interface classes that are used to dialog with an RP-compliant reader device. Based on each vendor-specific driver, two implementations were developed in order to provide minimal required services (inventory). Note that the communication with the Reader Device is done locally and directly via method calls. This avoids overhead when using MTB layers for message bindings. See also the web page [http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation/LightRP](http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation/LightRP)

- Version: 1.0

4.10. NFC Light ALE  
**Participants:** Nathalie Mitton [correspondant], Loic Schmidt, David Simplot-Ryl, Lei Zhang.
In order to provide minimal inventory services, at interface level, subsets of the Reading and the Logical Reader APIs are implemented: (I) Immediate mode: sufficient for user-triggered inventory. ii Fixed readers configuration: only some properties (Power, Session and InitialQ) can be updated.

The ALE engine manages tag grouping and filtering according to EPC standard patterns in input ECSpec objects. A lightweight custom CODEC was developed as well, in order to decode tag IDs using binary format (array of bytes) and in a garbage-free fashion. A filter engine is also made available for software filtering of tags. This leaves the choice for Reader Connectors to choose the best tradeoff between software and hardware filtering. Note that, because of the Java CDC constraint, the ECSpec and ECReports classes and subclasses were written manually despite of automatic generation from XSD files. This package has been developed for NFC connection on a mobile phone.

- Version: 1.0

4.11. RFID Tag Searcher

**Participants:** Roudy Dagher [correspondant], Nathalie Mitton, Loïc Schmidt, David Simplot-Ryl.

The objective is to use the PDA to look for an item in a given neighborhood. The user would be notified of item proximity via the change of the beep frequency.

Tag Searcher is composed of three main modules:

- The Reader interface is an abstraction of the required services for searching for a tag at a given RF power. The wrappers for CAEN and Intermec readers are straightforward.
- The Ticker class represents a periodic thread that beeps periodically using the standard java call java.awt.Toolkit.getDefaultToolkit().beep(). The ticker’s period is synchronized inside the PeriodSemaphore class.
- The Scanner class is able to scan for a tag ID and update the Ticker period according to the RF power at which the tag was observed: observation at a small RF power leads to a small tick period, and vice versa.

After testing on both PDAs (i.e. Psion, Intermec), a standalone application with an IHM was developed. It is able to retrieve a list (from a local file or an URL) so that the user choose the item to search for. See also the web page [http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation.EmbeddedTools/TagSearcher](http://wiki.aspire.ow2.org/xwiki/bin/view/Main.Documentation.EmbeddedTools/TagSearcher).

- Version: 1.0

4.12. SINGLE

**Participants:** Tony Ducrocq, Nathalie Mitton, David Simplot-Ryl [correspondant].

SINGLE pour Simple IN-door Geo-Localization systEm est une application pour réseaux de capteurs permettant la localisation géographique de capteurs sans fils dans un environnement intérieur.

See also the web page [http://www.senslab.info/](http://www.senslab.info/).

- Version: 1.0
RAP Project-Team (section vide)
5. Software

5.1. Coccinelle

Participants: Julia Lawall [correspondent], Gilles Muller [correspondent], Gaël Thomas, Suman Saha, Arie Middlekoop.

Coccinelle is a program matching and transformation engine which provides the language SmPL (Semantic Patch Language) for specifying desired matches and transformations in C code. Coccinelle was initially targeted towards performing collateral evolutions in Linux. Such evolutions comprise the changes that are needed in client code in response to evolutions in library APIs, and may include modifications such as renaming a function, adding a function argument whose value is somehow context-dependent, and reorganizing a data structure.

Beyond collateral evolutions, Coccinelle has been successfully used for finding and fixing bugs in systems code. One of the main recent results is an extensive study of bugs in Linux 2.6 [51] that has permitted us to demonstrate that the quality of code has been improving over the last six years, even though the code size has more than doubled.


5.2. Telex

Participants: Marc Shapiro [correspondent], Lamia Benmouffok, Pierre Sutra, Pierpaolo Cincilla.

Developing write-sharing applications is challenging. Developers must deal with difficult problems such as managing distributed state, disconnection, and conflicts. Telex is an application-independent platform to ease development and to provide guarantees. Telex is guided by application-provided parameters: actions (operations) and constraints (concurrency control statements). Telex takes care of replication and persistence, drives application progress, and ensures that replicas eventually agree on a correct, common state. Telex supports partial replication, i.e., sites only receive operations they are interested in. The main data structure of Telex is a large, replicated, highly dynamic graph; we discuss the engineering trade-offs for such a graph and our solutions. Our novel agreement protocol runs Telex ensures, in the background, that replicas converge to a safe state. We conducted an experimental evaluation of the Telex based on a cooperative calendar application and on benchmarks.

We report on application experience, building a collaborative application for model-oriented software engineering above Telex, in SAC 2011 [50]. Future work includes extending Telex to cloud computing, opportunistic mobile networks, and real-time collaboration, within several ANR projects: PROSE (Section 7.1.5), STREAMS (Section 7.1.4) and ConcoRDanT (Section 7.1.3).

The code is freely available on http://gforge.inria.fr/ under a BSD license.

5.3. Treedoc

Participants: Marc Shapiro [correspondent], Marek Zawirski.

A Commutative Replicated Data Type (CRDT) is one where all concurrent operations commute. The replicas of a CRDT converge automatically, without complex concurrency control. We designed and developed a novel CRDT design for cooperative text editing, called Treedoc. It is designed over a dense identifier space based on a binary trees. Treedoc also includes an innovative garbage collection algorithm based on tree rebalancing. In the best case, Treedoc incurs no overhead with respect to a linear text buffer. The implementation has been validated with performance measurements, based on real traces of social text editing in Wikipedia and SVN.
Work in 2010 has focused on studying large-scale garbage collection for Treedoc, and design improvements. Future work includes engineering a large-scale collaborative Wiki, and studying CRDTs more generally. This is the subject the PROSE, STREAMS and ConcoRDanT ANR projects (Sections 7.1.5, 7.1.4 and 7.1.3 respectively).

The code is freely available on http://gforge.inria.fr/ under a BSD license.

5.4. VMKit and .Net runtimes for LLVM

Participants: Harris Bakiras, Bertil Folliot [correspondent], Julia Lawall, Jean-Pierre Lozi, Gaël Thomas [correspondent], Gilles Muller, Thomas Preud’homme.

Many systems research projects now target managed runtime environments (MRE) because they provide better productivity and safety compared to native environments. Still, developing and optimizing an MRE is a tedious task that requires many years of development. Although MREs share some common functionalities, such as a Just In Time Compiler or a Garbage Collector, this opportunity for sharing has not been yet exploited in implementing MREs. We are working on VMKit, a first attempt to build a common substrate that eases the development and experimentation of high-level MREs and systems mechanisms. VMKit has been successfully used to build two MREs, a Java Virtual Machine and a Common Language Runtime, as well as a new system mechanism that provides better security in the context of service-oriented architectures.

VMKit project is an implementation of a JVM and a CLI Virtual Machines (Microsoft .NET is an implementation of the CLI) using the LLVM compiler framework and the MMTk garbage collectors. The JVM, called J3, executes real-world applications such as Tomcat, Felix or Eclipse and the DaCapo benchmark. It uses the GNU Classpath project for the base classes. The CLI implementation, called N3, is in early stages but can execute simple applications and the “pnetmark” benchmark. It uses the pnetlib project or Mono as its core library. The VMKit VMs compare in performance with industrial and top open-source VMs on CPU-intensive applications. VMKit is publicly available under the LLVM license.

http://vmkit.llvm.org/
5. Software

5.1. CloudWeaver suite

Participants: Paulo Gonçalves, Guilherme Koslovski, Fabienne Anhalt.

The following list of softwares, whose development was initiated at RESO, constitutes the main outcome of the research work delivered by Guilherme Koslovski (PhD, July 2011) and Fabienne Anhalt (PhD, July 2011). These products are also part of the technological transfer to Lyatiss (headed by Pascale Vicat-Blanc); embedded in the CloudWeaver Suite, they implement the solutions for virtual resources orchestration and infrastructure services.

- VXAlloc Dynamic allocation of virtual resources (Patent INPI:10/01626, 2010, Lyatiss, INRIA, ENS Lyon)
- HiperNet Automatic configuration of virtual networks, by programming virtual routers and configuring virtual links according to service requirements (APPcode: IDDN.FR.001.260010.000.S.P.2009.000.10700, 2009, Lyatiss, INRIA ENS Lyon)
- VXDL parser Interpretation and XML traduction of virtual infrastructures specifications (APPcode: IDDN.FR.001.260009.000.S.P.2009.000.10800)

Due to non-disclosure agreement between INRIA and Lyatiss, access to these software is now submitted to patent restriction.

5.2. Queueing-systems

Participant: Thomas Begin.

This tool aims at providing a simple web based interface to promote the use of our proposed solutions to numerically solve classical queueing systems. It currently implements the solution to get the distribution for the number of customers along with customary performance parameters for a queue with multiple servers, general arrivals, exponential services and a possibly finite buffer, (i.e., $Ph/M/c/N$-like queue). The steady-state solution to this queue is based on a simple and stable recurrence and was performed in collaboration with Pr. Brandwajn (UCSC). We will include new features and new models to this tool in the near future. Associated URL is: [http://queueing-systems.ens-lyon.fr](http://queueing-systems.ens-lyon.fr)

5.3. ECOFEN simulation framework

Participants: Anne-Cecile Orgerie, Laurent Lefevre.

The problem when evaluating new network architectures and protocols is that large testbed platforms are really expensive and difficult to manage. That is why we have designed ECOFEN whose user’s entries are the network topology and traffic. Based on configurable measurements of different network component (routers, switches, NICs, etc.), it provides the power consumption of the overall network including the end-hosts as well as the power consumption of each equipment over time. The ECOFEN simulator supports green network leverages such as Adaptive Link Rate and on/off. The aim of ECOFEN is to compute and expose the energy consumed by a network under a given traffic. Firstly based on NS2 and now developed on NS3, this simulator has been made in collaboration with Dino Lopez-Pacheco [29].
4. Software

4.1. Moose

Participants: Stéphane Ducasse [correspondant], Usman Bhatti, Andre Hora, Nicolas Anquetil, Cyrille Delaunay, Jannik Laval, Tudor Gîrba [University of Bern].

Web: http://www.moosetechnology.org/

The platform. Moose is a language-independent environment for reverse- and re-engineering complex software systems. Moose provides a set of services including a common meta-model, metrics evaluation and visualization, a model repository, and generic GUI support for querying, browsing and grouping. The development of Moose began at the Software Composition Group in 1997, and is currently contributed to and used by researchers in at least seven European universities. Moose offers an extensible meta-described metamodel, a query engine, a metric engine and several visualizations. Moose is currently in its fourth release and comprises 55,000 lines of code in 700 classes.

The RMoD team is currently the main maintainer of the Moose platform. There are 200 publications (journal, international conferences, PhD theses) based on execution or use of the Moose environment.

Here is the self-assessment of the team effort following the grid given at http://www.inria.fr/institut/organisation/instances/commission-d-evaluation .

- (A5) Audience : 5 – Moose is used by several research groups, a consulting company, and some companies using it in ad-hoc ways.
- (SO4) Software originality : 4 – Moose aggregates the last results of the teams that use it.
- (SM3) Software Maturity : 3 – Moose is developed since 1996 and got two main redesign phases.
- (EM4) Evolution and Maintenance : 4 – Moose will be used as a foundation of our start up so its maintenance is planned.
- (SDL4) Software Distribution and Licensing : 4 – BSD
- (OC) Own Contribution : (Design/Architecture)DA-4, (Coding/Debugging)-4, (Maintenance/Support)-4, (Team/Project Management)-3

4.2. Pharo

Participants: Stéphane Ducasse, Marcus Denker [correspondant], Damien Pollet, Mariano Martinez-Peck, Veronica Uquillas-Gomez, Igor Stasenko.

Web: http://www.pharo-project.org/

The platform. Pharo is a new open-source Smalltalk-inspired language and environment. It provides a platform for innovative development both in industry and research. By providing a stable and small core system, excellent developer tools, and maintained releases, Pharo’s goal is to be a platform to build and deploy mission critical Smalltalk applications.

The first stable version, Pharo 1.0, was released in 2010. The development of Pharo accelerated in 2011: Version 1.2 and 1.3 have been released, the development branch (1.4a) has seen already over 230 incremental releases as of mid November 2011. For 1.2 and 1.3, over 1000 bug tracker issues have been resolved. In 2011, the community organized five Pharo Sprints, RMoD organized the Deep into Smalltalk School in March 2011.
RMoD is the main maintainer and coordinator of Pharo. It is used widely in both research and industry. With Inria, RMoD is in the process of setting up a Pharo Consortium. There are 25 companies interested in supporting the consortium.

Here is the self-assessment of the team effort following the grid given at [http://www.inria.fr/institut/organisation/instances/commission-d-evaluation](http://www.inria.fr/institut/organisation/instances/commission-d-evaluation).

- **(A5) Audience:** 5 – Used in many universities for teaching, more than 25 companies.
- **(SO3) Software originality:** 3 – Pharo offers a classical basis for some aspects (UI). It includes new frameworks and concepts compared to other implementations Smalltalk.
- **(SM4) Software Maturity:** 4 – Bug tracker, continuous integration, large test suite are on place.
- **(EM4) Evolution and Maintenance:** 4 – Active user group, consortium is being set up.
- **(SDL4) Software Distribution and Licensing:** 4 – Pharo is licensed under MIT.
- **(OC5) Own Contribution:** (Design/Architecture) DA-5, (Coding/Debugging) CD-5, (Maintenance/Support) MS-5, (Team/Project Management) TPM-5

### 4.3. Coral

**Participants:** Damien Pollet [correspondant], Camillo Bruni.

**Web:** [http://rmod.lille.inria.fr/coral](http://rmod.lille.inria.fr/coral).

Coral extends the standard Pharo image, to integrate it into the host operating system shell environment and define system commands in Pharo. In term it will provide facilities for image preparation, configuration and deployment.

### 4.4. VerveineJ

**Participants:** Nicolas Anquetil [correspondant], Andre Hora.

**Web:** Inria project [https://gforge.inria.fr/projects/verveinej/](https://gforge.inria.fr/projects/verveinej/).

VerveineJ is a tool to export Java projects into the MSE format, which can then be imported inside Moose (see above). Although VerveineJ is not a research project in itself, it is an important building block for our research in that it allows us to run the Moose platform on legacy Java projects. Another similar tool, Infusion, already existed to fulfil the same needs, but it was closed sources and presented some errors that tainted the results we could obtain.

### 4.5. VerveineSharp

**Participant:** Usman Bhatti [correspondant].

**Web:** Inria project [https://gforge.inria.fr/projects/verveinesharp/](https://gforge.inria.fr/projects/verveinesharp/).

Similar to VerveineJ (see above), VerveineSharp is a tool to export C# projects into the MSE format, which can then be imported inside Moose. The reasons for creating this project are the same as for VerveineJ: it is an important building block for our research in that it allows us to run the Moose platform on legacy C# projects. Because C# is a proprietary platform, there are no other tools that can give us the same functionality.
RUNTIME Project-Team

5. Software

5.1. Common Communication Interface

Participant: Brice Goglin.

- The Common Communication Interface aims at offering a generic and portable programming interface for a wide range of networking technologies (Ethernet, InfiniBand, ...) and application needs (MPI, storage, low latency UDP, ...).
- CCI is developed in collaboration with the Oak Ridge National Laboratory and several other academics and industrial partners.
- CCI is in early development and currently composed of 19 000 lines of C.
- http://www.cci-forum.org

5.2. Hardware Locality

Participants: Brice Goglin, Samuel Thibault.

- Hardware Locality (HWLOC) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.
- It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics.
- HWLOC targets many types of high-performance computing applications [6], from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers already use HWLOC.
- HWLOC is developed in collaboration with the OPEN MPI project. The core development is still mostly performed by Brice Goglin and Samuel Thibault from the RUNTIME team-project, but many outside contributors are joining the effort, especially from the OPEN MPI and MPICH2 communities.
- HWLOC is composed of 33 000 lines of C.
- http://runtime.bordeaux.inria.fr/hwloc/

5.3. KNem

Participants: Brice Goglin, Stéphanie Moreaud.

- KNEM (Kernel Nemesis) is a Linux kernel module that offers high-performance data transfer between user-space processes.
- KNEM offers a very simple message passing interface that may be used when transferring very large messages within point-to-point or collective MPI operations between processes on the same node.
- Thanks to its kernel-based design, KNEM is able to transfer messages through a single memory copy, much faster than the usual user-space two-copy model.
- KNEM also offers the optional ability to offload memory copies on INTEL I/O AT hardware which improves throughput and reduces CPU consumption and cache pollution.
- KNEM is developed in collaboration with the MPICH2 team at the Argonne National Laboratory and the OPEN MPI project. These partners already released KNEM support as part of their MPI implementations.
- KNEM is composed of 7000 lines of C. Its main contributor is Brice Goglin.
- http://runtime.bordeaux.inria.fr/knem/
5.4. Marcel

**Participants**: Olivier Aumage, Yannick Martin, Samuel Thibault.

- MARCEL is the two-level thread scheduler (also called N:M scheduler) of the PM2 software suite.
- The architecture of MARCEL was carefully designed to support a large number of threads and to efficiently exploit hierarchical architectures (e.g. multicore chips, NUMA machines).
- MARCEL provides a seed construct which can be seen as a precursor of thread. It is only when the time comes to actually run the seed that MARCEL attempts to reuse the resources and the context of another, dying thread, significantly saving management costs.
- In addition to a set of original extensions, MARCEL provides a POSIX-compliant interface which thus permits to take advantage of it by just recompiling unmodified applications or parallel programming environments (API compatibility), or even by running already-compiled binaries with the Linux NPTL ABI compatibility layer.
- For debugging purpose, a trace of the scheduling events can be recorded and used after execution for generating an animated movie showing a replay of the execution.
- The MARCEL thread scheduling library is made of 80 000 lines of code.
- [http://runtime.bordeaux.inria.fr/marcel/](http://runtime.bordeaux.inria.fr/marcel/)
- Marcel has been supported for 2 years (2009-2011) by the INRIA ADT Visimar.

5.5. ForestGOMP

**Participants**: Olivier Aumage, Yannick Martin, Pierre-André Wacrenier.

- FORESTGOMP is an OPENMP environment based on both the GNU OPENMP run-time and the MARCEL thread library.
- It is designed to schedule efficiently nested sets of threads (derived from nested parallel regions) over hierarchical architectures so as to minimize cache misses and NUMA penalties.
- The FORESTGOMP runtime generates nested MARCEL bubbles each time an OPENMP parallel region is encountered, thereby grouping threads sharing common data.
- Topology-aware scheduling policies implemented by BUBBLESCHED can then be used to dynamically map bubbles onto the various levels of the underlying hierarchical architecture.
- FORESTGOMP allowed us to validate the BUBBLESCHED approach with highly irregular, fine grain, divide-and-conquer parallel applications.
- [http://runtime.bordeaux.inria.fr/forestgomp/](http://runtime.bordeaux.inria.fr/forestgomp/)

5.6. Open-MX

**Participants**: Brice Goglin, Ludovic Stordeur.

- The OPEN-MX software stack is a high-performance message passing implementation for any generic ETHERNET interface.
- It was developed within our collaboration with Myricom, Inc. as a part of the move towards the convergence between high-speed interconnects and generic networks.
- OPEN-MX exposes the raw ETHERNET performance at the application level through a pure message passing protocol.
- While the goal is similar to the old GAMMA stack [58] or the recent iWarp [57] implementations, OPEN-MX relies on generic hardware and drivers and has been designed for message passing.
OPEN-MX is also wire-compatible with Myricom MX protocol and interface so that any application built for MX may run on any machine without Myricom hardware and talk other nodes running with or without the native MX stack.

OPEN-MX is also an interesting framework for studying next-generation hardware features that could help ETHERNET hardware become legacy in the context of high-performance computing. Some innovative message-passing-aware stateless abilities, such as multiqueue binding and interrupt coalescing, were designed and evaluated thanks to OPEN-MX [23], [10].

Brice GOGLIN is the main contributor to OPEN-MX. The software is already composed of more than 45,000 lines of code in the Linux kernel and in user-space.

http://open-mx.org/

5.7. StarPU
Participants: Cédric Augonnet, Nicolas Collin, Nathalie Furmento, Cyril Roelandt, Samuel Thibault, Ludovic Courtès.

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Cell processors.
- STARPU offers a unified offloadable task abstraction named codelet. In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (e.g. one function for CUDA and one function for CPUs).
- STARPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. A high-level data management library enforces memory coherency over the machine: before a codelet starts (e.g. on an accelerator), all its data are transparently made available on the compute resource.
- STARPU obtains portable performances by efficiently (and easily) using all computing resources at the same time.
- STARPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.
- STARPU can also leverage existing parallel implementations, by supporting parallel tasks, which can be run concurrently over the machine.
- STARPU provides a reduction mode, which permit to further optimize data management when results have to be reduced.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- STARPU comes with a plug-in for the GNU Compiler Collection (GCC), which extends languages of the C family with syntactic devices to describe STARPU’s main programming concepts in a concise, high-level way.
- http://runtime.bordeaux.inria.fr/StarPU/

5.8. NewMadeleine
Participants: Alexandre Denis, François Trahay, Raymond Namyst.

- NEWMADELEINE is communication library for high performance networks, based on a modular architecture using software components.
- The NEWMADELEINE optimizing scheduler aims at enabling the use of a much wider range of communication flow optimization techniques such as packet reordering or cross-flow packet aggregation.
- **NEWMADELEINE** targets applications with irregular, multiflow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance.
- It is designed to be programmable through the concepts of optimization strategies, allowing experimentations with multiple approaches or on multiple issues with regard to processing communication flows, based on basic communication flows operations such as packet merging or reordering.
- The reference software development branch of the **NEWMADELEINE** software consists in 90,000 lines of code. **NEWMADELEINE** is available on various networking technologies: Myrinet, Infiniband, Quadrics and ETHERNET. It is developed and maintained by Alexandre Denis.
- [http://runtime.bordeaux.inria.fr/newmadeleine/](http://runtime.bordeaux.inria.fr/newmadeleine/)

5.9. **PadicoTM**

**Participant:** Alexandre Denis.

- PadicoTM is a high-performance communication framework for grids. It is designed to enable various middleware systems (such as CORBA, MPI, SOAP, JVM, DSM, etc.) to utilize the networking technologies found on grids.
- PadicoTM aims at decoupling middleware systems from the various networking resources to reach transparent portability and flexibility.
- PadicoTM architecture is based on software components. Puk (the PadicoTM micro-kernel) implements a light-weight high-performance component model that is used to build communication stacks.
- PadicoTM component model is now used in **NEWMADELEINE**. It is the cornerstone for networking integration in the projects “LEGO” and “COOP” from the ANR.
- PadicoTM is composed of roughly 60,000 lines of C.
- PadicoTM is registered at the APP under number IDDN.FR.001.260013.000.S.P.2002.000.10000.
- [http://runtime.bordeaux.inria.fr/PadicoTM/](http://runtime.bordeaux.inria.fr/PadicoTM/)

5.10. **MAQAO**

**Participants:** Denis Barthou, Andres Charif-Rubial.

- MAQAO is a performance tuning tool for OpenMP parallel applications. It relies on the static analysis of binary codes and the collection of dynamic information (such as memory traces). It provides hints to the user about performance bottlenecks and possible workarounds.
- MAQAO relies on binary codes and inserts probes for instrumentation directly inside the binary. There is no need to recompile. The static/dynamic approach of MAQAO analysis is the main originality of the tool, combining performance model with values collected through instrumentation.
- MAQAO has a static performance model for x86 architecture and Itanium. This model analyzes performance of the predecoder, of the decoder and of the different pipelines of the x86 architecture, in particular for SSE instructions.
- The dynamic collection of data in MAQAO enables the analysis of thread interactions, such as false sharing, amount of data reuse, runtime scheduling policy, ...
- MAQAO is in the project ”ProHMPT” from the ANR. A demo of MAQAO has been made in Jan. 2010 for SME/INRIA days and in Nov. 2010 at SuperComputing, INRIA Booth.
- [http://www.maqao.org/](http://www.maqao.org/)
5.11. QIRAL

Participants: Denis Barthou.

- QIRAL is a high level language (expressed through LaTeX) that is used to described Lattice QCD problems. It describes matrix formulations, domain specific properties on preconditionings, and algorithms.
- The compiler chain for QIRAL can combine algorithms and preconditionings, checking validity of the composition automatically. It generates OpenMP parallel code, using libraries, such as BLAS.
- This code is developed in collaboration with other teams participating to the ANR PetaQCD project.

5.12. TreeMatch

Participants: Emmanuel Jeannot, Guillaume Mercier.

- TREEMATCH is a library for performing process placement based on the topology of the machine and the communication pattern of the application.
- TREEMATCH provides a permutation of the processes to the processors/cores in order to minimize the communication cost of the application.
- Important features are: the number of processors can be higher than the number of processes; it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speed); it implements different placement algorithms that are switched according to the input size.
- TREEMATCH is implemented as a load-balancer in Charm++ and as an tool for performing rank reordering in OpenMPI and MPICH-2 [37]
SARDES Project-Team

4. Software

4.1. AAC_tactics

Participants: Thomas Braibant, Damien Pous [correspondant].

AAC_tactics is a plugin for the Coq proof-assistant that implements new proof tactics for rewriting modulo associativity and commutativity. It is available at http://sardes.inrialpes.fr/~braibant/aac_tactics and as part of the Coq distribution.

- ACM: D.2.4 Software/Program Verification
- Keywords: Rewriting, rewriting modulo AC, proof tactics, proof assistant
- Software benefit: AAC_tactics provides novel efficient proof tactics for rewriting modulo associativity and commutativity.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.2. ATBR

Participants: Thomas Braibant, Damien Pous [correspondant].

ATBR (Algebraic Tools for Binary Relations) is library for the Coq proof assistant that implements new proof tactics for reasoning with binary relations. Its main tactics implements a decision procedure for inequalities in Kleene algebras. It is available at http://sardes.inrialpes.fr/~braibant/atbr and as part of the Coq distribution contributed modules.

- ACM: D.2.4 Software/Program Verification
- Keywords: Binary relations, Kleene algebras, proof tactics, proof assistant
- Software benefit: ATBR provides new proof tactics for reasoning with binary relations.
- License: LGPL
- Type of human computer interaction: N/A
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Coq

4.3. MoKa

Participant: Sara Bouchenak [correspondant].
MoKa is a software framework for the modeling and capacity planning of distributed systems. It first provides a set of tools to build analytical models that describe the behavior of distributed computing systems, in terms of performance, availability, cost. The framework allows to include several model algorithms and to compare them regarding their accuracy and their efficiency. Furthermore, MoKa provides a set of tools to build capacity planning methods. A capacity planning method allows to find a distributed system configuration that guarantee given quality-of-service objectives. MoKa is able to include different capacity planning algorithms and to compare them regarding their efficiency and the optimality of their results. MoKa is available at: [http://sardes.inrialpes.fr/research/moka](http://sardes.inrialpes.fr/research/moka).

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems, D.2.9 Management
- Keywords: Caching, multi-tier systems, consistency, performance
- Software benefit: a novel end-to-end caching protocol for multi-tier services.
- License: TBD
- Type of human computer interaction: command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java

4.4. ConSer

**Participant:** Sara Bouchenak [correspondant].

ConSer is a software framework for the modeling and the concurrency and admission control of servers systems. It implements a fluid-based model that exhibits the dynamics and behavior of a server system in terms of service performance and availability. ConSer implements various novel admission control laws for servers such as AM-C, PM-C, AA-PM-C and PA-AM-C. A control law produces the server concurrency level that allows to trade-off and meet given service level objectives. ConSer’s modeling and control laws algorithms are implemented following a proxy-based approach for more transparency.

- ACM: C.4 Performance of Systems; D.2.9 Management
- Keywords: System management, capacity planning, performance management
- Software benefit: MoKa provides modeling, capacity planning and performance management facilities for application server clusters. Thanks to its model-based capacity planning, MoKa is able to enforce service level objectives while minimizing the service cost.
- License: LGPL
- Type of human computer interaction: web interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, AspectJ

4.5. e-Caching

**Participants:** Damián Serrano, Sara Bouchenak [correspondant].

E-Caching is a software framework for higher scalability of multi-tier Internet services through end-to-end caching of dynamic data. It provides a novel caching solution that allows to cache different types of data (e.g. Web content, database query results, etc.), at different locations of multi-tier Internet services. The framework allows to combine different caches and, thus, to provide higher scalability of Internet services. E-Caching maintains the integrity of the cached data through novel distributed caching algorithms that guarantee the consistency of the underlying data.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems
- Keywords: Caching, multi-tier systems, consistency, performance
- Software benefit: a novel end-to-end caching protocol for multi-tier services, consistency management, performance improvement.
- License: TBD
- Type of human computer interaction: command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java
4.6. MRB

**Participants:** Amit Sangroya, Dàmian Serrano, Sara Bouchenak [correspondant].

MRB is a software framework for benchmarking the performance and dependability of MapReduce distributed systems. It includes five benchmarks covering several application domains and a wide range of execution scenarios such as data-intensive vs. compute-intensive applications, or batch applications vs. interactive applications. MRB allows to characterize application workload, faultload and dataload, and it produces extensive performance and dependability statistics.

- ACM: C.2.4 Distributed Systems, C.4 Performance of Systems
- Keywords: Benchmark, performance, dependability, MapReduce, Hadoop, Cloud Computing
- Software benefit: the first performance and dependability benchmark suite for MapReduce systems.
- License: TBD
- Type of human computer interaction: GUI and command-line interface
- OS/Middleware: Windows, Linux, MacOS X
- Programming language: Java, Unix Shell scripts

4.7. BZR

**Participants:** Eric Rutten [correspondant], Gwenaël Delaval [POP ART team].

BZR is a reactive language, belonging to the synchronous languages family, whose main feature is to include discrete controller synthesis within its compilation. It is equipped with a behavioral contract mechanisms, where assumptions can be described, as well as an "enforce" property part; the semantics of the latter is that the property should be enforced by controlling the behaviour of the node equipped with the contract. This property will be enforced by an automatically built controller, which will act on free controllable variables given by the programmer.

BZR is now further developed with the Pop-Art team, where G. Delaval got a position. It has been designed and developed in the Sardes team in relation with the research topic on Model-based Control of Adaptive and Reconfigurable Systems. It is currently applied in different directions: component-based design and the Fractal framework; real-time control systems and the Orccad design environment; operating systems and administration loops in virtual machines; hardware and reconfigurable architecture (FPGAs).

See also the web page [http://bzr.inria.fr](http://bzr.inria.fr).

- ACM: D.3.3 [Programming Languages]: Language Constructs and Features—Control structures; C.3 [Special-purpose and Application-based Systems]: Real-time and embedded systems; D.2.2 [Software Engineering]: Design Tools and Techniques—Computer-aided software engineering, State diagrams; D.2.4 [Software Engineering]: Software / Program Verification—Formal methods, Programming by contract
- Keywords: Discrete controller synthesis, modularity, components, contracts, reactive systems, synchronous programming, adaptive and reconfigurable systems
- Software benefit: the first integration of discrete control synthesis in a compiler, making it usable at the level of the programming language.
- License: TBD
- Type of human computer interaction: programming language and command-line interface
- OS/Middleware: Linux
- Programming language: Caml; generates C or Java or Caml executable code
5. Software

5.1. QualiPSo Factory: Next Generation Forge

Participants: Gérald Oster [contact], Jérôme Blanchard, Christophe Bouthier.

The QualiPSo Factory is a next generation forge based on Service Oriented Architecture developed within the QualiPSo european project. Forges transform foreigners into collaborators, sometimes into developers. Forges are online services that allow instantiation, composition and management of collaborative services. Traditionally, provided collaborative services are version control systems, issue trackers, forums, mailing lists or wikis. In the framework of the european QualiPSo project, we are designing and implementing the next generation of forges. The QualiPSo factory framework aims to ease collaboration between forge users and integration of new collaborative services by developers. Our proposal relies on a software oriented architecture (SOA) and thereby allows composition of services. The current architecture provides core services such as security, notification, indexation, composition and naming which are externalized to other collaborative services. The Factory has been delivered as an outcome of the Qualipso project. Its future needs to be clarified.

1 http://qualipso.gforge.inria.fr/
2 http://www.qualipso.org
4. Software

4.1. Introduction

SWING develops several tools supporting its research like SOCLIB and Wiplan. Moreover, SWING is an active contributor to WSnet (http://wsnet.gforge.inria.fr/) a multi-hop wireless network discrete event simulator. WSnet was created in the ARES team and it is now supported by the D-NET team of INRIA Rhône-Alpes. SWING is one of the most important contributor for the design of protocol libraries in WSnet.

4.2. SOCLIB

Participant: Tanguy Risset [correspondant].

SocLib is a library of simulation models for virtual components (IP cores) for Systems on Chip. Many simulation models are under development, SocLib currently contains simulation models for processors (Mips, ARM), memories and network on chips (Spin and DSspin developed at LIP6 laboratory. SocLib permits to simulate at cycle accurate application running on embedded computing systems such as mobile phones. SWING use this platform to prototype design techniques either for embedded software or for hardware parts of signal processing applications.

See also the web page https://www.soclib.fr/trac/dev/wiki.

4.3. Wiplan

Participants: Jean-Marie Gorce [correspondant], Guillaume Villemaud, Meiling Luo, Dmitry Umansky, Tao Wang.

Wiplan is a software including an Indoor propagation engine and a wireless LAN optimization suite, which has been registered by INSA-Lyon. The heart of this software is the propagation simulation core relying on an original method, MR-FDPF (multi-resolution frequency domain ParFlow). The discrete ParFlow equations are translated in the Fourier domain providing a wide linear system, solved in two steps taking advantage of a multi-resolution approach. The first step computes a cell-based tree structure referred to as the pyramid. In the second phase, a radiating source is simulated, taking advantage of the pre-processed pyramidal structure. Using of a full-space discrete simulator instead of classical ray-tracing techniques is a challenge due to the inherent high computation re-quests. However, we have shown that the use of a multi-resolution approach allows the main computation load to be restricted to a pre-processing phase. Extensive works have been done to make predictions more realistic. The network planning and optimization suite is based on a multi-criteria model relying on a Tabu solver. The development of the wiplan software is a part of the european project iPLAN (IAPP-FP7 project).

See also the web page http://wiplan.citi.insa-lyon.fr.
5. Software

5.1. Gibbs’ Sampler

Participant: Chung Shue Chen.

The work on the self optimization of cellular networks based on Gibbs’ sampler (see Section 6.1.1.3) carried out in the joint laboratory with Alcatel-Lucent, led to the development of a software prototype that was presented by C. S. Chen at the INRIA Alcatel-Lucent joint laboratory seminar in March 2010 and demonstrated at the Alcatel-Lucent Bell Labs Open Days in May 2010. It was also demonstrated in the LINCS opening ceremony in April 2011.

5.2. PSI2

Participant: Ana Bušić.

The work on perfect sampling (see Section 6.2.3) has been partially implemented in a software tool PSI2, in collaboration with MESCAL team [INRIA Grenoble - Rhône-Alpes]; https://gforge.inria.fr/projects/psi.
TRISKELL Project-Team

5. Software

5.1. Kermeta

Participants: Didier Vojtisek [correspondant], Olivier Barais, Cédric Bouhours, Xavier Dolques, Jacques Falcou, François Fouquet, Marie Gouyette, Jean-Marc Jézéquel, Hajanirina Johary Rambelontsalama.

Nowadays, object-oriented meta-languages such as MOF (meta-object Facility) are increasingly used to specify domain-specific languages in the model-driven engineering community. However, these meta-languages focus on structural specifications and have no built-in support for specifications of operational semantics. Integrated with the industrial standard Ecore and aligned with the OMG standard EMOF 2.0, the Kermeta language consists in a extension to these meta languages to support behavior definition. The language adds precise action specifications with static type checking and genericity at the meta level. Based on object-orientation and aspect orientation concepts, the Kermeta language adds model specific concepts. It is used in several use cases:

- to give a precise semantic of the behavior of a metamodel which then can be simulated.
- to act as a model transformation language.
- to act as a constraint language.

The development environment built for the Kermeta language provides an integrated workbench based on Eclipse. It offers services such as: model execution, text editor (with syntax highlighting, code autocompletion), additional views and various import/export transformations.

Thanks to Kermeta it is possible to build various frameworks dedicated to domain specific metamodels. Those frameworks are organised into MDKs (Model Development Kits). For example, Triskell proposes MDKs to work with the metamodels such as Java5, UML2, RDL (requirements), Ecore, Traceability,...

In 2011, Kermeta tooling has been refactored into a version 2.0.x in order to ease the integration of various MOF related languages in the tool chain. This new version also focuses on a fully compiled mode that allows to deploy kermeta programs in production environments.

See also the web page http://www.kermeta.org.

- APP: IDDN.FR.001.420009.000.S.P.2005.000.10400
- Version: 2.0.1
- Programming language: Java, Scala, Kermeta

5.2. Kevoree

Participants: Olivier Barais [correspondant], François Fouquet, Erwan Daubert, Johann Bourcier, Gregory Nain, Noël Plouzeau.

The Kevoree project ¹ defines a framework dedicated to distributed systems design, using the models at runtime paradigm and a component-based software architecture approach. This framework offers a high-level abstraction for managing components and their interactions. It also provides concepts to describe the underlying infrastructure: resources, logical nodes and their topology.

Kevoree also provides a set of tools to manipulate model abstraction easily, relying in part on a Domain Specific Language (DSL) called KevScript. This DSL makes the architecture model modifications easier. Our DSL can also be used in a reasoning engine to dynamically adapt the running system by applying some changes at different level (SaaS, PaaS and IaaS). Kevoree has several runtime platform implementations, allowing execution of applications on various devices (e.g. JavaSeE, Android, µController such as Arduino, and cloud virtual nodes).

¹http://kevoree.org
See also the web page [http://www.kevoree.org](http://www.kevoree.org).

- Version: 1.0
- Programming language: Java, Scala, Kermeta

### 5.3. Pramana

**Participants:** Benoit Baudry, Juan-Jose Cadavid Gomez, Benoit Combemale, Xavier Dolques, Hajanirina Johary Rambelontsalama, Didier Vojtisek [correspondant].

Pramana is an open-source tool, which automatically generates valid instances of a metamodel. These instances can then be used for analysis, verification, simulation or validation of the metamodel. The core mechanism for model generation relies on the bounded constraint-solver of Alloy, a lightweight model checker developed at the MIT. Alloy is integrated in Kermeta to allow the generation of instances of Ecore or Kermeta metamodels. Pramana implements this integration through a series of transformations and analysis, all implemented in Kermeta.

Metamodel instances can be used as input data for model transformation testing, and in particular for the testing of Kermeta code. For this purpose Pramana includes the K-Yeti module implementing a binding between the Kermeta language and the generic testing framework Yeti². This module allows a Kermeta user to automatically run test cases.

See also the web page [https://www.irisa.fr/triskell/Softwares/pramana](https://www.irisa.fr/triskell/Softwares/pramana).

- Version: 1.0
- Programming language: Java, Alloy, Kermeta

---

5. Software

5.1. Graphite

Participants: Phuong Ho, Bruno Lévy, David Lopez, Romain Merland, Vincent Nivoliers, Jeanne Pellerin, Nicolas Ray.

Graphite is a research platform for computer graphics, 3D modeling and numerical geometry. It comprises all the main research results of our “geometry processing” group. Data structures for cellular complexes, parameterization, multi-resolution analysis and numerical optimization are the main features of the software. Graphite is publicly available since October 2003. It is hosted by Inria GForge since September 2008 (1000 downloads in two months). Graphite is one of the common software platforms used in the frame of the European Network of Excellence AIMShape.

5.2. MicroMegas

Participant: Samuel Hornus.

Micromegas is a 3D modeler, developed as a plugin of Graphite, dedicated to molecular biology. Micromegas is developed in cooperation with the Fourmentin Guilbert foundation. Biologists need simple spatial modeling tools to help in understanding the role of objects’ relative position in the functioning of the cell. In this context, we offer a tool for easy DNA modeling. The tool generates DNA along a Bézier curve, open or closed, allows fine-tuning of atoms’ position and, most importantly, exports to PDB.

5.3. OpenNL - Open Numerical Library

Participants: Thomas Jost, Bruno Lévy, Nicolas Ray, Rhaleb Zayer.

OpenNL is a standalone library for numerical optimization, especially well-suited to mesh processing. The API is inspired by the graphics API OpenGL, this makes the learning curve easy for computer graphics practitioners. The included demo program implements our LSCM [7] mesh unwrapping method. It was integrated in Blender by Brecht Van Lommel and others to create automatic texture mapping methods. OpenNL is extended with two specialized modules:

- CGAL parameterization package: this software library, developed in cooperation with Pierre Alliez and Laurent Saboret, is a CGAL package for mesh parameterization.
- Concurrent Number Cruncher: this software library extends OpenNL with parallel computing on the GPU, implemented using the CUDA API.

5.4. Intersurf

Participants: Xavier Cavin, Nicolas Ray.

Intersurf is a plugin of the VMD (Visual Molecular Dynamics) software. VMD is developed by the Theoretical and Computational Biophysics Group at the Beckmann Institute at University of Illinois. The Intersurf plugin is released with the official version of VMD since the 1.8.3 release. It provides surfaces representing the interaction between two groups of atoms, and colors can be added to represent interaction forces between these groups of atoms. We plan to include in this package the new results obtained this year in molecular surface visualization by Matthieu Chavent.

5.5. Gocad

Participants: Guillaume Caumon, Nicolas Cherpeau, Bruno Lévy, Romain Merland, Jeanne Pellerin.
Gocad is a 3D modeler dedicated to geosciences. It was developed by a consortium headed by Jean-Laurent Mallet, in the Nancy School of Geology. Gocad is now commercialized by Earth Decision Sciences (formerly T-Surf), a company which was initially a start-up company of the project-team. Gocad is used by all major oil companies (Total-Fina-Elf, ChevronTexaco, Petrobras, etc.), and has become a de facto standard in geo-modeling. Luc Buatois’s work on GPU-based numerical solvers is now integrated in Gocad’s grid generation software SKUA.

5.6. LibSL

Participants: Anass Lasram, Sylvain Lefebvre.

LibSL is a Simple library for graphics. Sylvain Lefebvre continued development of the LibSL graphics library (under CeCILL-C licence, filed at the APP). LibSL is a toolbox for rapid prototyping of computer graphics algorithms, under both OpenGL, DirectX 9/10, Windows and Linux. The library is actively used in both the REVES / INRIA Sophia-Antipolis and the Alice / INRIA Nancy Grand-Est teams.
ALPAGE Project-Team

5. Software

5.1. Syntax

Participants: Pierre Boullier [correspondant], Sattisvar Tandabany, Benoît Sagot.

See also the web page http://syntax.gforge.inria.fr/.

The (currently beta) version 6.0 of the Syntax system (freely available on INRIA GForge) includes various deterministic and non-deterministic CFG parser generators. It includes in particular an efficient implementation of the Earley algorithm, with many original optimizations, that is used in several of Alpage’s NLP tools, including the pre-processing chain SXPipe and the LFG deep parser SXLFG. This implementation of the Earley algorithm has been recently extended to handle probabilistic CFG (PCFG), by taking into account probabilities both during parsing (beam) and after parsing (n-best computation). Syntax 6.0 also includes parsers for various contextual formalisms, including a parser for Range Concatenation Grammars (RCG) that can be used among others for TAG and MC-TAG parsing.

Direct NLP users of Syntax for NLP, outside Alpage, include Alexis Nasr (Marseilles) and other members of the Sequoia ANR project (see section 8.2.1), Owen Rambow and co-workers at Columbia University (New York), as well as (indirectly) all SXPipe and/or SXLFG users. The project-team VASY (INRIA Rhône-Alpes) is one of Syntax’ user for non-NLP applications.

5.2. System DyALog

Participant: Éric Villemonte de La Clergerie [maintainer].

DYALOG on INRIA GForge: http://dyalog.gforge.inria.fr/

DYALOG provides an environment to compile and execute grammars and logic programs. It is essentially based on the notion of tabulation, i.e. of sharing computations by tabulating traces of them. DYALOG is mainly used to build parsers for Natural Language Processing (NLP). It may nevertheless be used as a replacement for traditional PROLOG systems in the context of highly ambiguous applications where sub-computations can be shared.

The current release 1.13.0 of DYALOG is freely available by FTP under an open source license and runs on Linux platforms for x86 and architectures and on Mac OS intell (both 32 and 64bits architectures). A partial port for Window Cygwin has been successful but has not yet been integrated and finalized.

The current release handles logic programs, DCGs (Definite Clause Grammars), FTAGs (Feature Tree Adjoining Grammars), FTIGs (Feature Tree Insertion Grammars) and XRCGs (Range Concatenation Grammars with logic arguments). Several extensions have been added to most of these formalisms such as intersection, Kleene star, and interleave operators. Typed Feature Structures (TFS) as well as finite domains may be used for writing more compact and declarative grammars [127].

C libraries can be used from within DYALOG to import APIs (mysql, libxml, sqlite, ...).

DYALOG is largely used within ALPAGE to build parsers but also derivative softwares, such as a compiler of Meta-Grammars (cf. 5.3). It has also been used for building a parser from a large coverage French TIG/TAG grammar derived from a Meta-Grammar. This parser has been used for the Parsing Evaluation campaign EASY, the two Passage campaigns (Dec. 2007 and Nov. 2009), cf. [125], [126], and very large amount of data (700 millions of words) in the SCRIBO project.

DYALOG is used at LORIA (Nancy), University of Coruña (Spain), Instut Gaspard Monge (Univ. Marne La Vallé), University of Nice, and a few other users.

DYALOG and other companion modules are available on INRIA GForge.
5.3. Tools and resources for Meta-Grammars

Participant: Éric Villemonte de La Clergerie [maintainer].

MGCOMP, MGTOOLS, and FRMG on INRIA GForge: http://mgkit.gforge.inria.fr/

DYALOG (cf. 5.2) has been used to implement MGCOMP, Meta-Grammar compiler. Starting from an XML representation of a MG, MGCOMP produces an XML representation of its TAG expansion.

The current version 1.5.0 is freely available by FTP under an open source license. It is used within ALPAGE and (occasionally) at LORIA (Nancy) and at University of Pennsylvania.

The current version adds the notion of namespace, to get more compact and less error-prone meta-grammars. It also provides other extensions of the standard notion of Meta-Grammar in order to generate very compact TAG grammars. These extensions include the notion of Guarded nodes, i.e. nodes whose existence and non-existence depend on the truth value of a guard, and the use of the regular operators provided by DYALOG on nodes, namely disjunction, interleaving and Kleene star. The current release provides a dump/restore mechanism for faster compilations on incremental changes of a meta-grammars.

The current version of MGCOMP has been used to compile a wide coverage Meta-Grammar FRMG (version 2.0.1) to get a grammar of around 200 TAG trees [129]. Without the use of guarded nodes and regular operators, this grammar would have more than several thousand trees and would be almost intractable. FRMG has been packaged and is freely available.

To ease the design of meta-grammars, a set of tools have been implemented, mostly by Éric de La Clergerie, and collected in MGTOOLS (version 2.2.2). This package includes a converter from a compact format to a XML pivot format, an Emacs mode for the compact and XML formats, a graphical viewer interacting with Emacs and XSLT stylesheets to derive HTML views. A new version is under development to provide an even more compact syntax and some checking mechanisms to avoid frequent typo errors.

The various tools on Metagrammars are available on INRIA GForge.

5.4. The Bonsai PCFG-LA parser

Participants: Benoît Crabbé [correspondant], Marie Candito, Pascal Denis, Djamé Seddah.

Web page: http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html

Alpage has developed as support of the research papers [81], [70], [71], [12] a statistical parser for French, named Bonsai, trained on the French Treebank. This parser provides both a phrase structure and a projective dependency structure specified in [4] as output. This parser operates sequentially: (1) it first outputs a phrase structure analysis of sentences reusing the Berkeley implementation of a PCFG-LA trained on French by Alpage (2) it applies on the resulting phrase structure trees a process of conversion to dependency parses using a combination of heuristics and classifiers trained on the French treebank. The parser currently outputs several well known formats such as Penn treebank phrase structure trees, Xerox like triples and CONLL-like format for dependencies. The parsers also comes with basic preprocessing facilities allowing to perform elementary sentence segmentation and word tokenisation, allowing in theory to process unrestricted text. However it is believed to perform better on newspaper-like text. The parser is available under a GPL license.

5.5. The MICA parser

Participants: Benoît Sagot [correspondant], Marie Candito, Pierre Boullier, Djamé Seddah.

Web page: http://mica.lif.univ-mrs.fr/

MICA (Marseille-INRIA-Columbia-AT&T) is a freely available dependency parser [61] currently trained on English and Arabic data, developed in collaboration with Owen Rambow and Daniel Bauer (Columbia University) and Srinivas Bangalore (AT&T). MICA has several key characteristics that make it appealing to researchers in NLP who need an off-the-shelf parser, based on Probabilistic Tree Insertion Grammars and on the SYNTAX system. MICA is fast (450 words per second plus 6 seconds initialization on a standard high-end machine) and has close to state-of-the-art performance (87.6% unlabeled dependency accuracy on the Penn Treebank).
MICA consists of two processes: the supertagger, which associates tags representing rich syntactic information with the input word sequence, and the actual parser, based on the INRIA SYNTAX system, which derives the syntactic structure from the n-best chosen supertags. Only the supertagger uses lexical information, the parser only sees the supertag hypotheses.

MICA returns $n$-best parses for arbitrary $n$; parse trees are associated with probabilities. A packed forest can also be returned.

5.6. **Alpage’s linguistic workbench, including SxPipe**

**Participants:** Benoît Sagot [correspondant], Rosa Stern, Pierre Boullier, Éric Villemonte de La Clergerie.

See also the web page http://lingwb.gforge.inria.fr/.

Alpage’s linguistic workbench is a set of packages for corpus processing and parsing. Among these packages, the SxPipe package is of a particular importance

SxPipe, now in version 2 [109] is a modular and customizable chain aimed to apply to raw corpora a cascade of surface processing steps. It is used

- as a preliminary step before Alpage’s parsers (FRMG, SxLFQ);
- for surface processing (named entities recognition, text normalization...).

Developed for French and for other languages, SxPipe 2 includes, among others, various named entities recognition modules in raw text, a sentence segmenter and tokenizer, a spelling corrector and compound words recognizer, and an original context-free patterns recognizer, used by several specialized grammars (numbers, impersonal constructions, quotations...).

5.7. **MElt**

**Participants:** Pascal Denis [correspondant], Benoît Sagot.

MElt is a part-of-speech tagger, trained for French (on the French TreeBank and coupled with the Lefff), English [89], Spanish, Kurmanji Kurdish [131] and Persian [56], [42]. It is state-of-the-art for French. It is distributed freely as a part of the Alpage linguistic workbench.

5.8. **The Alexina framework: the Lefff syntactic lexicon, the Aleda entity database and other Alexina resources**

**Participants:** Benoît Sagot [correspondant], Laurence Danlos.

See also the web page http://gforge.inria.fr/projects/alexina/.

Alexina is Alpage’s Alexina framework for the acquisition and modeling of morphological and syntactic lexical information. The first and most advanced lexical resource developed in this framework is the Lefff, a morphological and syntactic lexicon for French.

Historically, the Leff 1 was a freely available French morphological lexicon for verbs that has been automatically extracted from a very large corpus. Since version 2, the Lefff covers all grammatical categories (not just verbs) and includes syntactic information (such as subcategorization frames); Alpage’s tools, including Alpage’s parsers, rely on the Lefff. The version 3 of the Lefff, which has been released in 2008, improves the linguistic relevance and the interoperability with other lexical models.

Other Alexina lexicons are under development, in particular for Spanish (the Leffé), Polish, Slovak, English, Galician, Persian, Kurdish.

Alexina also hosts Aleda [124], an large-scale entity database currently developed for French but under development for English, extracted automatically from Wikipedia and Geonames. It is used among others in the SxPipe processing chain and its NP named entity recognition, as well as in the NOMOS named entity linking system.
5.9. The free French wordnet WOLF

Participants: Benoît Sagot [correspondant], Marianna Apidianaki.

The WOLF (Wordnet Libre du Français) is a wordnet for French, i.e., a lexical semantic database. The development of WOLF started in 2008 [113], [8]. At this time, we focused on benefitting from available resources of three different types: general and domain-specific bilingual dictionaries, multilingual parallel corpora and Wiki resources (Wikipedia and Wiktionaries). This work was achieved in a large part in collaboration with Darja Fišer (University of Ljubljana, Slovenia), in parallel with the development of a free Slovene wordnet, sloWNet. Since 2008, work specific to each of both resources has been done [114], but since end-2010 the collaboration has been re-activated. This is due among others to the fact that the joint development of WOLF and sloWNet is one of the main objectives of the two-year PROTEUS bilateral PHC project co-headed by Benoît Sagot (2010-2011, see section 8.3.2). Moreover, the EDyLex project also contributed to funding the improvement of the WOLF, in particular through the work of Marianna Apidianaki.

The WOLF is freely available under the Cecill-C license. It has already been used in various experiments, within and outside Alpage.

5.10. Automatic construction of distributional thesauri

Participants: Enrique Henestroza Anguiano [correspondant], Pascal Denis.

FREDIST is a freely-available (LGPL license) Python package that implements methods for the automatic construction of distributional thesauri [31].

We have implemented the context relation approach to distributional similarity, with various context relation types and different options for weight and measure functions to calculate distributional similarity between words. Additionally, FREDIST is highly flexible, with parameters including: context relation type(s), weight function, measure function, term frequency thresholding, part-of-speech restrictions, filtering of numerical terms, etc.

Distributional thesauri for French are also available, one each for adjectives, adverbs, common nouns, and verbs. They have been constructed with FreDist and use the best settings obtained in an evaluation. We use the L’Est Republicain corpus (125 million words), Agence France-Presse newswire dispatches (125 million words) and a full dump of the French Wikipedia (200 million words), for a total of 450 million words of text.

5.11. Tools and resources for time processing

Participants: Laurence Danlos [correspondant], Pascal Denis, Philippe Muller.

Apetite provides a set of tools to handle ISO-TimeML annotations, predict temporal structures from timex/event mark-ups, and different ways of evaluating the results. It is licensed under the Cecill, a GPL-like license http://www.irit.fr/~Philippe.Muller/tools/apetite-0.7.tgz.

In parallel, Alpage developed the French TimeBank [22], [21], a freely-available corpus annotated with ISO-TimeML-compliant temporal information (dates, events and relations between events).

5.12. System EasyRef

Participant: Éric Villemonte de La Clergerie [maintainer].

PASSAGE action

A collaborative WEB service EASYREF has been developed, in the context of ANR action Passage, to handle syntactically annotated corpora. EASYREF may be used to view annotated corpus, in both EASY or PASSAGE formats. The annotations may be created and modified. Bug reports may be emitted. The annotations may be imported and exported. The system provides standard user right management. The interface has been designed with the objectives to be intuitive and to speed edition.
EASYREF relies on a Model View Controller design, implemented with the Perl Catalyst framework. It exploits WEB 2.0 technologies (i.e. AJAX and JavaScript).

Version 2 has been used by ELDA and LIMSI to annotate a new corpus of several thousands words for PASSAGE.

A preliminary version 3 has been developed by François Guéris and revised by Éric de La Clergerie, relying on Berkeley DB XML to handle very large annotated corpora and to provide a complete query language expanded as XQuery expressions. EASYREF is maintained under INRIA GForge.
ARIANA Project-Team

5. Software

5.1. Software

5.1.1. Deposits

- The software WAIHEKE was deposited with the APP in October 2011. It was developed for classifying 3D-point data generated from airborne lidar systems or multi-view imagery. The input point cloud is labeled into four classes of interest (building, ground, vegetation and clutter).

- The software SIKORA was deposited with the APP in October 2011. It was developed for extracting 3D-segments, planes, cylinders, cones, spheres and tori by region growing from 3D-point clouds.

- The software MOJOPIN was deposited with the APP in October 2011. It was developed for performing planimetric arrangements of urban components, including roof sections and trees, from labeled point clouds.

- The software SCOMBO v1.0 and Hierarchical SCOMBO v1.0 were deposited with the APP in December 2011. It deals with the supervised classification of multiband optical images by using Markov random fields and hierarchical Markov random fields, respectively.
AROBAS Project-Team

5. Software

5.1. Experimental Testbeds

Methodological solutions to the multi-faceted problem of robot autonomy have to be combined with the ever present preoccupation of robustness and real-time implementability. In this respect, validation and testing on physical systems is essential, not only as a means to bring together all aspects of the research done in AROBAS—and thus maintain the coherence and unity of the project-team—but also to understand the core of the problems on which research efforts should focus in priority. The instrumented indoor and outdoor wheeled robots constitute a good compromise in terms of cost, security, maintenance, complexity and usefulness to test much of the research conducted in the project-team and to address real size problems currently under investigation in the scientific community. For the next few years, we foresee on site testbeds dedicated to ground robotic applications (figure 1 Left and Center).

- **HANNIBAL Indoor mobile robot**
  Our cart-like platform, built by Neobotix can operate on flat surfaces, in both indoor and outdoor environments. This platform is equipped with the various sensors needed for SLAM purposes, autonomous navigation and sensor-based control. With its programming further developed to become user-friendly, it has become one of the team’s main testbeds for fast prototyping of perception, control and autonomous navigation algorithms.

- **CyCab Urban electrical car**
  Two instrumented electrical cars of the CyCab family are destined to validate researches in the domain of Intelligent urban vehicle. CyCabs are used as experimental testbeds in several national projects.

- **Hexacopter VTOL vehicle**
  A basic version of this machine was recently acquired from Mikrokopter Inc.(Germany) by our colleagues (T. Hamel, G. Ducard, M.-D. Hua) from the SIS (Signal, Images et Systèmes) research pole at I3S-UNSA-CNRS. It has a diameter of 90cm, weights about 1.5 kg, and can carry a payload up to 1.5 kg (figure 1 Right). The flight time autonomy varies between 6mn and 18mn, depending on the payload, and it can be extended provided that the battery capacity is extended accordingly. The machine’s external envelope has been modified for safety reasons. Initial flight tests have been conducted, and the aircraft is currently being equipped with various sensors (GPS, accelerometers, gyrometers, camera,...). We are working with our colleagues from I3S to control this vehicle with the aim of providing it with large autonomy capabilities and robust performance. It is also a benchmark to validate various estimation/control issues that we are currently investigating.
Figure 1. Left: The Hannibal platform. Center: The Cycab vehicle. Right: Hexacopter
ARTIS Project-Team

5. Software

5.1. Introduction

ARTIS insists on sharing the software that is developed for internal use. These are all listed in a dedicated section on the web site http://artis.imag.fr/Software.

5.2. libQGLViewer: a 3D visualization library

libQGLViewer is a library that provides tools to efficiently create new 3D viewers. Simple and common actions such as moving the camera with the mouse, saving snapshots or selecting objects are not available in standard APIs, and libQGLViewer fills this gap. It merges in a unified and complete framework the tools that every one used to develop individually. Creating a new 3D viewer now requires 20 lines of cut-pasted code and 5 minutes. libQGLViewer is distributed under the GPL licence since January 2003, and several hundreds of downloads are recorded each month 1.

5.3. PlantRad

Participant: Cyril Soler [contact].

PlantRad is a software program for computing solutions to the equation of light equilibrium in a complex scene including vegetation. The technology used is hierarchical radiosity with clustering and instantiation. Thanks to the latter, PlantRad is capable of treating scenes with a very high geometric complexity (up to millions of polygons) such as plants or any kind of vegetation scene where a high degree of approximate self-similarity permits a significant gain in memory requirements. Its main domains of applications are urban simulation, remote sensing simulation (See the collaboration with Noveltis, Toulouse) and plant growth simulation, as previously demonstrated during our collaboration with the LIAMA, Beijing.

5.4. High Quality Renderer

Participant: Cyril Soler [contact].

In the context of the European project RealReflect, the ARTIS team has developed the HQR software based on the photon mapping method which is capable of solving the light balance equation and of giving a high quality solution. Through a graphical user interface, it reads X3D scenes using the X3DToolKit package developed at ARTIS, it allows the user to tune several parameters, computes photon maps, and reconstructs information to obtain a high quality solution. HQR also accepts plugins which considerably eases the development of new algorithms for global illumination, those benefiting from the existing algorithms for handling materials, geometry and light sources. HQR is freely available for download 2.

5.5. MobiNet

Participants: Fabrice Neyret [contact], Joëlle Thollot.

The MobiNet software allows for the creation of simple applications such as video games, virtual physics experiments or pedagogical math illustrations. It relies on an intuitive graphical interface and language which allows the user to program a set of mobile objects (possibly through a network). It is available in public domain 3 for Linux, Windows and MacOS, and originated in a collaboration with the EVASION project-team.

---

1 http://artis.imag.fr/Software/QGLViewer/
2 http://artis.imag.fr/~Cyril.Soler/HQR
3 http://mobinet.inrialpes.fr
The main aim of MobiNet is to allow young students at high school level with no programming skills to experiment, with the notions they learn in math and physics, by modeling and simulating simple practical problems, and even simple video games. This platform has been massively used during the Grenoble INP "engineer weeks" since 2002: 150 senior high school pupils per year, doing a 3 hour practice. This work is partly funded by Grenoble INP. Various contacts are currently developed in the educational world. Besides "engineer weeks", several groups of "monitors" PhD students conducts experimentations based on MobiNet with a high school class in the frame of the courses. Moreover, presentation in workshops and institutes are done, and a web site repository is maintained.

5.6. Freestyle

Freestyle is a software for Non-Photorealistic Line Drawing rendering from 3D scenes. It is designed as a programmable interface to allow maximum control over the style of the final drawing: the user "programs" how the silhouettes and other feature lines from the 3D model should be turned into stylized strokes using a set of programmable operators dedicated to style description. This programmable approach, inspired by the shading languages available in photorealistic renderers such as Pixar’s RenderMan, overcomes the limitations of integrated software with access to a limited number of parameters and permits the design of an infinite variety of rich and complex styles. The system currently focuses on pure line drawing as a first step. The style description language is Python augmented with our set of operators. Freestyle was developed in the framework of a research project dedicated to the study of stylized line drawing rendering from 3D scenes. This research has lead to two publications [31], [30].

Figure 1. As a GPL and OpenSource software, Freestyle get a new life from the blender developer community.

In 2008, Freestyle get a new life, completely outside ARTIS or INRIA: it was the basis of one of the 6 Google Summer of Code projects awarded to the Blender Foundation. The goal of the project was to integrate Freestyle to the well known free 3D modeler Blender, as its standard NPR line-drawing renderer. Maxime Curioni (under the mentoring of Jean-Luc Peurière from the Blender Foundation), is currently making the integration. First beta versions are publicly available, and tested by enthusiasts around the web.

5.7. Diffusion Curves

Participant: Joëlle Thollot [contact].

---

4 http://www.blender.org/
We provide an implementation of the vector drawing tool described in the 2008 Diffusion Curves Siggraph paper. This prototype is composed of the Windows binary, along with the required shader programs (ie. in source code). The software is available for download for free, for non-commercial research purposes.

5.8. TiffIO: Qt 3 binding for TIFF images

Participant: Jean-Dominique Gascuel [contact].

TiffIO is a plug-in that add TIFF images read/write capabilities to all Qt3 and Qt4 applications using the reference QImage class. TiffIO come with a self-test suite, and have been compiled and used successfully on a wide variety of systems, compilers and Qt version combination. A demo application enables to quickly test image loading and viewing on any platform. All TIFF operations are based on libtiff 3.8.0, this plugin is just a wrapper that enable to use it transparently from the QImage class, and the architecture defined by Qt.

TiffIO has been downloaded by a large number of developer, and integrated in a variety of commercial or internal tools, such as by Pixar. TiffIO is freely available for download.

5.9. VRender: vector figures

Participant: Cyril Soler [contact].

The VRender library is a simple tool to render the content of an OpenGL window to a vectorial device such as Postscript, XFig, and soon SVG. The main usage of such a library is to make clean vectorial drawings for publications, books, etc.

In practice, VRender replaces the z-buffer based hidden surface removal of OpenGL by sorting the geometric primitives so that they can be rendered in a back-to-front order, possibly cutting them into pieces to solve cycles.

VRender is also responsible for the vectorial snapshot feature of the QGLViewer library. VRender is released under the LGPL licence and is freely available for download.

---

5 http://artis.imag.fr/Publications/2008/0BWBT08
6 http://artis.imag.fr/Software/TiffIO
7 http://artis.imag.fr/Software/VRender
5.10. ProLand

**Participants:** Fabrice Neyret [contact], Eric Bruneton.

ProLand (for procedural landscape) is a software platform originally developed at the Evasion team-project by Eric Bruneton, and currently funded by the ANR-JCJC SimOne. The goal of this platform is the real-time quality rendering and editing of large landscapes. All features can work with planet-sized terrains, for all viewpoints from ground to space. Most of the work published by Eric Bruneton and Fabrice Neyret has been done within ProLand, and a large part has been integrated in the main branch. Several licences have been transferred to companies. A free software version is about to be distributed. Eric Bruneton was hired by Google-Zürich in September 2011, but will be able to keep some participation in the project.

5.11. GigaVoxel

**Participants:** Fabrice Neyret [contact], Morgan Armand, Eric Bruneton, Cyril Crassin, Pascal Guehl, Eric Heitz.

GigaVoxel is a software platform initiated from the PhD work of Cyril Crassin, and currently funded by the ANR CONTINT RTIGE. The goal of this platform is the real-time rendering of very large very detailed scenes. Performances permit showing details over deep zooms and walk through very crowdy scenes (which are rigid, for the moment). The principle is GPU ray-tracing of volumetric-encoded multiscale data with minimal just-in time generation of data (accounting visibility and needed resolution) kept in a cache on GPU. The representation eases the cheap management of soft shadows, depth of field, anti-aliasing and geometric LOD. Beside the representation, data management and base rendering algorithm themself, we also worked on realtime light transport, and on quality prefiltering of complex data. This work led to numerous publications ([16], [22], [23]). Several licences have been sold to companies. A free software version is about to be distributed.
AVIZ Project-Team

5. Software

5.1. The Obvious Toolkit

Participants: Pierre-Luc Hémery, Jean-Daniel Fekete [correspondant].

The Obvious Toolkit is a new Interactive Graphics Toolkit written in Java to facilitate the interoperability between Information Visualization toolkits and components (Fig. 1).

The Obvious Toolkit is an abstraction layer above visualization toolkits. Currently, it connects the most popular toolkits in Java: Prefuse, the InfoVis Toolkit, Improvise, as well as other libraries such as the Java Database Communication Toolkit (JDBC) and some others.

It is meant to provide an abstraction layer for information visualization application builders so that they can postpone their choice of a concrete toolkit to use. When faced with the final choice, application builders can use one of the toolkits or connect all of them dynamically to Obvious. Obvious is available at http://code.google.com/p/obvious. A paper on Obvious was presented at the IEEE Visual Analytics Science and Technology conference (VAST 2011).

5.2. GeneaQuilts

Participants: Jean-Daniel Fekete [correspondant], Pierre Dragicevic, Anastasia Bezerianos, Julie Bae, Ben Watson, Maike Gilliot [correspondant].

GeneaQuilts is a new genealogy exploration software that allows genealogists and historians to visualize and navigate in large genealogies of up to several thousand individuals (Fig. 2). The visualization takes the form of a diagonally-filled matrix, where rows are individuals and columns are nuclear families. The GeneaQuilts system includes an overview, a timeline, search and filtering components, and a new interaction technique called Bring & Slide that allows fluid navigation in very large genealogies. The tool has been featured in several InfoVis and genealogy Websites and the website has been visited over 9000 times.
Figure 2. The genealogy of the Simpsons family (left) and of the Greek Pantheon (right), produced by the GeneaQuilts software.

See also the web page http://www.aviz.fr/geneaquilts/.

- Version: 1.0.4

5.3. Diffamation

Participants: Fanny Chevalier, Pierre Dragicevic [correspondant], Anastasia Bezerianos, Jean-Daniel Fekete.

The Diffamation system [3] allows rapid exploration of revision histories such as Wikipedia or subversion repositories by combining text animated transitions with simple navigation and visualization tools. Diffamation can be used for example to get a quick overview of the entire history of a Wikipedia article or to see what has happened to one’s contributions. Diffamation complements classical diff visualizations: once moments of interest have been identified, classical diff visualizations can come in useful to compare two given revisions in detail.
The Diffamation revision exploration system is available at http://www.aviz.fr/diffamation/. It has been presented at the plenary session of the Ubuntu Developer Summit.

5.4. The InfoVis Toolkit

**Participant:** Jean-Daniel Fekete [correspondant].

The InfoVis Toolkit [5] is an Interactive Graphics Toolkit written in Java to facilitate the development of Information Visualization applications and components.

The InfoVis Toolkit implements several visualization techniques, as well as interaction techniques related. It has been used for teaching the Information Visualization course (Masters level, Univ. of Paris-Sud) and is the basis for all AVIZ contracts. It is our main development platform for information visualization; most of our Information Visualization prototypes rely on it. It is available at http://ivtk.sourceforge.net.

In the forthcoming years, it will be superseded by extensions of the Obvious Toolkit (see 5.1).

- Version: version0.9 beta 2

5.5. GraphDice

**Participants:** Jean-Daniel FEKETE [correspondant], Pierre Dragicevic, Niklas Elmqvist, Anastasia Bezerianos.

GraphDice [1] is a visualization system for exploring multivariate networks (Fig. 4). GraphDice builds upon our previous system ScatterDice (best paper award at the IEEE InfoVis 2008 conference) [4]: it shows a scatter plot of 2 dimensions among the multiple ones available and provides a very simple paradigm of 3D rotation to change the visualized dimensions. The navigation is controlled by a scatter plot matrix that is used as a high-level overview of the dataset as well as a control panel to switch the dimensions.

While ScatterDice works on any tabular dataset (e.g., CSV file), the GraphDice system show networks using a node-link diagram representation as a scatter plot with links drawn between connected nodes. See the web page http://graphdice.gforge.inria.fr for more information.

- Version: version 1.0

5.6. Gliimpse

**Participants:** Pierre Dragicevic [correspondant], Stéphane Huot, Fanny Chevalier.
Gliimpse is a quick preview technique that smoothly transitions between document markup code (HTML, LaTeX,...) and its visual rendering. This technique allows users to regularly check the code they are editing in-place, without leaving the text editor. This method can complement classical preview windows by offering rapid overviews of code-to-document mappings and leaving more screen real-estate. A proof-of-concept editor can be downloaded for free at http://www.aviz.fr/gliimpse/.
4. Software

4.1. Introduction
From its creation, AxIS has proposed new methods, approaches and software validated experimentally on various applications: Data Mining, Web usage Mining, Information Retrieval, Activity Modeling. Some of our results are under process to be part of the FocusLab platform (CPER Télus 5.5.2) which is based on a Service oriented Architecture. The development process has started this year, finding ways to fund human resources. Such a platform aims the community of Living Labs domain.

4.2. Data Mining

4.2.1. Classification and Clustering Methods

**Participants:** Marc Csernel, Yves Lechevallier [co-correspondant], Brigitte Trousse [co-correspondant].

We developed and maintained a collection of clustering and classification software, written in C++ and/or Java:

**Supervised methods**
- a Java library (Somlib) that provides efficient implementations of several SOM (Self-Organizing Map) variants [87], [86], [106], [105], [110], especially those that can handle dissimilarity data (available on Inria’s Gforge server (public access) https://gforge.inria.fr/projects/somlib/, developed by AxIS Rocquencourt and Brieuc Conan-Guez from Université de Metz.
- a functional Multi-Layer Perceptron library, called FNET, that implements in C++ supervised classification of functional data [101], [104], [103], [102] (developed by AxIS Rocquencourt).

**Unsupervised methods: partitioning methods**
- two partitioning clustering methods on the dissimilarity tables issued from a collaboration between AxIS Rocquencourt team and Recife University, Brazil: CDis and CClimst [111]. Both are written in C++ and use the “Symbolic Object Language” (SOL) developed for SODAS. And one partitioning method on interval data (Div).
- two standalone versions improved from SODAS modules, SCluster and DIVCLUS-T [84] (AxIS Rocquencourt).

**Unsupervised methods: agglomerative methods**
- a Java implementation of the 2-3 AHC (developed by AxIS Sophia Antipolis). The software is available as a Java applet which runs the hierarchies visualization toolbox called HCT for Hierarchical Clustering Toolbox (see [85]).

A Web interface developed in C++ and running on our Apache internal Web server is available for the following methods: SCluster, Div, yCdis, CClimst.

Previous versions of the above software have been integrated in the SODAS 2 Software [98] which was the result of the European project ASSO 5 (2001-2004). SODAS 2 softsodaslinkware supports the analysis Stof multidimensional complex data (numerical and non numerical) coming from databases mainly in statistical offices and administration using Symbolic Data Analysis [82]. This software is registered at APP. The latest executive version of the SODAS 2 software, with its user manual can be downloaded at http://www.info.fundp.ac.be/asso/sodaslink.htm [88], [112].

---

5 ASSO: Analysis System of Symbolic Official data
4.2.2. Extracting Sequential Patterns with Low Support

Participants: Brigitte Trousse [correspondant].

Two methods for extracting sequential patterns with low support have been developed by D. Tanasa in his thesis (see Chapter 3 in [108] for more details) in collaboration with F. Masseglia and B. Trousse:

- **Cluster & Divide** [108]
- **and Divide & Discover** [13], [108].

4.2.3. Mining Data Streams

Participants: Brigitte Trousse [correspondante], Mohamed Gaieb.

In Marascu’s thesis (2009) [95], a collection of software have been developed for knowledge discovery and security in data streams. Three clustering methods for mining sequential patterns (Java) in data streams have been developed in Java:

- **SMDS** compares the sequences to each others with a complexity of $O(n^2)$.
- **SCDS** is an improvement of SMDS, where the complexity is enhanced from $O(n^2)$ to $O(n.m)$ with $n$ the number of navigations and $m$ the number of clusters.
- **ICDS** is a modification of SCDS. The principle is to keep the clusters’ centroids from one batch to another.

Such methods take batches of data in the format “Client-Date-Item” and provide clusters of sequences and their centroids in the form of an approximate sequential pattern calculated with an alignment technique.

In 2010 the Java code of one method called SCDS has been integrated in the MIDAS demonstrator (cf. 6.2.1) and a C++ version has been implemented by F. Masseglia for the CRE contract with Orange Labs with the deliverability of a licence) with a visualisation module (in Java).

It has been tested on the following data:

- Orange mobile portal logs (100 million records, 3 months) in the context of Midas project (Java version) and the CRE (Orange C++ version)
- Inria Sophia Antipolis Web logs (4 million records, 1 year, Java version)
- Vehicle trajectories (Brinkhoff generator http://iapg.jade-hs.de/personen/brinkhoff/generator/ in the context of MIDAS project (Java version)

This year it has been integrated as a Web service (Java version) in the first version of FocusLab platform in the ELLIOT context (cf. 5.5.2); a demonstration was made on San Rafaelle Hospital media use case at the first ELLIOT review at Brussels (cf. 6.3.1.1).

4.3. Web Usage Mining

4.3.1. AWLH for Pre-processing Web Logs

Participants: Yves Lechevallier [co-correspondant], Brigitte Trousse [co-correspondant].

AWLH (AxIS Web Log Preprocessing and Data Stream extraction) for Web Usage Mining (WUM) is issued from AxISlogminer preprocessing software which implements the multi-site log preprocessing methodology developed by D. Tanasa in his thesis [16] for Web Usage Mining (WUM). In the context of the Eiffel project (2008-2009), we isolated and redesigned the core of AxISlogMiner preprocessing tool (we called it AWLH) composed of a set of tools for pre-processing web log files. AWLH can extract and structure log files from several Web servers using different input format. The web log files are cleaned as usually before to be used by data mining methods, as they contain many noisy entries (for example, robots bring a lot of noise in the analysis of user behaviour then it is important in this case to identify robot requests). The data are stored within a database whose model has been improved.
Now the current version of our Web log processing (Available on INRIA's gforge website with private access) offers:

- Processing of several log files from several servers,
- Support of several input formats (CLF, ECLF, IIS, custom, ...);
- Incremental pre-processing;
- Java API to help integration of AWLH in external application.

An additional tool has been developed for capturing user actions in real time based on an open source project called "OpenSymphony ClickStream". An extension version of AWLH called AWLH-Debate has been developed for recording and structuring data issued from annotated documents inside discussion forums.

### 4.3.2. ATWUEDA for Analysing Evolving Web Usage Data

**Participants:** Yves Lechevallier [correspondant], Brigitte Trousse, Mohamed Gaieb, Yves Lechevallier.

ATWUEDA for Web Usage Evolving Data Analysis [90] was developed by A. Da Silva in her thesis [89] under the supervision of Y. Lechevallier. This tool was developed in Java and uses the JRI library in order to allow the application of R which is a programming language and software environment for statistical computing [http://www.r-project.org/](http://www.r-project.org/) functions in the Java environment.

ATWUEDA is able to read data from a cross table in a MySQL database. It splits the data according to the user specifications (in logical or temporal windows) and then applies the approach proposed in the Da Silva’s thesis in order to detect changes in dynamic environment. The proposed approach characterizes the changes undergone by the usage groups (e.g. appearance, disappearance, fusion and split) at each timestamp. Graphics are generated for each analyzed window, exhibiting statistics that characterizes changing points over time.


This year we have demonstrated the efficiency of ATWUEDA [51] by applying it on another real case study on condition monitoring data streams of an electric power plant provided by EDF (cf. section 5.5.1).

ATWUEDA is used by Telecom Paris Tech and EDF [51].

### 4.4. Information Retrieval

#### 4.4.1. CBR*Tools for Managing and Reusing Past Experiences based on Historical Data

**Participant:** Brigitte Trousse [correspondant].

CBR*Tools [92], [93] is an object-oriented framework [94], [91] for Case-Based Reasoning which is specified with the UMT notation (Rational Rose) and written in Java. It offers a set of abstract classes to model the main concepts necessary to develop applications integrating case-based reasoning techniques: case, case base, index, measurements of similarity, reasoning control. It also offers a set of concrete classes which implements many traditional methods (closest neighbors indexing, Kd-tree indexing, neuronal approach based indexing, standards similarities measurements). CBR*Tools currently contains more than 240 classes divided in two main categories: the core package for basic functionality and the time package for the specific management of the behavioral situations. The programming of a new application is done by specialization of existing classes, objects aggregation or by using the parameters of the existing classes.

CBR*Tools addresses application fields where the re-use of cases indexed by behavioral situations is required. The CBR*Tools framework was evaluated via the design and the implementation of several applications such as Broadway-Web, Educaid, BeCKB, Broadway-Predict, e-behaviour and Be-TRIP.

CBR*Tools is concerned by two past contracts: EPIA and MobiVIP.

CBR*Tools will be available for research, teaching and academic purpose via the FocusLab platform. The user manual can be downloaded at the URL: [http://www-sop.inria.fr/axis/cbrtools/manual/](http://www-sop.inria.fr/axis/cbrtools/manual/).
See also the web page http://www-sop.inria.fr/axis/cbrtools/manual/.

4.4.2. Broadway*Tools for Building Recommender Systems on the Web

Participant: Brigitte Trousse [correspondant].

Broadway*Tools is a toolbox supporting the creation of adaptive recommendation systems on the Web or in an Internet/Intranet information system. The toolbox offers different servers, including a server that computes recommendations based on the observation of the user sessions and on the re-use of user groups’ former sessions. A recommender system created with Broadway*tools observes navigations of various users and gathers evaluations and annotations, to draw up a list of relevant recommendations (Web documents, keywords, etc).

Based on Jaczynski’s thesis [92], different recommender systems have been developed for supporting Web browsing, but also browsing inside a Web-based information system or for query formulation in the context of a meta search engine.

4.5. Activity Modeling

4.5.1. K-MADe for Describing Human Operator or User Activities

Participant: Dominique Scapin [correspondant].

K-MADe tool (Kernel of Model for Human Activity Description Environment). The K-MADe is intended for people wishing to describe, analyze and formalize the activities of human operators, of users, in environments (computerized or not), in real or simulated situation; in the field, or in the laboratory. Although all kinds of profiles of people are possible, this environment is particularly intended for ergonomics and HCI (Human Computer Interaction) specialists. It has been developed through collaboration between ENSMA (LISI XSlaboratory) and INRIA. The last release was delivered on November 1st 2010 based on the work of Caffiau and al. [83].

Its history, documentation and tool are available at: http://kmade.sourceforge.net/index.php
5. Software

5.1. Introduction

Software development is an essential part of the research done by COPRIN since a large part of our methods can only be validated experimentally. Software developments follow various directions:

1. interval arithmetic: although we do not plan to work in this very specialized area (we generally rely on existing packages) interval arithmetic is an important part of our interval analysis algorithms and we may have to modify the existing packages so as to deal, in particular, with multi-precision and arithmetic extensions

2. interval analysis libraries: we daily use two libraries that have been designed in the project and are still under development. A long term work is to develop a generic programming framework that allows for modularity and flexibility, with the objectives of testing new functionalities easily and building specific solvers by a simple juxtaposition of existing modules

3. interface to interval analysis: in our opinion interval analysis software must be available within general purpose scientific software (such as Maple, Mathematica, Scilab) and not only as a stand-alone tool. Indeed most end-users are reluctant to learn a new programming language just to solve problems that are only small elements of a more general problem. Furthermore interval analysis efficiency may benefit from the functionalities available in the general purpose scientific software.

5.2. Interval analysis libraries

5.2.1. ALIAS

Participants: David Daney, Jean-Pierre Merlet [correspondant], Odile Pourtallier.

The ALIAS library (Algorithms Library of Interval Analysis for Systems), whose development started in 1998, is a collection of procedures based on interval analysis for systems solving and optimization.

ALIAS is made of two parts:

- ALIAS-C++: the C++ library (87,000 code lines) which is the core of the algorithms
- ALIAS-Maple: the Maple interface for ALIAS-C++ (55,000 code lines). This interface allows one to specify a solving problem within Maple and get the results within the same Maple session. The role of this interface is not only to generate the C++ code automatically, but also to perform an analysis of the problem in order to improve the efficiency of the solver. Furthermore, a distributed implementation of the algorithms is available directly within the interface.

These libraries can be freely downloaded.

5.2.2. Int4Sci : a Scilab interface for interval analysis

Participants: David Daney, Gilles Trombettoni, Bertrand Neveu.

In 2006, we have started the development of a Scilab interface to C++ Bias/Profil interval arithmetic package and to the library ALIAS. The first version of Int4Sci has been released in 2008 – see http://www-sop.inria.fr/coprin/logiciels/Int4Sci/ for Linux, MacOS and Windows. A second version, compatible with Scilab 5.3 is in preparation. This interface provides an interval arithmetic, basic interval manipulation tools as well as the solving of linear interval systems. All functions are documented and a tutorial is available.

5.2.3. Mathematica Interface to Interval Analysis

Participants: Yves Papegay [correspondant], Jean-Pierre Merlet.
Since 2006, we have been implementing in Mathematica a high-level modular interface to the ALIAS library. The initial aim of providing the Mathematica users community a transparent access to the functionalities of ALIAS, and of extending the dissemination of our library, has progressively turned into the aim of providing ALIAS advanced users and developers with a high-level modular interface for prototyping, easy testing and quick implementation of new interval analysis algorithms and procedures relying on symbolic computation skills. This includes symbolic preprocessing of expressions, and symbolic specializations of interval analysis algorithms.
DAHU Project-Team (section vide)
5. Software

5.1. Introduction

The pieces of software described in this section are prototypes implemented by members of the project. They are not available through the APP. Any interested person should contact relevant members of the project.

5.2. QTempIntMiner: quantitative temporal sequence mining

QTempIntMiner (Quantitative Temporal Interval Miner) is a software that implements several algorithms presented in [42] and [8].

The software is mainly implemented in Matlab. A standalone application is now available. It uses the Mixmod toolbox [28] to compute multi-dimensional Gaussian distributions. The main features of QTempIntMiner are:

- a tool for generating synthetic noisy sequences of temporal events,
- an implementation of the QTempIntMiner, QTIAPriori and QTIPrefixSpan algorithms,
- a graphical interface that enables the user to generate or import data set and to define the parameters of the algorithm and that displays the extracted temporal patterns.
- a sequence transformer to process long sequences of temporal events. Long sequences are transformed into a database of short temporal sequences that are used as input instances for the available algorithms.

This year the software has been updated to include two new algorithms: QTIAPriori and QTIPrefixSpan. The software has been used to compare the efficiency of three algorithms. The software is currently applied to the characterization of cardiac arrhythmias.

The following website gives many details about the algorithms and provides the latest stable implementation of QTempIntMiner: http://www.irisa.fr/dream/QTempIntMiner/.

5.3. Sacadeau: qualitative modeling and decision-aid to preserve the water quality from pollutants as herbicides

Sacadeau is a software that implements the Sacadeau transfer model presented in section 8.2.1. The Sacadeau simulation model couples two qualitative models, a transfer model describing the pesticide transfer through the catchment and a management model describing the farmer decisions. Giving as inputs a climate file, a topological description of a catchment, and a cadastral repartition of the plots, the Sacadeau model simulates the application of herbicides by the farmers on the maize plots, and the transfer of these pollutants through the catchment until the river. The two main simulated processes are the runoff and the leaching. The output of the model simulation is the quantity of herbicides arriving daily to the stream and its concentration at the outlets. The originality of the model is the representation of water and pesticide runoffs with tree structures where leaves and roots are respectively up-streams and down-streams of the catchment.

The software allows the user to see the relationships between these tree structures and the rules learnt from simulations. A more elaborated version allows to launch simulations and to learn rules on-line. This year, we have developed this new version by enabling access to two recommendation action algorithms (see section 6.3.5). The user can choose different parameters (set of classification rules from which actions will be built, parameters concerning action feasibility, etc) before asking for action recommending process, and then easily visualize the characteristics of situations to improve (polluted ones) compared with the different recommended actions. The software is mainly in Java.
5.4. Ecomata

We have proposed a new qualitative approach for ecosystem modeling based on timed automata (TA) formalism combined to a high-level query language for exploring scenarios. EcoMata is a tool-box for modeling and exploring qualitatively trophic-food web using this approach. To date, it is dedicated to ecosystems that can be modeled as a collection of species (prey-predator systems) under various human pressures and to environmental disturbances. This tool is made of two main parts: the Network Editor and the Query Launcher. The Network Editor let a stakeholder describe the trophic food web in a graphical way (the species icons and interactions between them). Only few ecological parameters are required and the user can save species in a library. The number of qualitative biomass levels is set as desired. An efficient algorithm generates automatically the network of timed automata. EcoMata provides also a dedicated window to help the user to define different fishing pressures, a nice way being by using chronograms. In the Query Launcher, the user selects the kind of query and the needed parameters (for example the species biomass levels to define a situation). Results are provided in a control panel or in files that can be exploited later. Several additional features are proposed in EcoMata: building a species library, import/export of ecosystem model, batch processing for long queries, etc. EcoMata is developed in Java (Swing for the GUI) and the model-checker called for the timed properties verification is UPPAAL.

The following website is devoted to the presentation of the ECOMATA:  http://oban.agrocampus-ouest.fr:8080/ecomata .
4. Software

4.1. PROTEUS

Participants: Amaury Nègre, Juan Lahera-Perez.

This toolkit offers an automatic mobile robot driver, some sensors drivers (sensors as Sick laser, GPS, motion tracker, mono or stereo camera), and a 3D Simulator.

The latest developments have been focused on the robotics simulator. This simulator is based on the simulation and 3D rendering engine "mgEngine" (http://mgengine.sourceforge.net/) embedded with the physics engine "bullets physics" (http://bulletphysics.org) for realistic robot dynamic simulation. We also worked on the interface with the robotics middleware "ROS" (http://www.ros.org) in order to offer interoperability with many robotics applications. This software is developed in C++ and the simulator operates with the Lua scripting language.

The simulation software is used in the ANR Proteus (http://www.anr-proteus.fr), as a simulation engine for the PROTEUS Toolkit.

![Figure 1. Screenshot of the Mobile Robot Simulator. Simulation of a Cycab robot in the "Pavin" environment provided by the LASMEA.](image)

- Version: 2.0
- APP: IDDN.FR.001.510040.000.S.P.2005.000.10000
- Programming language: C/C++, Lua

4.2. AROSDYN

Participants: Igor Paromtchik, Mathias Perrollaz, Alex Christos Makris, Amaury Nègre, Christian Laugier.
ArosDyn (http://arosdyn.gforge.inria.fr/) is a system which integrates our recently developed techniques to provide a real-time collision risk estimation in a dynamic environment. The main features of this software are:

1. The deliberate design provides high maintainability, scalability and reusability of the models and algorithms.
2. The software has a user interface (UI) which is user-friendly.
3. The software facilitates the parameter tuning of the models.
4. It uses the GPU to accelerate the computation.
5. Working together with the Hugr middleware (http://gforge.inria.fr/projects/cycabtk), it can run on our experimental vehicle in real-time.

The software is developed in C/C++ in Linux and its architecture is shown in Fig. 2.

In this example, we demonstrate a typical sensor fusion application. We retrieve the raw data from the Hugr middleware and store them in individual sensor objects. Then, by using this framework, we integrate the IBEO Bayesian Occupancy Filter (BOF) sensor model, the stereo sensor processor model, the stereo BOF sensor model and the BOF model together. Finally, different aspects of the computational results are visualized in several viewers. At the same time, all the parameters used by the algorithms can be tuned online.

Several windows of this application are shown in Fig. 3. Here we demonstrate the main window, the 2D viewer of the stereo camera and the lidar, the disparity map of the stereo vision and the compounded BOF grid which is the result of the sensor fusion.

Another important property of this software is a large part of the computation task executed on GPU. As the processing of stereo image and the computation in the BOF can be highly parallelized, we run these tasks on the GPU to improve the time performance, as shown in Fig. 4. In this way, the software can work in real-time.
Figure 3. Windows of the ArosdynTestSuite software
Perception, Cognition, Interaction - Software - Project-Team E-MOTION

Figure 4. Time performance of BOF on GPU

The GPU calculation is based on CUDA library and is carried out in an independent thread. The schematic graph of the GPU computational thread is shown in Fig. 5.

Figure 5. The GPU computational thread

Furthermore, thanks to the deliberated design of the software, we can easily add new models to it and let them work together. The fast detection and tracking algorithm (FCTA) and the Gaussian process based collision assessment algorithm are added into this framework.

4.3. Bayesian Occupancy Filter


The BOF toolbox is a C++ library that implements the Bayesian Occupancy Filter. It is often used for modelling dynamic environments. It contains the relevant functions for performing bayesian filtering in grid spaces. The output from the BOF toolbox are the estimated probability distributions of each cell’s occupancy and velocity. Some basic sensor models such as the laser scanner sensor model or Gaussian sensor model for gridded spaces are also included in the BOF toolbox. The sensor models and BOF mechanism in the BOF toolbox provides the necessary tools for modelling dynamic environments in most robotic applications. This toolbox is patented under two patents: “Procédé d’assistance à la conduite d’un véhicule et dispositif associé”
n. 0552735 (9 september 2005) and “Procédé d’assistance à la conduite d’un véhicule et dispositif associé amélioré” n. 0552736 (9 september 2005) and commercialized by ProBayes.

- Version: 1
- Programming language: C/C++

4.4. PROBT

People involved: Juan-Manuel Ahuactzin, Kamel Mekhnacha, Pierre Bessière, Emmanuel Mazer, Manuel Yguel, Christian Laugier.

ProBT is both available as a commercial product (ProBAYES.com) and as a free library for public research and academic purposes (http://emotion.inrialpes.fr/BP/spip.php?rubrique6). Formerly known as OPL, ProBT is a C++ library for developing efficient Bayesian software. It is available for Linux, Unix, PC Windows (Visual C++), MacOS9, MacOSX and Irix systems. The ProBT library (http://www.probayes.com/) has two main components: (i) a friendly Application Program Interface (API) for building Bayesian models, and (ii) a high-performance Bayesian Inference Engine (BIE) allowing to execute all the probability calculus in exact or approximate way. ProBT is now commercialized by our start-up Probayes; it represents the main Bayesian programming tool of the e-Motion project-team, and it is currently used in a variety of external projects both in the academic and industrial field (e.g. for the European project BACS and for some industrial applications such as Toyota or Denso future driving assistance systems).
5. Software

5.1. Corese

**Participants:** Olivier Corby [correspondant], Fabien Gandon.

Corese\(^1\) (COnceptual REsource Search Engine) is a Semantic Web Factory. It enables users to load and process RDFS schemas, RDF metadata and to query the base of annotations thus created, by using the SPARQL Query Language.

Corese implements RDF, RDFS and SPARQL 1.1 Query Language & Update. Furthermore, Corese query language integrates original features such as approximate search, SQL or XPath. Approximate search consists of searching the best approximate answers to a query according to the ontology types. Corese also integrates a SPARQL-based Rule Language for RDF.

Corese is a Semantic Web Factory that enables us to design and develop Semantic Web applications; it is available for download. In the past, Corese benefited from an INRIA software development support (ADT) with two software engineers. Corese is registered at the APP and in 2007 we decided to distribute it as open source software under license CeCILL-C.

Corese is used and has been used in more than 45 applications, 21 PhD Thesis and is used for education by 8 institutions. It is used as a Semantic Factory in such projects as Ontorule, Palette, SevenPro and SeaLife european projects, in e-WOK Hub, Neurolog, ISICIL and Kolflow ANR projects, BioMarker and KmP projects, Semantic Web Import Plugin for Gephi visualization and ECCO ontology editor. The work on Corese was published in [57], [58], [59], [56], [1], [5], [3], [2], [4].

This year we released a major new version 3.0 based on the KGRAM SPARQL 1.1 interpreter. KGRAM (see 6.1.1) is a generic SPARQL interpreter that can query not only RDF but also labeled graphs.

Web page: [http://www.inria.fr/sophia/edelweiss/software/corese](http://www.inria.fr/sophia/edelweiss/software/corese)

5.2. Semantic Web Import Plugin for Gephi visualization

**Participants:** Erwan Demairy [correspondant], Fabien Gandon, Olivier Corby.

The SemanticWebImport\(^2\) plugin is intended to allow the import of semantic data into Gephi open graph visualisation platform. Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs. The imported data are obtained by processing a SPARQL request on the semantic data. The data can be accessed following three manners:

1. by accessing local RDF & RDFS files and using the embedded Corese engine to apply the SPARQL request;
2. by accessing a remote REST SPARQL endpoint. In that case, the SPARQL request is applied remotely and the graph is built locally by analyzing the result sent by the endpoint;
3. by accessing a remote SOAP SPARQL endpoint. As for the REST endpoint, the resulting graph is built from the result returned by the endpoint.

The software is released under version 1.0.


---

\(^1\)http://www.inria.fr/sophia/edelweiss/software/corese

\(^2\)http://wiki.gephi.org/index.php/SemanticWebImport
5.3. ISICIL

Participants: Sébastien Comos, Nicolas Delaforge, Fabien Gandon [resp.].

The ISICIL software platform is made of several software components:

- XUL (XML-based User interface Language) extensions for the Firefox browser to assist the technology watch and business intelligence tasks by collecting relevant metadata according to the navigation context of the user.
- An application server based on Tomcat publishes services using the REST protocol to process requests of the users’ applications and in particular the navigation extensions.

This architecture is summarized in Figure 1. Its major interest lies in the flexibility introduced by the loose coupling between REST services and navigators extensions or other applications.

In the context of the ISICIL ANR project, we have developed a Semantic Web server which provides core services to manage simple tagging of resources (internal or from the Web) and to assist the semantic enrichment of the folksonomy of our communities of users. This server’s implementation is based on the ISICIL main framework. The tagging model combines already existing ontologies such as SIOC3, SCOT, and Newman’s Tag Ontology4 as shown in Figure 2. SRTag, the model of folksonomy enrichment, is based on a named graph mechanism in order to maintain diverging statements made between tags using SKOS (for thesaurus like relation between tags) or SCOT (for spelling variant relations), and is shown in Figure 3.

---

3 http://sioc-project.org
4 http://www.holygoat.co.uk/owl/redwood/0.1/tags
Figure 2. Model of tagging used in the Semantic Tag Server

Figure 3. Folksonomy enrichment model
The functionalities of this server can be divided into three categories:

1. Tagging: creating a tag; get tag suggestions based on the input characters; create a tagging, i.e. a link between a resource, a user, and a tag.
2. Computing: an external library (exported as a java jar file) has been developed to perform computations on the tagging data. Two types of computations have been implemented:
   1. Spelling Variant detection.
   2. Related tag detection based on the computation of the similarity between tags [63].
3. Managing Semantic relations between tags: get semantically related tags, reject or propose new semantic relations.

We developed a Firefox extension to help users navigate within a folksonomy and organize semantically the tags. The main idea behind this tool is to combine organization tasks with everyday tasks in the least intrusive way, that is to say, without forcing the user in any way, and by providing a user friendly graphical interface. This tool, developed using the XUL framework⁵, is supported by the SRTag model and the Semantic Tag Server. Users are provided with a search bar for navigating the folksonomy. When available, other tags are suggested and ordered according to their semantic relation with the searched tag (broader, narrower, related, spelling variant). Each suggestion can be either:
   - clicked to search content tagged with this tag;
   - rejected by clicking a checkbox;
   - modified thanks to a drag-and-drop mechanism where a tag can be dropped in another category of semantic relation.

Web page: https://gforge.inria.fr/projects/isicil/

⁵ http://developer.mozilla.org/en/XUL
5. Software

5.1. MyCorporisFabrica

Participants: Ali-Hamadi Dicko, François Faure, Olivier Palombi.

My Corporis Fabrica (MyCF) is an anatomical knowledge database (see fig. 1). During 2011, we have added new anatomical entities and improved some parts of FMA (Foundational Model of Anatomy). The FMA's license is now under Creative Commons licenses (CC-by: Licensees may copy, distribute, display and perform the work and make derivative works based on it only if they give the author or licensor the credits in the manner specified by these). The license of MyCF is not yet defined. Our new contribution this year, is the creation of a brand new ontology about human functions. Based on the International Classification of Functioning, Disability and Health, also known as ICF, we have organized human functions through a tree of 4330 items. A original journal paper must be submitted soon. MyCF browser is now available on line: http://www.mycorporisfabrica.org/. The MyCF’s generic programming framework can be used for other domains. The link with semantic and 3D models matches research activities of IMAGINE towards interactive digital creation media. Anatomy can be seen as a study case.

Figure 1. My Corporis Fabrica is an anatomical knowledge database developed in our team.

5.2. SOFA

Participants: Guillaume Bousquet, Ali Hamadi Dicko, François Faure, François Jourdes.
SOFA is a C++ library primarily targeted at medical simulation research. Based on an advanced software architecture, it allows to (1) create complex and evolving simulations by combining new algorithms with algorithms already included in SOFA; (2) modify most parameters of the simulation – deformable behavior, surface representation, solver, constraints, collision algorithm, etc. – by simply editing an XML file; (3) build complex models from simpler ones using a scene-graph description; (4) efficiently simulate the dynamics of interacting objects using abstract equation solvers; and (5) reuse and easily compare a variety of available methods. The GPU capabilities of SOFA have been demonstrated at a SIGGRAPH talk [16] (see fig. 2) and presented in a book chapter [27].

SOFA is currently used by company Digital Trainers to develop basic skill endoscopic simulators. A start-up company based on SOFA, InSimo, is being created in the Strasbourg IHU, and is expected to start in first semester 2012.

\[ \text{Figure 2. GPU methods in SOFA for detailed deformable objects at interactive rates.} \]

5.3. AESTEM Studio

Participants: Adrien Bernhardt, Marie-Paule Cani, Maxime Quiblier.

AESTEM Studio is dedicated to free form shape Modeling through Interactive Sketching and Sculpting gestures. The goal is to provide a very intuitive way to create 3D shapes, as easy to use for the general public as roughly sketching a shape or modeling it with a piece of clay. This software is developed in the framework of a research contract with the company Axiatec. It enables to create a 3D shape by successively painting in 2D and smoothly blending different components: the painting step takes place at different scales and from different viewing angles. 3D is inferred from a 2D painted region by using an isotropic implicit surface along the skeleton of the region. Then, implicit blending, restricted to the intersection areas, is computed to connect the new component with the existing ones. This relies on our researches on free-form sketch-based modeling using geometric skeletons and on convolution surfaces. Our prototype is written in C++. It uses the Ogre open-source library and our new library Convol dedicated to convolution surfaces. Future extensions will include the combination of sketching with modeling gestures related to clay sculpting, such as deforming a shape through pulling, pushing, bending or twisting gestures.

5.4. Convol

Participants: Marie-Paule Cani, Maxime Quiblier, Cédric Zanni.
Convol is a new C++ library we develop for easing our work on implicit surfaces – and more particularly on the sub-class of convolution surfaces. It enables us to make our latest research results soon available to the rest of the group and easily usable in our industrial partnerships. Convol incorporates all the necessary material for constructive implicit modeling: skeleton-based distance and convolution primitives, with closed form solution for the field values and gradient whenever possible; a variety of blending operators; and several methods for tessellating an implicit surface into a mesh, and for refining the later in highly curved regions. This development is funded by INRIA as support to our research group.
5. Software

Exmo's work can be implemented in software: in particular, we have developed an API for expressing ontology alignment (§ 5.1) and a library of ontology distances and similarities OntoSim (§ 5.2).

5.1. Alignment API: manipulating ontology alignments

Participants: Jérôme Euzenat [Contact], Jérôme David, Cássia Trojahn dos Santos.

We have designed a format for expressing alignments in a uniform way [1]. The goal of this format is to be able to share available alignments on the web. It should help systems using alignments, e.g., mergers, translators, to take advantage of any alignment algorithm and it will help alignment algorithms to be used in many different tasks. This format is expressed in RDF, so it is freely extensible, and has been defined by a DTD (for RDF/XML), an OWL ontology and an RDF Schema.

The API itself [3] is a Java description of tools for accessing the common format. It defines five main interfaces (OntologyNetwork, Alignment, Cell, Relation and Evaluator) and proposes the following services:

- Storing, finding, and sharing alignments;
- Piping matching algorithms (improving an existing alignment);
- Manipulating alignments (thresholding and hardening);
- Generating processing output (transformations, axioms, rules);
- Comparing alignments.

We provide an implementation for this API which can be used for producing transformations, rules or bridge axioms independently from the algorithm which produced the alignment. The proposed implementation features:

- a base implementation of the interfaces with all useful facilities;
- a library of sample matchers;
- a library of renderers (XSLT, SWRL, OWL, C-OWL, SEKT mapping language);
- a library of evaluators (various generalisation of precision/recall, precision/recall graphs);
- a library of wrapper for several ontology API;
- a parser for the format.

To instantiate the API, it is sufficient to refine the base implementation by implementing the `align()` method. Doing so, the new implementation will benefit from all the services already implemented in the base implementation.

We have developed on top of the Alignment API an Alignment server that can be used by remote clients for matching ontologies and for storing and sharing alignments. It is developed as an extensible platform which allows to plug-in new interfaces. The Alignment server can be accessed through HTML, web service (SOAP and REST) and agent communication interfaces.

This year, within the SEALS project (see § 8.2.1), we have developed a flexible test generation framework within the Alignment API which allows for generating new evaluation datasets [12].

The Alignment API is used in the Ontology Alignment Evaluation Initiative data and result processing. It is also used by more than 30 other teams worldwide.

The Alignment API is freely available since December 2003 under the LGPL licence at http://alignapi.gforge.inria.fr
5.2. The OntoSim library

Participants: Jérôme David [Contact], Jérôme Euzenat.

OntoSim is a library offering similarity and distance measures between ontology entities as well as between ontology themselves. It materialises our work towards better ontology proximity measures.

There are many reasons for measuring a distance between ontologies. For example, in semantic social networks, when a peer looks for a particular information, it could be more appropriate to send queries to peers having closer ontologies because it will be easier to translate them and it is more likely that such a peer have the information of interest. OntoSim provides a framework for designing various kinds of similarities. In particular, we differentiate similarities in the ontology space from those in the alignment space. The latter ones make use of available alignments in an ontology network while the former only rely on ontology data. OntoSim is provided with 4 entity measures which can be combined using various aggregation schemes (average linkage, Hausdorff, maximum weight coupling, etc.), 2 kinds of vector space measures (boolean and TF.IDF), and 4 alignment space measures. In addition, the framework embeds external similarity libraries which can be combined to our own.

This year, we have implemented the measures of agreement and disagreement between ontologies recently proposed by Mathieu d’Aquin (Open university).

OntoSim is based on an ontology interface allowing for using ontology parsed with different APIs. OntoSim is written in Java and is available under the LGPL license at http://ontosim.gforge.inria.fr.
5. Software

5.1. Perception Tools

Participants: David Filliat [correspondant], Natalia Lyubova.

5.1.1. Perception Abstraction Engine

Participants: David Filliat [correspondant], Natalia Lyubova.

PAE (Perception Abstraction Engine) is a C++ library developed to provide a uniform interface to existing visual feature detector such as SIFT, SURF, MSER, superpixels, etc... Its main goal is to be able to use these various feature detectors in a "bag of feature" approach for applications such as robot localisation and object recognition. Several approach are also implemented for the visual vocabularies, in particular the fast incremental vocabularies developed in the team.

The library provide common C++ interfaces to feature detectors, visual features and visual vocabularies. A factory approach make it possible to change the feature detectors and visual vocabularies types and parameters through configuration strings, without the need to recompile. Some applications are also included in the library, in particular topological robot localization (room recognition) and visual object recognition. An Urbi interface is also provided for these modules.

5.1.2. Incremental object discovery

Participants: Natalia Lyubova [correspondant], David Filliat.

This software makes it possible to detect, model and recognize objects in a scenario of interaction between a humanoid robot and a human teacher. It is based either on standard images, or on the kinect camera to take advantage of the depth information. The software is written in C++ and relies mainly on PAE and OpenCV.

The software implements several modules: candidate object segmentation based on motion information, keypoint-based object tracking, incremental object model construction integrating multiple features (keypoints + superpixels) and object categorisation based on mutual information with robot motors (making it possible to segment robot parts, objects and humans).

5.2. Learning Algorithms

5.2.1. Neural online learning library

Participant: Alexander GEPPERTH [correspondant].
nnLib is a C/Python-based library for the efficient simulation of neural online learning algorithms. The core user API is implemented in Python as an object-oriented hierarchy, allowing the creation of neural network layers from configuration files in a completely opaque way, as well as the adaptation of multiple parameters at runtime. Available learning algorithms are: PCA (subspace rule and stochastic gradient ascent), sparse coding, self-organizing map, logistic regression and several variants of Hebbian learning (normalized, decaying, ...). nnLib is under development and will be made available to the public under the GPL in 2012.

### 5.2.2. RLPark - Reinforcement Learning Algorithms in JAVA

**Participant:** Thomas Degris [correspondant].

RLPark is a reinforcement learning framework in Java. RLPark includes learning algorithms, state representations, reinforcement learning architectures, standard benchmark problems, communication interfaces for three robots, a framework for running experiments on clusters, and real-time visualization using Zephyr. More precisely, RLPark includes:

- **Online Learning Algorithms:** Sarsa, Expected Sarsa, Q-Learning, Actor-Critic with normal distribution (continuous actions) and Boltzmann distribution (discrete action), average reward actor-critic, TD, TD(λ), GTD(λ), GQ(λ), TDC
- **State Representations:** tile coding (with no hashing, hashing and hashing with murmur2), Linear Threshold Unit, observation history, feature normalization, radial basis functions
- **Interface with Robots:** the Critterbot, iRobot Create, Nao
- **Benchmark Problems:** mountain car, swing-up pendulum, random walk, continuous grid world

An example of RLPark running an online learning experiment on a reinforcement learning benchmark problem is shown in Figure 2.

RLPark was started in spring 2009 in the RLAI group at the university of Alberta (Canada) when Thomas Degris was a postdoc in this group. RLPark is still actively used by RLAI. Collaborators and users include Adam White (patches for bug fixes, testing), Joseph Modayil (implementation of the NAO interface, patches for bug fixes, testing) and Patrick Pilarski (testing) from the University of Alberta. RLPark has also been used by Richard Sutton, a professor and iCORE chair in the department of computing science at the University of Alberta, for a demo in his invited talk *Learning from Data* at the Neural Information Processing Systems (NIPS) 2011. Future developments include the implementation of additional algorithms (the Dyna architecture, back propagation in neural networks, ...) as well as optimizations of vector operations using GPU (with OpenCL) and additional demos. Future dissemination includes a paper in preparation for the JMLR Machine Learning Open Source Software. Documentation and tutorials are included on the http://thomasdegris.github.com/rlpark/ RLPark web site. RLPark is licensed under the open source Eclipse Public License.

### 5.2.3. Autonomous or Guided Explorer (AGE)

**Participant:** Sao Mai NGUYEN [correspondant].

The "Autonomous or Guided Explorer" program is designed for the systematic evaluation and comparison of different exploration mechanisms allowing a simulated or a real robot to learn and build models by self-exploration or social learning. Its conception allows an easy selection of different intrinsically motivated exploration or classical social learning mechanisms. Are provided algorithms such as Random Exploration, SAGG-RIAC, SGIM-D, imitation learning, learning by Observation. The program uses the new objet-oriented programming capability of Matlab, to enhance flexibility and modularity. The main program is built around objects that represent the different modules and the general architecture of such learning algorithms: action space exploration, goal space exploration, interaction with a human, robot control, model computation, but also evaluation and visualisation modules.
The software is designed to easily tune learning parameters and to be easily plugged to other robotic setups. Its object-oriented structure allows safe adaptation to different robotic setups, learning tasks where the structure of the model to learn differs, but also different action or goal spaces. This program is used by Sao Mai Nguyen of the team to compare the performance of different learning algorithms. These results were partly published in [27]. Future work will take advantage of its flexibility and implement new default robotic setups, robot control, action and goal spaces, and most of all, new types of interaction with a human.

5.2.4. NMF Python implementation

**Participant:** Olivier Mangin [correspondant].

This library is meant to implement various algorithms for Non-negative Matrix Factorization in the Python programming language, on top of the Numpy and Scipy scientific libraries.

Some Python NMF libraries already exist, such as the one present in the scikit-learn project. However most of them are quite limited in comparison to recent advances in these techniques (for example extension of NMF algorithms to wider families of penalties such as the beta-divergence family). On the other hand existing MATLAB software has been released by the authors of some of these algorithms but, first, code is not available for every interesting algorithm and none of those various pieces of code implements the whole set of features that one would like to use.

This project is in a very early stage and yet only for internal use in the team. It could, however, be released in the future, for example integrated in the previously mentioned scikit-learn project.

5.3. Software Platforms

5.3.1. JBox2D wrapper

**Participant:** Fabien BENUREAU [correspondant].
ProcBox2D is a wrapping of Processing and JBox2D to satisfy common robotic research needs. In order to quickly prototype research ideas, a simple and efficient simulation framework is of great use. JBox2D is a 2D rigid-body physic engine. Written in Java, it is very fast, typically allowing to compute simulation 60 times faster than real time. Mass simulations can be carried in a timely manner, and improving the process of iterating the conception and implementation of new algorithms. Processing is a graphical framework in Java, and is used to display the simulations of JBox2D. An example of a simulation rendering is visible in Figure 3.

![Simulation](image)

Figure 3. A JBox2D simulation rendered with Processing using ProcBox2D. A robotic arm is interacting with a dynamic object (in pink and yellow); the environment contains obstacles and walls (in dark purple).

While several libraries exist that expose the JBox2D engine to the Processing framework, they suffer from binding Processing irrevocably into the experiment description. As such, simulations without a graphical context, a situation commonly encountered on remote servers and computing clusters, are impossible using these libraries. ProcBox2D was written to fill this gap. It allows the conception of experiments to be done using Processing display capability, while, later on, without modifications of the code, to execute the simulations without any dependency to Processing, on a cluster for instance. The use of Processing allows interactions with the scene via the mouse, which makes ProcBox2D a potential tool in demonstration or imitation learning experiments.

ProcBox2D also provides a sensor and controller interface. Actuated joints can be controlled in torque and velocity, and a PID controller for position control is planned. ProcBox2D implementation began in November 2011 and was presented and made available to the team in December 2011. It is expected that it will increase productivity of researchers that previously had to work out a solution for themselves, often using in early stage of research complex and time-consuming simulation frameworks.

5.3.2. V-REPBridge

**Participant:** Paul FUDAL [correspondant].

V-REPBridge (formerly uV-REPBridge) is a set of software tools to control V-REP through an external client; it consists of a plugin for V-REP and an API to control V-REP.

V-REP - the Virtual Robot Experimentation Platform - is a robot simulator which allows the editing and simulation of robotic systems and sub-systems. Also, it can be integrated and combined using a complete API.
V-REPBridge is a way to interact with a simulation loaded through an Urbi script or a Python application. Based on network communication, V-REPBridge can be used locally (V-REP and the client on the same computer) or remotely. The V-REP simulator’s main use is to perform experiments with virtual robots and environments. But, because V-REPBridge API provides classic functionality like, for example, setting position of a joint or its torque, getting sensor value, etc... an existing application built on top of V-REPBridge can be easily repurposed to use the interface of a real robots.

The development of the plugin for V-REP is made under Windows environment using the V-REP and Windows API. The plugin acts as a server to which a client can connect in order to control the simulation. The client is provided as an API written in C++. This API is available for Windows, Mac and Linux and bindings are available for UrbiScript and Python. The bindings are based on the Urbi API and the Boost Python Library.

Today, V-REPBridge is fully functional and already used in several research experiments, and provide more than 130 V-REP API functions which can be called by the client; here is an non-exhaustive list of V-REP functionalities available in the client :

- joint functionality (position, velocity, torque, etc...),
- object functionality (position, orientation, etc...),
- force sensor functionality,
- inverse kinematic and geometric functionality,
- proximity sensors functionality,
- collision detection functionality,
- minimum distance calculation functionality,
- path planning functionality,
- dynamic functionality,
- ...

V-REPBridge is also provided with an user documentation which includes some howtos (build, use), a complete list of available functions (with synopsis and parameters/returned value description) and some short examples written in Urbi and Python.

Finally, a developer documentation will be available soon to help developers who wants to implement missing V-REP calls both in the plugin and the client, or wants to implements theirs owns functions callable in the client.

The development of V-REPBridge was started at the beginning of year 2011. First release was made in February for testing and debugging foundation of the software. After this short period, time was spent expanding the software and adding new functionalities to bring a response to the needs of the team. First experiments with V-REPBridge was made for IJCAI in july (Mai NGUYEN), ICDL in august (Mai NGUYEN/Matthieu LAPEYRE) and Humanoid in october 2011 (Matthieu LAPEYRE). It was a good feedback for improving the performance and to identify potential improvements.

Work is still in progress for minor bugfixes, support of V-REP minor releases and preparation of the future version of V-REP which will run not only Windows but also on Linux and Mac OS X. A first private beta of V-REP 3 will be available at the end of january.

5.3.3. Rhoban Move Studio

Participants: Olivier Ly [correspondant], Hugo Gimbert, Jérôme Béchu, Paul Fudal.

5.3.3.1. Main software stack

RhobanMoveStudio is a software suite to easily create and control robots, Acroban, FLOWERS Fields/Robot Lamps and Ergorobots/FLOWERS Fields in particular.

This platform has already been presented last year, but it has evolved, in particular for the motor control part. The software architecture has been kept similar but performance has been improved.
Figure 4. The complete software architecture of Rhoban Move Studio
The system runs on an electronic board (based on ARM9 processor) and uses a linux distribution (OpenWrt).

The software is composed of several layers:

- **Kernel module** The role of the module is to implement the electronic communication with devices. It enables to manage Dynamixel\(^1\) motors, generates PWM\(^2\) signals, uses digital readers/writers, I2C bus and more. This year the motor communication have been significantly improved and gained support for accelerometers. This module is designed to run in root mode, to guarantee execution without system interruption, as required by robotic application.

- **Low level** This set of functions is used to communicate with the module through a dedicated shared memory.

- **Move Scheduler** This library provides enables a high level specification of low level motor control loop based on graph of input/output blocks (see Section 5.3.3.2).

- **Rhoban server** This software offers access to the full API of rhoban features through a TCP Socket.

- **Librhoban** This TCP client library provides communication with the Rhoban Server and thus to the whole API. It is a dynamic library, thread safe and secure.

Except for the kernel module which is written in C ANSI, this softwares are written in C++.

### 5.3.3.2. Move Scheduler

Recently (October 2011) a new layer was added to the software. Its role is to enable low level motor control loops through a high level representation.

This representation introduces the concept of blocks. Each block is a computing unit with inputs and outputs. The output of a block can be the input of another one, thus forming a graph of interaction between those unit. Each block is a function (for example addition, multiplication, derivation, integration, spline generation). Special blocks are also provided for sensor inputs and motor outputs.

Graphical interface was developed to easily design such movements. It is called Move Scheduler Modeler, and written in Python (PyQt). This software has import/export capabilities to XML files.

### 5.3.4. UFlow

**Participant:** Jérome Béchu [correspondant].

We developed some new UObjects to enrich the UFlow Toolbox. The UFlow Toolbox is a collection of various software modules for programming and scripting robot sensorimotor loops, aimed at allowing rapid prototyping in the FLOWERS team, and integrated in the URBI framework. URBI, developed by GOSTAI, supports the integration of heterogeneous robotic software modules. It uses a dynamic scripting language, which manages parallel and event processing. Each module, called UObject, is written in C++. We still continue to develop this collection of UObjects for the team.

#### 5.3.4.1. USoundManager

This UObject is used to play sound. It's possible to update the sound while playing. This new version is already based on FMOD.

A new version has just been made. Based on OpenAL, this UObject has the exact same interface as the previous one expect that we include a media manager. With this functionality we load just one time the same sound (We keep it in memory a dictionary of sounds).

#### 5.3.4.2. URhoban

wrap the API of the librhoban (see the previous chapter). This tool is especially develop to control Bioloid motors in high frequency. With that software we can create instance of motors scanned and directly read and write features like position, torque, load, speed.

---

2. Electronic signal : Pulse With Modulation
5.3.4.3. **UXml**

is an UObject based on TinyXml. It is designed to quickly save and restore URBI List in a xml file. It is generally used to store/load parameters like the list of motors in the ErgoRobot platform.

5.3.4.4. **USmartLed**

was created to use the LinkM USB Device to control RGB lights. It is based on the linkm driver (modified to support multiple USB devices). We can control intensity of each light for each primary color.

5.3.4.5. **UGui**

is designed to draw basic 2D primitives. A new version based on SFML was developed this year. It is used in the ErgoRobot project to run a simulation of the setup with a graphical interface.

5.3.4.6. **USqlite**

is an UObject to wrap functionalities of SQLite in URBI. SQLite is a software library that implements a tiny SQL database engine.

5.3.4.7. **UNamingGame**

is UObject used to play the Naming Game. The Naming Game is an algorithm based on communication between agents, who progressively agree meanings of words.

5.3.5. **ErgoRobot/Flowers Field Software**

**Participants:** Jérôme Béchu [correspondant], Pierre-Yves Oudeyer, Pierre Rouanet, Olivier Mangin, Fabien Benureau, Mathhieu Lapeyre.

In the context of its participation to the exhibition “Mathematics: A Beautiful Elsewhere” at Fondation Cartier pour l’Art Contemporain in Paris, starting from 19th October 2011 and to be held until 18th March 2012, the team has elaborated and experimented a robotic experimental set-up called “Ergo-Robots/FLOWERS Fields”. This set-up is not only a way to share our scientific research on curiosity-driven learning, human-robot interaction and language acquisition with the general public, but, as described in the Results and Highlights section, attacks a very important technological challenge impacting the science of developmental robotics: How to design a robot learning experiment that can run continuously and autonomously for several months?
The global scenario for the robots in the installation/experiment is the following. In a big egg that has just opened, a tribe of young robotic creatures evolves and explores its environment, wreathed by a large zero that symbolizes the origin. Beyond their innate capabilities, they are outfitted with mechanisms that allow them to learn new skills and invent their own language. Endowed with artificial curiosity, they explore objects around them, as well as the effect their vocalizations produce on humans. Human, also curious to see what these creatures can do, react with their own gestures, creating a loop of interaction which progressively self-organizes into a new communication system established between man and ergo-robots.

We now outline the main elements of the software architectures underlying this experimental setup.

### 5.3.5.1. System components

The software architecture is organized to control the experiment at several levels, and in particular:

- **Scenes**: The organization of behavioural scenes, managing the behaviours that are allowed to each robot at particular times and in particular contexts;
- **Behaviours**: The individual behaviours of robots, also called stems, which are outlined in the next section;
- **stems**: The low-level actions and perception of robots while executing their behaviours, including motors control on the five physical stems, color and intensity of lights inside the stem head, production of sounds through speakers. Sensors are the kinect used to interact with visitors, and motor feedback capabilities.

In addition to that a video projector is used to display some artistic view of stem agents internal state.

![Diagram](image.png)

**Figure 6. Three important concepts in ErgoRobots**

### 5.3.5.2. Behaviours

A number of innate behaviours were designed and are used by the robots as elementary behaviours of more complex behaviours, including the three following learning behaviours.

**The Naming Game** is a behaviour played by stems two-by-two and based on computational models of how communities of language users can self-organize shared lexicons. In the naming game, stems interact with each other in a stylized interaction. Repeated interactions lead to the development of a common repertoire of words for naming objects. More precisely, objects belong to meaning spaces. Two such spaces have been implemented for the exhibition. The first one is related to object spatial categorization and the second one is related to movement categorization. The object space contains stems, some holes in walls and the interaction zone. The movement space contains representations of small dances that stem can produce and reproduce.

**Object Curiosity** is a behaviour in controlling intrinsically motivated exploration of the physical environment by the stems. A small wood object is present in the reachable physical environment of the stem, attached on the top of a spring so that it is guaranteed that it comes back to its original position. The stem uses a motor primitive to act on the object and motor feedback to detect movements of the object. The robot learns through active exploration what kind of parameters motor primitive will result in touching the object.
Figure 7. A Stem with the head designed by David Lynch and an Object
*Birds Curiosity* is a behaviour that drives robots to explore, through curiosity-driven learning, interaction with humans. One stem, generally the stem in the center, plays a sound, predicts the visitor reaction, look the interaction zone and wait the gesture of the visitor. To produce a sound the visitor have to make a gesture in space. In the next iterations, the robot chooses to produce sounds to human which produce most surprising responses from the human (i.e. the robot is “interested” to explore sound interactions which are not easily predictable by itself).. As describe in the picture, the space is split in four. Each zone corresponding with a sound.

![A virtual visitor interact with a virtual grid](image)

*Figure 8. A virtual visitor interact with a virtual grid*

### 5.3.5.3. Programming tools

The system is based on URBI and used some UObjects from UFlow. The most important part of the system is written in URBI script. Python and freenect³ are used too.

The system at the startup detects motors and lights. It create dynamically a list of Stem. A Stem is one robot with 6 motors as described in hardware part.

To interact with people, we used the freenect library to interface with the kinect, with a binding to python where detection and following of gestures is made.

For the display, we display an abstract rendering of the structure inside each ErgoRobot, using a python parser to read and parse log file from the ErgoRobot system, and the Bloom/Processing software to create and display the rendering. Currently, the system has three displays, one for the naming game, another one for birds curiosity and the last one for objects curiosity.

The sound system used the UObject USoundManager. It plays sounds when required by a behaviour, it also plays word sounds in Naming Game behaviour.

The Light system used Linkm technologies. In the head of each ErgoRobot we put two lights devices. Each light device is a RGB Light. We can control the intensity of each primary color through I2C control. To control lights we used LinkM USB Device. And finally we used an UObject dedicated to communicate with the USB Device.

³Kinect library
5.3.5.4. Maintenance

A dedicate maintenance software is used to switch off, switch on the system. This software is written in Python (and Qt). The status of ErgoRobots is display on the graphical interface. Buttons are present too: Start, Stop, Reset and Take a video.

Recently we added a video system to have a visual feedback of motors usage and also to detect eventual problems. This is a screenshot of the application:

5.4. Visualization Tools

5.4.1. Zephyr - Visualization Platform

Participant: Thomas Degris [correspondant].

Zephyr is a software in Java and Eclipse Rich Client Platform to visualize numeric variables and data structure in real time and at different time scale. Zephyr is practical for developers because it requires only minimal changes in the code to debug: it uses Java reflexivity to automatically detect variables in the code to monitor and data structure with an associated dedicated view. Zephyr can easily be extended with new plugins because it is based on the popular Eclipse Rich Client Platform. Consequently, Zephyr takes advantage of already existing and fully operational Eclipse plugins for many of its functionalities. Finally, Zephyr is distributed with a Java python virtual machine named Jython and a lisp implementation named Clojure. An example of a Zephyr screen is shown in Figure 11.

Zephyr was started in fall 2009 in the RLAI group at the university of Alberta (Canada) when Thomas Degris was a postdoc in this group. Zephyr is still actively used by RLAI. Users include Adam White, Joseph Modayil and Patrick Pilarski from the University of Alberta. Moreover, Zephyr has been registered on the Eclipse marketplace since October 2011 where it has been downloaded a few times by anonymous users. Future dissemination includes the implementation of demos and tutorial videos. Documentation about Zephyr is included on http://thomasdegris.github.com/zephyr/. Zephyr is licensed under the open source Eclipse Public License.

5.4.2. Bloom - particle-based physical engine

Participants: Fabien BENUREAU [correspondant], Olivier Mangin [correspondant].
Figure 10. Maintenance Software for the ErgoRobots.

Figure 11. An example of Zephyr showing the different steps of a video processing pipeline in real-time.
Bloom is a particle-based physical engine that was coded for the Ergorobot exhibition. Written in a matter of days in September 2011, Bloom is based on Processing and coded in Java. It is currently running all of the projected visualisation of the Ergorobot installation. Bloom greatest strength is to provide an intuitive and lightweight tool to display of complex and dynamic information, such as the morphology of a robot vocabulary, as shown in Figure 12. As such, it should permit to examine the state of complex data structure in real-time during experiment, getting insights and allow detection and tracking of issues in algorithms being developed. Bloom is a great complement in research work to the use of charts and graphs. Bloom has since be made available and presented to the team in December 2011.

Figure 12. The vocabulary of 2 interacting robots of the ErgoRobot installation. The blue particles represent meanings, while the orange ones represent words. The strength of association between them is represented by the length of the edges linking them (shorter is stronger). It easy to spot the presence of synonyms and of difference of topology in the vocabulary of the two robots.

5.5. Hardware

5.5.1. The Ergo-Robots Hardware Platform

Participants: Jerome BECHU [correspondent], Fabien BENUREAU, Haylee FOGG, Hugo GIMBERT, Matthieu LAPEYRE, Olivier LY, Olivier MANGIN, Pierre-Yves OUDEYER, Pierre ROUANET.

ErgoRobots is a hardware platform for showcasing a number of curiosity and learning behaviours for the public to interact with. The platform can also have future uses inside the lab for experiments that require more than one robot to complete. Although this system is entirely new this year, a very different previous version existed with the name FLOWERSField. It consists of five ErgoRobots, a control system, an interaction system, a display system, a sound system and a light system. There is an external system which monitors the ErgoRobots which contains a control system, a power system, a surveillance system and a metric capture system. This system went live on October 19 2011 without lights which will be added in late December.

The Ergo-Robot system: The robots themselves are each composed of six motors (see figure). Currently, the heads of the robots have been created in wax by David Lynch and the entire system is displayed at Fondation Cartier inside a large egg shaped orb as shown in the following diagram. The control system module contains both an MMNET1002 control board with an UART-RS485 breakout board which communicates with a ubuntu Linux PC via an ethernet cable. The mment board communicates with the motors, but all other ErgoRobot systems communicate with the PC directly. The sound system is currently externally provided and communicates with the PC. The light system is a series of two or three BlinkM RGB leds placed
inside each ErgoRobot head that are controlled through two LinkM USB devices directly with the computer. A kinect placed in front of the system operates as the means for the public to interact with the platform and communicates directly through USB to the PC. The display system is currently an externally provided projector that projects visualisations of the field’s current state behind the ErgoRobots.

**The external system:** This system allows anyone that is monitoring the system to externally control the ErgoRobots system. The PC with which the software control takes place is a Ubuntu Linux system which communicates with the ErgoRobot control system via an ethernet cable. The ErgoRobot hardware system can be managed by an external power system which includes a 15.5V bench top power supply for the ErgoRobot motors, an external 12V plug in adapter for the mment board, an external 5V plug in adapter for the LED lights which are all controlled via an emergency stop button. The maintenance system can be located out of direct view of the ErgoRobot field as it has a surveillance system: a kinect that can display the current state of the field. More surveillance is conducted through a metric capture system that communicates with the ErgoRobots to obtain various state values of the ErgoRobots through the motor sensors and other data. This surveillance is not entirely in place as of 2011 and will be implemented in early 2012.

![Figure 13. Ergo-Robots](image)

**5.5.2. Flowers Field/Robot Lamps**

**Participants:** Jérôme BÉCHU [correspondant], Pierre-Yves OUDÉYER, Olivier LY, Fabien BENUREAU.

We continued to develop the FLOWERS FIELDS/Robot Lamps experimental set-up, see Figure 14. This set-up explores new forms and new functions of robotics. When we think of robots, we traditionally have in mind either humanoid robots that look like humans and are supposed to do similar things as humans, or industrial robotic arms which should work in factories. On the contrary, the future may come with unforeseen kinds of robots that may enter our everyday homes: for examples, as houses become themselves intelligent
with domotics, we could imagine that furnitures themselves could become robots. Chairs, tables, televisions, or lamps may become robots. In FLOWERS FIELDS/Robot Lamps, we show robotic lamps which move like living entities, with their own moods and their own system of interaction. They can be thought to be in houses partly as aesthetic objects, and partly for their social presence. Indeed, not only their movements and sounds are life-like, but they are sensible to human presence and can become interested in looking and interacting with people through those movements and sounds. In the future, we could imagine additionally that these robot lamps could serve as a friendly interface with the numeric world: for example, some gestures may be used towards the lamps to tell their hifi system to play a given song in your library.

This year, a major update of the platform consisted in shifting the whole servomotor technology to the RX Robotis Series, allowing much more robustness and sophistication of control. The software was adapted to these new motors, requiring indeed a new mode of control together with a new electronic board. This installation was demonstrated in March 2011 at the INNOROBO International Summit on Personal Robotics in Lyon.

5.5.3. Humanoid Robot Torso

**Participant:** Haylee FOGG [correspondant].

The Humanoid Robot Torso is a hardware platform that is intended for use in the lab for either experiments or demonstrations. It consists of a humanoid robot that contains just a torso, arms and head. It is entirely new this year, but it has been updated once during the year. The previous version was inspired by Acroban and consisted of 20 degrees of freedom. The update began in November of 2011 where 3 degrees of freedom was removed from the spine and one degree of freedom was removed from the head.

The Torso has two arms. At the time of writing one arm consists of a three fingered claw that is controlled by a single motor, and the other is just a flat push mechanism. The arm with the claw contains seven degrees of freedom (including ‘grip’) and the other only five. The torso itself has two degrees of freedom. The head is soon to consist of an iPhone for the face and a separate usb camera for the ‘eyes’ with the ability to move in two degrees (pitch and roll) in early 2012.

The hardware is both robotis Dynamixel RX-28 and R-64 motors attached together with standard robotis frames and a substance called polymorph. Polymorph is used to attach a series of springs and elastic to many
of the degrees of freedom to increase smoothing and absorb backlash of the motors. Polymorph was added in November 2011 to replace the previous version's metal that was tooled in the lab.

A method for controlling the motors of the Torso will be under review in 2012.
5. Software

5.1. Cogui

Participants: Alain Gutierrez, Michel Chein, Michel Leclère, Marie-Laure Mugnier, Madalina Croitoru.

Cogui (http://www.lirmm.fr/cogui) is a tool for building and verifying knowledge bases. It is a freeware written in Java (version 1.2, 2005–2010 GPL Licence). Currently, it supports Conceptual Graphs and import/export in RDFS. It relies on CoGITaNT for reasoning tasks.

Here are the major evolutions of the version delivered this year:

- XML Datatypes are now supported.
- The use of URIs as identifiers and the notion of namespaces have been introduced to facilitate interoperability with RDF/XML.
- A pure java solver has been implemented to preserve reasoning capabilities on all platforms.
- A scripting language has been introduced on top of Cogui to satisfy specific applications requirements and facilitate the writing of prototypes. Scripts can be serialized in Cogui projects and give end-users the ability to manipulate objects of the knowledge base and use reasoning features through the Cogui core API.

5.2. Towards Large Knowledge Bases

Participants: Jean-François Baget, Madalina Croitoru, Bruno Paiva Lima Da Silva.

We have began to study different storage solutions for large databases, first as part of a Master’s thesis, and now with the PhD of Bruno Paiva Lima Da Silva [29]. The goal of this work is to evaluate different storage paradigms and systems (e.g., relational databases MySQL and Sqlite; triple stores Sesame and graph databases Neo4J, DEX, HyperGraphDB and OrientDB) with respect to our particular requirements (mainly ontological conjunctive query answering with large knowledge bases), and to integrate them in a unified way in a software tool (answering our genericity requirement and paving the way for hybrid KBs). We believe this work to be a necessary step for our next generation of software tools.
5. Software

5.1. ElevatorRoutePlanner

Participants: Fawzi Nashashibi [correspondant], Paulo Lopes Resende.

This software is dedicated to the building of a decision system that performs task planning and especially route planning for a multiple vehicles based system. Each vehicle sends remotely and asynchronously its position and speed to the system and the "elevator route planner" decides for the destination of each vehicle among a predefined map of fixed stations. Since the stations are on one side of the road, the vehicles are possibly sharing the same stations, leading sometimes to conflictual situations the system has to solve.

- Version: V1

5.2. MELOSYM

Participants: Fawzi Nashashibi [correspondant], Jianping Xie.

MELOSYM is the acronym for “Modélisation de l’Environnement et Localisation en temps réel pour un Système Mobile autonome ou pas, fondé sur des données du capteur laser”. This is a SLAM based algorithm for the environment mapping and vehicle localisation in real time using laser data. The particularity of the algorithm is its hierarchical approach that improves the accuracy of the system and speeds up the computations.

- Version: V1

5.3. ObstaclesDetectionLaser

Participants: Fawzi Nashashibi [correspondant], Paulo Lopes Resende, Laurent Bouraoui.

This is a software for obstacle detection by processing laser range finders data and return the position and distance of the nearest obstacle with regard to the vehicle. The data are from single or multi-layer sensors with different scopes and ranges. The sensors can operate simultaneously or individually, synchronously or not.

- Version: v1

5.4. Path2TrajectoryPlanner

Participants: Fawzi Nashashibi [correspondant], Paulo Lopes Resende.

This software can calculate the exact trajectory of a vehicle from its route decided by another decision-making system. The trajectory is expressed in terms of position and orientation and velocity versus time.

- Version: V1

5.5. SimpleController

Participants: Fawzi Nashashibi [correspondant], Paulo Lopes Resende.

This software enables the development of simple commands or controllers to be applied to drive members (actuators) of a vehicle allowing it to perform a pre-calculated trajectory. The component implements a path following controller. It takes as inputs a trajectory and a vehicle state and it determines the steering and velocity command to be performed by a car type vehicle.

- Version: V1
5.6. CCI

Participants: Fawzi Nashashibi [correspondant], Carlos Holguin.

This software provides a visual and audio interface for the users in a vehicle, in which they can select a destination and see the status of the trip and vehicle systems. It is formed by a component that runs in RTMaps that communicates through a tunnel with a C# application.

- Version: V1
5. Software

5.1. IKONA/MAESTRO Software

Participants: Vera Bakić, Nozha Boujemaa, Jean-Paul Chièze, Raffi Enficiaud, Alexis Joly, Laurent Joyeux, Olfa Mzoughi, Souheil Selmi, Itheri Yahiaoui.

IKONA is a generalist software dedicated to content-based visual information indexing and retrieval. It has been designed and implemented in our team during the last years [21]. Its main functionalities are the extraction, the management and the indexing of many state-of-the-art global and local visual features. It offers a wide range of interactive search and navigation methods including query-by-example, query-by-window, matching, relevance feedback, search results clustering or automatic annotation. It can manage several types of input data including images, videos and 3D models.

Based on a client/server architecture, it is easily deployable in any multimedia search engine or service. The communication between the two components is achieved through a proprietary network protocol. It is a set of commands the server understands and a set of answers it returns to the client. The communication protocol is extensible, i.e. it is easy to add new functionalities without disturbing the overall architecture. can be replaced by any new or existing protocol dealing with multimedia information retrieval.

The main processes are on the server side. They can be separated in two main categories:

- off-line processes: data analysis, features extraction and structuration
- on-line processes: answer the client requests

Several clients can communicate with the server. A good starting point for exploring the possibilities offered by IKONA is our web demo, available at http://www-roc.inria.fr/cgi-bin/imedia/circario.cgi/bio_diversity?select_db=1. This CGI client is connected to a running server with several generalist and specific image databases, including more than 23,000 images. It features query by example searches, switch database functionality and relevance feedback for image category searches. The second client is a desktop application. It offers more functionalities. More screen-shots describing the visual searching capabilities of IKONA are available at http://www-rocq.inria.fr/imedia/chir-demo.html.

IKONA is a pre-industrial prototype, with exploitation as a final objective. Currently, there does not exist a licensed competitor with the same range of functionalities. It exists several commercial softwares or systems exploiting technologies similar to some functionalities of IKONA but usually not the most advanced ones. We can for example cite the SDK developed by LTU company, the service proposed by AdVestigo company, etc. Many prototypes and demonstrators, industrial or academic, share some functionalities of IKONA but here again not the most advanced (e.g. Google Image Similarity Search Beta, IBM Muffin, etc.).

The main originality of IKONA is its genericity (in terms of visual features, metrics, input data, storage format, etc.), its adaptivity (to new visual features, new indexing structures or new search algorithms), its innovative interactive search functionalities (Local and Global Relevance Feedback, Local Search with Query Expansion, Search results clustering, etc.) and its scalability thanks to a generic indexing structure module than can support the integration of any new advances.

Current Users of IKONA include European and National Projects Participants through its integration in prototype multimedia systems, commercial companies through user trials (EXALEAD, INA, BELGA, AFP), General or Specific Public through Web demos (Pl@ntNet leaf identification demo).

IKONA software provides a high degree of visibility to IMEDIA scientific works through demos in commercial, scientific and general public events (notably in most INRIA national showrooms). It is also the mainstay of several Multimedia Systems developed at the European level, in conjunction with many Leader European Companies and Research Centers.
5. Software

5.1. jBricks

Participants: Stéphane Huot, Emmanuel Pietriga [correspondent], Mathieu Nancel, Romain Primet.

jBricks (Figure 1) is a Java toolkit that integrates a high-quality 2D graphics rendering engine based on ZVTM (section 5.2) and a versatile input configuration module (based on ICon [40] and FlowStates 5.4) into a coherent framework, enabling the exploratory prototyping of interaction techniques and rapid development of post-WIMP applications running on cluster-driven interactive visualization platforms such as wall-sized displays. The goal of this framework is to ease the development, testing and debugging of interactive visualization applications. It also offers an environment for the rapid prototyping of novel interaction techniques and their evaluation through controlled experiments.

Figure 1. jBricks applications running on the WILD platform (32 tiles for a total resolution of 20 480 × 6 400 pixels). (a) Zoomed-in visualization of the North-American part of the world-wide air traffic network (1 200 airports, 5 700 connections) overlaid on NASA’s Blue Marble Next Generation images (86 400 × 43 200 pixels) augmented with country borders ESRI shapefiles. (b) Panning and zooming in Spitzer’s Infrared Milky Way (396 032 × 12 000 pixels). (c) Controlled laboratory experiment for the evaluation of mid-air multi-scale navigation techniques.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Java (Linux, Mac OS X, Windows)
- Required library or software: several, managed through Maven
- Programming language: Java

5.2. The Zoomable Visual Transformation Machine

Participants: Caroline Appert, Rodrigo de Almeida, Olivier Chapuis, Arjit Gupta, Julien Husson, Emmanuel Pietriga [correspondent], Mathieu Nancel, Romain Primet.
ZVTM provides application programmers with building blocks for implementing complex multi-scale interface components that cannot be handled by traditional WIMP widgets. Featuring off-the-shelf visualisation and navigation components that are easy to combine, ZVTM provides a simple yet powerful API and handles low-level operations such as multi-threading, clipping, repaint requests and animation management. The toolkit is based on the metaphor of universes that can be observed through smart movable/zoomable cameras. The graphical object model permits management of a large number of complex geometrical shapes. It emphasizes perceptual continuity via an advanced animation module that can animate virtually any on-screen modification. This ranges from camera movements and activation of distortion lenses to modification of the visual variables of graphical objects. Various temporal pacing functions are available to control the execution of these animations. ZVTM is now one of the core components of our jBricks toolkit for wall-sized displays (Section 5.1), and current development activities around the toolkit focus on making applications run transparently on cluster-driven ultra-high-resolution wall-sized displays such as that of the WILD visualization platform. The toolkit is also used to develop advanced visualization components for the ALMA observatory’s operations monitoring and control software [29].

Initially developed by Xerox Research Centre Europe and the World Wide Web Consortium (W3C) team at MIT, ZVTM has been available as open-source software under the GNU Lesser General Public License (LGPL) since early 2002. It is used in both academic and industrial projects such as IsaViz (http://www.w3.org/2001/11/IsaViz/), W3C’s visual browser/editor for RDF, Blast2GO (Figure 2 – left) (http://www.blast2go.org/), or ZGRViewer (http://zvtm.sourceforge.net/zgrviewer.html) for viewing large graphs generated by AT&T GraphViz 4 (Figure 2 – right). The development of the toolkit is now supported by INRIA. More information can be found at http://zvtm.sourceforge.net and [43] and [24].

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- License: LGPL
- Type of human computer interaction: Graphique
- OS/Middleware: Java (Linux, Mac OS X, Windows)
- Required library or software: several, managed through Maven
- Programming language: Java

5.3. The SwingStates Toolkit

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon.

4 http://www.graphviz.org
SwingStates [1] is a library that adds state machines and a graphical canvas to the Java Swing user interface toolkit. It was motivated by the lack of widely disseminated toolkits that support advanced interaction techniques and the observation that HCI research toolkits are little used outside the lab. By extending the popular Java Swing toolkit rather than starting from scratch, the goal is to facilitate the dissemination and adoption of SwingStates by practitioners.

SwingStates uses state machines to specify interaction. It provides programmers with a natural syntax to specify state machines and reduces the potential for an explosion of the number of states by allowing multiple state machines to work together or separately. SwingStates can be used to add new interaction techniques to existing Swing widgets, e.g. to select buttons and checkboxes by crossing rather than clicking. It can also be used with the SwingStates canvas (see below) and to control high-level dialogues.

SwingStates also provides a powerful canvas widget. The canvas can contain any Java2D shape, including geometric shapes, images, text strings and even Swing widgets. Shapes can be manipulated individually or collectively, through tags. An intensive use of polymorphism allows to apply almost any command to a tag: the command is then applied to all objects with this tag. Tags are also used in conjunction with state machines, to specify transitions that occur only on objects with a given tag. For example, pie menus can be implemented by creating a canvas in the overlay layer of any Swing application (Figure 3).

Figure 3. A numeric text field whose value can be set by a joystick-like interaction (left) and a semi-transparent menu to change the background color of Swing widgets (right)

SwingStates tightly integrates state machines, the Java language and the Swing toolkit to provide programmers with a natural and powerful extension to their natural programming environment. SwingStates is available at http://swingstates.sf.net under the GNU Lesser General Public License (LGPL).

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: Java virtual machine
- Programming language: Java

5.4. The FlowStates Toolkit

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon, Stéphane Huot.

FlowStates [33], is a new toolkit to program advanced interaction techniques which require non standard input (e.g., two different mice that act independently, a joystick, a tablet, etc.). It is built on top of two existing toolkits: SwingStates [1] and ICon [40].
With FlowStates the developer can program interaction logic using state machines like SwingStates does but does not restrict the set of possible input channels to Java AWT standard input (a single couple <mouse, keyboard>). The state machines just have to define the virtual input events that are required to trigger their transitions so that FlowStates turns these machines into ICon devices which can be plugged to any physical input channels (Figure 4). An ICon device is a data flow building block that has input and output slots in order to be connected to other devices in the simple graphical environment provided by ICon. State machines can also send out events which appear as output slots in the data flow model.

With FlowStates we showed how two models for programming interaction (state machines and data flow) can be fully integrated to offer a huge power of expression. The explicit decision to not set strict limits between the roles of each model makes this hybrid approach highly flexible, the developer setting himself the limit between the two according to his needs and habits.


- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: ICon, Java virtual machine
- Programming language: Java

5.5. TouchStone

Participants: Caroline Appert [correspondant], Michel Beaudouin-Lafon, Wendy Mackay.

TouchStone [5] is a platform for designing, running and analyzing the results of controlled experiments (Figure 5). While it focuses on experiments comparing interaction techniques, it can be used in a wide variety of contexts.

With the Touchstone design platform, a user specifies the factors and the measures of the experiment, the blocking and counterbalancing of trials, and assess the time it will take to run the experiment. Multiple designs can be explored in parallel to assess the various trade-offs. The output of the design platform is an XML file that can be used as input for the run platform.
The Touchstone run platform provides a framework to implement and run an experiment and to collect experimental data. It uses a flexible plug-in architecture to manage a variety of input devices and interaction techniques. The runs of the experiment are controlled by an XML script that can be produced by the design platform.

The analysis platform currently consists of data analysis tools such as JMP, R or Excel. Log data produced by the run platform can be directly loaded into any of these tools. In a future version, analysis sketches will be derived from the experimental design to assist with the analysis.

Touchstone has been used heavily at INSITU over the past three years for the many experiments that we design and run. It has also been used for teaching for the first time in 2011. Students used it to design various experiments during tutorial classes in Master 2 Interaction (“Introduction to HCI” module).


- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X, Linux, Windows
- Required library or software: Java virtual machine
- Programming language: Java

### 5.6. Metisse

**Participant:** Olivier Chapuis [correspondant].

Metisse [38] is a window system that facilitates the design, implementation and evaluation of innovative window management techniques. The system is based on a compositing approach, making a clear distinction between the rendering and the interactive compositing processes. The Metisse server is a modified X server that supports both input and output redirection. The default compositor is a combination of a slightly modified version of FVWM, a standard window manager, with an interactive viewer application called FvwmCompositor.
FvwmCompositor uses OpenGL to display windows, which offers a rich graphics model well adapted to the exploration of new window management techniques. Texture mapping, for example, makes it possible to transform the window shapes in real-time (Figure 6, left). Alpha blending makes it easy to create translucent objects and shadows. Scaling, rotation and translation can also be used to position windows in 2D or 3D (Figure 6, middle and right). Input redirection makes it still possible to interact with applications no matter the visual transformations applied to the windows. It also makes it possible to adapt, reconfigure or re-combine existing graphical interfaces [45]. This year we used again Metisse to implement novel desktop interaction techniques [3].

Figure 6. Sample window management techniques implemented with Metisse: extended paper metaphor (left), interactive table configuration that allows to duplicate and rotate windows (middle) and zoomable 3D desktop (right).

- Web: http://insitu.lri.fr/metisse/
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [38], [45], [39], [41] and [3].
- License: GPL
- Type of human computer interaction: Graphique
- OS/Middleware: X Window et Mac OS X
- Required library or software: OpenGL via nucleo and some usual C/C++ libraries
- Programming language: * C/C++

5.7. Wmtrace

Participant: Olivier Chapuis [correspondant].

Wmtrace [37] includes two tools that help us study an individual user’s window management activity. The first tool runs in the background of an X Window session and continuously logs information about windows and how they are being manipulated. The second uses a VCR-like interface (Figure 7) to replay the resulting logs and analyze the entire session. This tool provides several ways to filter the logs and extract high-level information, including interactive move events and mouse speed. Both tools allow HCI researchers to perform qualitative and quantitative statistical analyses of window management activity.

- Web: http://insitu.lri.fr/~chapuis/software/wmtrace/.
- ACM: H.5.2 [User Interfaces]: Windowing systems
- Software benefit: see [37], [41], [36].

5 http://interaction.lille.inria.fr/~roussel/projects/nucleo/index.html
5.8. The Substance Middleware

Participants: Michel Beaudouin-Lafon [correspondant], Clemens Klokmose, Tony Gjerlufsen, James Eagan, Clement Pillias.

Substance is a middleware based on a novel programming paradigm called data-oriented programming and was designed to facilitate the development of multi-surface interactive applications [20]. Such applications are distributed by nature as they involve a varying number of display and interaction surfaces that are controlled by different computers. For example, our WILD room includes a 32-monitor display wall driven by 16 computers plus a front-end, a multi-touch table, various mobile devices such as iPodTouch and iPads, and the laptops that the users of the room may bring with them. We want to support seamless interaction techniques across these surfaces, such as the pick-and-drop technique pioneered by Rekimoto [44].

Data-oriented programming consists of attaching functionality to a tree data structure through facets attached to the individual nodes of the tree. Facets can be added and removed dynamically, and notified of changes in the tree. Substance supports two powerful ways to share nodes and facets: mounting, where access to the shared tree is managed through remotely, and replication, where the shared tree is replicated at each site and synchronized.

Substance has been used to create two full-scale applications (Figure 8): a generalized Canvas that can display and manage graphics, PDF files, image files and other content (through an extensible content manager) across surfaces spanning multiple displays and computers; SubstanceGrise, which uses multiple instances of the Anatomist/BrainVISA application to display coordinated 3D imagery of many brains in parallel on the WILD wall and control from a physical model of the brain.
Figure 8. The Canvas (left) and SubstanceGrise (right) applications developed with Substance. (©CNRS-Phototheque - Cyril FRESILLON for SubstanceGrise).

Substance is available at http://substance-env.sourceforge.net/ under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X, Linux
- Required library or software: several, managed by Python install
- Programming language: Python

5.9. Scotty

Participants: Michel Beaudouin-Lafon [correspondant], James Eagan.

The goal of Scotty is to support malleable interfaces, i.e. interfaces that can be modified at run-time in ways not anticipated by the designers [18]. Scotty is a toolkit that allows a programmer to extend an existing Mac OS X application without access to its source code. Scotty provides the following abstractions: hooks to alter the appearance of windows and widgets, event funnels to alter their behavior, glass sheets to overlay graphics and add new interaction methods, dynamic code loading and object proxies to redefine and extend existing objects. Scotty also provides a higher-level interface based on instrumental interaction [34]. Scotty currently runs on Mac OS X for applications written with the Cocoa user interface framework.

Scotty has been used to create a number of extensions (Figure 9). Scribbler is a generic extension that uses glass sheets to allow handwritten annotations of any Cocoa window. Teleportation is another generic extension that can teleport and resize the content of any Cocoa window onto another computer, including an iPhone or iPad. The user can interact with the teleported content as if it was on the original computer. It was used to create a content provider for the Substance Canvas (see above), making it possible to display any application running on a laptop onto the WILD wall display and/or table. When vector-based content is available, e.g., for text, Scotty provides smooth rescaling without the typical pixelation apparent when enlarging bitmap images. Finally Stylesheet is an extension to the Pages word processor that provides a semi-transparent toolglass for specifying the styles of paragraphs.
Scotty is available at [http://insitu.lri.fr/Projects/Scotty](http://insitu.lri.fr/Projects/Scotty) under a GNU GPL 3.0 licence.

- ACM: H.5.2 [User Interfaces]: Graphical user interfaces (GUI)
- OS/Middleware: Mac OS X
- Required library or software: none
- Programming language: Objective-C, Python
5. Software

5.1. Eigen

Participant: Gaël Guennebaud [correspondant].

Keywords:

Web: http://eigen.tuxfamily.org/

Eigen is a fast, versatile, and elegant C++ template library for linear algebra and related algorithms. It provides fixed and dynamic size matrices and vectors, sparse matrices and vectors, matrix decompositions (LU, LLT, LDLT, QR, eigenvalues, etc.), some basic geometry features (transformations, quaternions, axis-angles, Euler angles, hyperplanes, lines, etc.), automatic differentiations, etc. Thanks to expression templates, Eigen provides a very powerful and easy to use API. Explicit vectorization is performed for the SSE (2 and later), Altivec and ARM NEON instruction sets, with graceful fallback to non-vectorized code. Expression templates allow to perform these optimizations globally for whole expressions, and to remove unnecessary temporary objects.

Eigen is already a famous library with about 15000 unique visitors of the website per month, while the mailing list holds about 250 members with a very high traffic (400 message per month). After two years of development since the 2.0 release, we released this year the new major 3.0 version.

- Version: 3.0.4
- Programming language: C++

5.2. Expressive Rendering shaders

Participants: Pascal Barla, Benoit Bossavit.

Shaders developed in the course of our research on expressive rendering have been published under the CeCILL-B license, and distributed on the Animaré project webpage (https://iparla.inria.fr/collaborations/animare/). The goal of such a publication is to let members of the scientific community test and compare with our techniques. This also includes plugins for MeshLab and Nuke.

5.3. Navidget - Easy 3D Camera Positioning from 2D Inputs

Participant: Martin Hachet [correspondant].

Keywords:

Web: https://iparla.inria.fr/software/navidget/

Navidget is a new interaction technique for camera positioning in 3D environments. Unlike the existing POI techniques, Navidget does not attempt to automatically estimate where and how the user wants to move. Instead, it provides good feedback and control for fast and easy interactive camera positioning. Navidget can also be useful for distant inspection when used with a preview window.

This new 3D User interface is totally based on 2D input. As a result, it is appropriate for a wide variety of visualization systems, from small handheld devices to large interactive displays. A user study on TabletPC shows that the usability of Navidget is very good for both expert and novice users. Apart from these tasks, the Navidget approach can be useful for further purposes such as collaborative work and animation.

We have developed a C++/OpenGL library, called LibNavidget, which allows you to integrate Navidget in your own applications. A sample application is included in the package.
5.4. ArcheoTUI

Participants: Patrick Reuter [correspondant], Nicolas Mellado.

Keywords:
ArcheoTUI is a software for the virtual reassembly of fractured archaeological objects via tangible interaction with foot pedal declutching. ArcheoTUI is designed to easily change assembly hypotheses, beyond classical undo/redo, by using a scene graph. The software connects to the database of the broken fragments that are organized in an SQL database. In 2011, we extended the ArcheoTUI software in order to account for a physically-based deformation prototype. Moreover, we integrated multi-touch input with a constraint-based reassembly method.
5. Software

5.1. ViSP: a visual servoing platform

Participants: Fabien Spindler [correspondant], Filip Novotny, Eric Marchand, François Chaumette.

Since 2005, we develop and release under the terms of the GPLv2 licence, ViSP, an open source library that allows fast prototyping of visual tracking and visual servoing tasks. ViSP was designed to be independent with the hardware, to be simple to use, expandable and cross-platform.

ViSP allows to design vision-based tasks for eye-in-hand and eye-to-hand visual servoing that contains the most classical visual features that are used in practice. It involves a large set of elementary positioning tasks with respect to various basic visual features (points, straight lines, circles, spheres, cylinders, frames, image moments...) that can be combined together, and image processing algorithms that allows tracking of visual cues (dots, segments, ellipses,...) or tracking of 3D model-based objects. Simulation capabilities are also available. ViSP and its full functionalities are presented in Fig. 1 and described in [ 7 ].

This year, we continued to improve the software and documentation quality. A new version available at http://www.irisa.fr/lagadic/visp/visp.html was released in October 2011. To ease ViSP installation, we provide also precompiled ViSP SDK including pre-built ViSP library and headers.

This last release code has been downloaded 400 times during the first month of availability. It is used in research labs in France, USA, Japan, Korea, India, China, Lebanon, Italy, Spain, Portugal, Hungary, Canada. For instance, it is used as a support in a graduate course delivered at MIT, at IFMA Clermont-Ferrand and ESIR Rennes engineer schools. ViSP is now also a ROS stack and ViSP 3D model-based tracker has been proposed by the community as a ROS package (see http://www.ros.org/wiki/vision_visp ).
5.2. Development work: Robot vision platforms

Participants: Fabien Spindler [correspondant], Romain Tallonneau.

We exploit two industrial robotic systems built by Afma Robots in the nineties to validate our researches in visual servoing and active vision. The first one is a Gantry robot with six degrees of freedom, the other one is a cylindrical robot with four degrees of freedom (see Fig. 2). These robots are equipped with cameras. The Gantry robot allows also to embed grippers on its end-effector.

This platform is by far the most-used one by Lagadic members (9 papers published by Lagadic in 2011 enclose results validated on it or data acquired on it). Note that this platform is also open to researcher from other labs. For example, the work done in [24] was validated on the Gantry robot.

These equipments require specific hardware, but also software maintenance actions and new developments in order to make them evolve. Training and assistance of the users, presentation of demonstrations also form part of the daily activities.

To improve the panel of demonstrations and to highlight our research activities, we have developed a new demonstration that combines 3D model-based visual tracking and visual servoing techniques provided in ViSP (see Section 5.1) to pick up cubes in order to build a tower. One of the challenges was to automate the initial object localization requested to initialize the tracker. At this end we have developed a generic template pose estimation algorithm based on Surf points of interest matched with the corresponding points provided in a database computed offline during a learning step.

5.3. Development work: Medical robotics platforms

Participants: Fabien Spindler [correspondant], Alexandre Krupa.

This platform is composed by two robots, a six degrees of freedom Hippocrates medical arm designed by the Sinters company (see Fig. 3.a) and an Adept Viper S850 arm (see Fig. 3.b). Ultrasound probes connected either to a SonoSite 180 Plus or an Ultrasonix SonixTouch imaging system can be mounted on a force torque sensor attached to the robot end-effector.
The research and experiments concerning ultrasound visual servoing applied to positioning or tracking tasks conducted with this medical robotics platforms are described in Section 6.3. Note that 4 papers published in 2011 by Lagadic enclose results validated on it.

5.4. Development work: Cycab

**Participants:** Fabien Spindler [correspondant], Andrea Cherubini.

The Cycab is a four wheel drive autonomous electric car dedicated to vision-based autonomous navigation (see Fig. 4). A pan-tilt head (Biclops PTM) equipped with a firewire Marlin camera with about 70 degrees field of view is mounted on the front bumper, as well as a Sick LDMRS laser rangefinder. Concerning the computer units, the Cycab is equipped with two microprocessors dedicated to the low level control of the vehicle actuators and a laptop dedicated to high level visual navigation. They are connected through an internal CAN bus. The camera, the pan-tilt head and the laser rangefinder are connected to the laptop. The research and experiments conducted with the Cycab are described in Section 6.2.3. Note that 4 papers published by Lagadic in 2011 enclose experimental results obtained with the Cycab.
Figure 4. Lagadic Cycab vehicle
5. Software

5.1. Face recognition

Participants: Jakob Verbeek [correspondant], Guillaume Fortier.

In a collaboration with Technosens (a start-up based in Grenoble) we are developing an efficient face recognition library. During 18 months Guillaume Fortier, financed by INRIA’s technology transfer program, streamlines code developed by different team members on various platforms. This encompasses detection of characteristic points on the face (eyes, nose, mouth), computing appearance features on these points, and learning metrics on the face descriptors that are useful for face verification (faces of the same person are close, faces of different people are far away). The code will be ported to run in real-time on the mini-pc system of Technosens that implements advanced user interfaces to TV-top videophone systems.

5.2. Large-scale image search

Participants: Matthijs Douze [correspondant], Mohamed Ayari, Cordelia Schmid.

LEAR’s image search demonstration was extended to 100M images. The image dataset was provided by Exalead. Search at this scale is possible due to the Fisher vector representation and the pqcodes software. The search time on a single core is about 250 ms.

In collaboration with Hervé Jégou, from the INRIA Texmex team, we stabilized and improved the pqcodes software package. The software was extended to implement matrix multiplications in the PQ-compressed domain. A non-exclusive license on pqcodes was sold to Technicolor. Another agreement is under negotiation with Morpho (a company owned by Safran).

LEAR’s implementation of the Fisher descriptor was improved in several ways. A new method to train the GMM was developed and the computation time of second-order derivatives (w.r.t. \( \sigma \)) was significantly reduced. Furthermore, the extraction of dense SIFT descriptors was improved in quality and speed.

5.3. Video descriptors

Participants: Heng Wang, Cordelia Schmid.

We have developed and made on-line available software for video description based on dense trajectories and motion boundary histograms [18]. The trajectories capture the local motion information of the video. A state-of-the-art optical flow algorithm enables a robust and efficient extraction of the dense trajectories. Descriptors are aligned with the trajectories and based on motion boundary histograms (MBH) which are robust to camera motion.
LEO Team

5. Software

5.1. AlignViz

Name: AlignViz
Contact: Fayçal Hamdi (hamdi@lri.fr)
Other contacts: Brigitte Safar (safar@lri.fr) and Chantal Reynaud (chantal.renaud@lri.fr)
Presentation: a visualization tool for alignments between ontologies

5.2. AnnoViP

Name: AnnoViP
Contact: Konstantinos Karanasos (konstantinos.karanasos@inria.fr)
Other contacts: Ioana Manolescu (ioana.manolescu@inria.fr) and Jesús Camacho_Rodriguez (jesus.camacho-rodriguez@inria.fr)
Presentation: a tool for editing and exploiting XML documents with annotations in a distributed P2P setting

5.3. EAP Framework

Name: EAP Framework
Contact: Nadjet Zémirline (nadjet.zemirline@supelec.fr)
Other contacts: Chantal Reynaud (chantal.reynaud@lri.fr)
Presentation: a prototype helping to express adaptation strategies based on the use and a semi-automatic combination of patterns

5.4. EdiFlow

Name: EdiFlow (http://scidam.gforge.inria.fr)
Contact: Wael Khemiri (wael.khemiri@inria.fr)
Other contacts: Ioana Manolescu (ioana.manolescu@inria.fr), Jean-Daniel Fekete (jean-daniel.fekete@inria.fr), Pierre-Luc Hémery (pierre-luc.hemery@inria.fr), Véronique Benzaken (veronique.benzaken@lri.fr)
Presentation: A platform for data-intensive visual analytics

5.5. KD2R

Name: KD2R
Contact: Danai Symeonidou (danai.symeonidou@lri.fr)
Other contacts: Nathalie Pernelle (nathalie.pernelle@lri.fr), Fatima Saïs (fatiha.sais@lri.fr)
Presentation: a tool for OWL2 key discovery on RDF datasets

5.6. Glucose2

Name: Glucose 2.0
Contact: Laurent Simon (simon@lri.fr)
Other contacts: Gilles Audemard (audemard@cril.univ-artois.fr)
Presentation: The new version of Glucose (released in 2009), with auto-adaptive clause database management.

5.7. GlucosER

Name: GlucosER
Contact: Laurent Simon (simon@lri.fr)
Other contacts: George Katsirelos (gkatsi@gmail.com) and Gilles Audemard (audemard@cril.univ-artois.fr)
Presentation: a SAT Solver based on Glucose 1.0 with Extended Resolution.

5.8. LiquidXML

Name: Liquid XML (http://vip2p.saclay.inria.fr/?page=liquidxml)
Contact: Asterios Katsifodimos (asterios.katsifodimos@inria.fr)
Other contacts: Ioana Manolescu (ioana.manolescu@inria.fr)
Presentation: a prototype for automatically recommending XML materialized views in order to improve the performance of a query workload.

5.9. LN2R

Name: LN2R
Contact: Fatiha Saïs (sais@lri.fr)
Other contacts: Nathalie Pernelle (pernelle@lri.fr)
Presentation: a logical and numerical tool for reference reconciliation.

5.10. MESAM

Name: MESAM
Contact: Nadjet Zémirline (nadjet.zemirline@supelec.fr)
Other contacts: Chantal Reynaud (chantal.reynaud@lri.fr)
Presentation: a plug-in for Protege 2000 to merge generic and specific models

5.11. RDFViewS

Name: RDFViewS (http://tripleo.saclay.inria.fr/rdfvs/)
Contact: Konstantinos Karanasos (konstantinos.karanasos@inria.fr)
Other contacts: François Goasdoué (fg@lri.fr), Julien Leblay (julien.leblay@inria.fr), and Ioana Manolescu (ioana.manolescu@inria.fr)
Presentation: a storage tuning wizard for RDF applications

5.12. SomeWhere

Name: SomeWhere
Contact: François Goasdoué (fg@lri.fr)
Other contacts: Philippe Chatalic (chatalic@lri.fr) and Laurent Simon (simon@lri.fr)
Presentation: a peer-to-peer infrastructure for propositional reasoning

### 5.13. SpyWhere

- **Name:** SpyWhere
- **Contact:** François-Elie Calvier (fcalvier@gmail.com)
- **Other contacts:** Chantal Reynaud (chantal.reynaud@lri.fr)
- **Presentation:** a generator of mapping candidates for enriching peer ontologies

### 5.14. TaxoMap

- **Name:** TaxoMap ([http://taxomap.lri.fr](http://taxomap.lri.fr))
- **Contact:** Fayçal Hamdi (hamdi@lri.fr)
- **Other contacts:** Brigitte Safar (safar@lri.fr) and Chantal Reynaud (chantal.reynaud@lri.fr)
- **Presentation:** a prototype to automate semantic mappings between taxonomies

### 5.15. TaxoMap Framework

- **Name:** TaxoMap Framework
- **Contact:** Fayçal Hamdi (hamdi@lri.fr)
- **Other contacts:** Brigitte Safar (safar@lri.fr) and Chantal Reynaud (chantal.reynaud@lri.fr)
- **Presentation:** an environment to specify treatments to refine mappings and to enrich ontologies

### 5.16. ViP2P

- **Name:** ViP2P (views in peer-to-peer, [http://vip2p.saclay.inria.fr](http://vip2p.saclay.inria.fr))
- **Contact:** Ioana Manolescu (ioana.manolescu@inria.fr)
- **Other contacts:** Jesús Camacho_Rodriguez (jesus.camacho-rodriguez@inria.fr), Asterios Katsifodimos (asterios.katsifodimos@inria.fr), Konstantinos Karanasos (konstantinos.karanasos@inria.fr)
- **Presentation:** a P2P platform for disseminating and querying XML and RDF data in large-scale distributed networks.

### 5.17. XUpOp

- **Name:** XUpOp (XML Update Optimization)
- **Contact:** Dario Colazzo (colazzo@lri.fr)
- **Other contacts:** Nicole Bidoit (bidoit@lri.fr), Marina Sahakian (Marina.Sahakyan@lri.fr), and Mohamed Amine Baazizi (baazizi@lri.fr)
- **Presentation:** a general purpose type based optimizer for XML updates

### 5.18. XUpIn

- **Name:** XUpIn (XML Update Independence)
- **Contact:** Federico Ulliana (Federico.Ulliana@lri.fr)
- **Other contacts:** Dario Colazzo (colazzo@lri.fr), Nicole Bidoit (bidoit@lri.fr)
- **Presentation:** an XML query-update independence tester
5. Software

5.1. Software

Our software efforts are integrated in a library called RAlib which contains our research development on image processing, registration (2D and 3D) and visualization. This library is licensed by the APP (French agency for software protection).

The visualization module is called QGLSG: it enables the visualization of images, 2D and 3D objects under a consistent perspective projection. It is based on Qt\(^1\) and OpenScenegraph\(^2\) libraries. The QGLSG library integrates innovative features such as online camera distortion correction, and invisible objects that can be incorporated in a scene so that virtual objects can cast shadows on real objects, and occlusion between virtual and real objects are easier to handle. The library was also ported to Mac OS and Windows and a full doxygen documentation was written.

\(^1\) http://www.trolltech.com  
\(^2\) http://www.openscenegraph.org/projects/osg
4. Software

4.1. FF

Participant: Jörg Hoffmann [correspondant].

FF is an automatic planning system, taking as input a high-level description of the planning task in the PDDL language (planning domain definition language), and returning a plan for the task. FF was continuously developed by Jörg Hoffmann over a time span of several years (ca. 1999 – 2006), before joining INRIA. FF has convincingly won the international planning competition in the year 2000, and has been one of the most widely used and cited planning systems (around 1000 citations up to now) ever since then. It still is competitive with the state of the art today. There are several different versions, for deterministic planning with Boolean state variables, for deterministic planning with numeric state variables, for non-deterministic planning with no probabilities (all outcomes are assumed to be equally likely), and finally a version tackling a particular variant of probabilistic planning.

4.2. TorchLight

Participant: Jörg Hoffmann [correspondant].

TorchLight is a system for automatic domain analysis in planning. It automatically infers properties of the search space surface under a particular heuristic function, called $h^+$, that underlies most current state of the art planning systems (including FF). TorchLight examines certain structural properties of the PDDL input, and exploits a number of connections between this structure and the search space surface under $h^+$. For example, one of its outputs provides an estimate of the fraction of states that lie on local minima.

4.3. AA4MM

Participants: Vincent Chevrier [correspondant], Julien Siebert.

This work is undertaken in a joint PhD Thesis between MAIA and Madynes Team. Laurent Ciarletta (Madynes team, LORIA) is co-advisor of this PhD and correspondent for this software.

AA4MM (Agents and Artefacts for Multi-modeling and Multi-simulation) is a framework for coupling existing and heterogeneous models and simulators in order to model and simulate complex systems. This is the first implementation of the AA4MM meta-model proposed in Julien Siebert’s PhD. It is written in Java and relies upon Java Messaging Services (JMS) for its distributed version.

4.4. MASDYNE

Participants: Vincent Chevrier [correspondant], Julien Siebert.

This work is undertaken in a joint PhD Thesis between MAIA and Madynes Team. Laurent Ciarletta (Madynes team, LORIA) is co-advisor of this PhD and correspondent for this software.

Other contributors to this software are: Tom Leclerc, François Klein, Christophe Torin, Marcel Lamenu, Guillaume Favre and Amir Toly.

MASDYNE (Multi-Agent Simulator of DYnamic Networks usErs) is a multi-agent simulator for modeling and simulating users behaviors in mobile ad hoc network. This software is part of joint work with MADYNES team, on modeling and simulation of ubiquitous networks.
METISS Project-Team

5. Software

5.1. Audio signal processing, segmentation and classification toolkits

Participant: Guillaume Gravier.

Guillaume Gravier is now with the TEXMEX group but this software is being used by several members of the METISS group.

The SPro toolkit provides standard front-end analysis algorithms for speech signal processing. It is systematically used in the METISS group for activities in speech and speaker recognition as well as in audio indexing. The toolkit is developed for Unix environments and is distributed as a free software with a GPL license. It is used by several other French laboratories working in the field of speech processing.

In the framework of our activities on audio indexing and speaker recognition, AudioSeg, a toolkit for the segmentation of audio streams has been developed and is distributed for Unix platforms under the GPL agreement. This toolkit provides generic tools for the segmentation and indexing of audio streams, such as audio activity detection, abrupt change detection, segment clustering, Gaussian mixture modeling and joint segmentation and detection using hidden Markov models. The toolkit relies on the SPro software for feature extraction.

Contact: guillaume.gravier@irisa.fr

5.2. Irene: a speech recognition and transcription platform

Participant: Guillaume Gravier.

Guillaume Gravier is now with the TEXMEX group but this software is being used by several members of the METISS group.

In collaboration with the computer science dept. at ENST, METISS has actively participated in the past years in the development of the freely available Sirocco large vocabulary speech recognition software [113]. The Sirocco project started as an INRIA Concerted Research Action now works on the basis of voluntary contributions.

The Sirocco speech recognition software was then used as the heart of the transcription modules within a spoken document analysis platform called IRENE. In particular, it has been extensively used for research on ASR and NLP as well as for work on phonetic landmarks in statistical speech recognition.

In 2009, the integration of IRENE in the multimedia indexing platform of IRISA was completed, incorporating improvements benchmarked during the ESTER 2 evaluation campaign in december 2008. Additional improvements were also carried out such as bandwidth segmentation and improved segment clustering for unsupervised acoustic model adaptation. The integration of IRENE in the multimedia indexing platform was mainly validated on large datasets extracted from TV streams.

Contact: guillaume.gravier@irisa.fr
http://gforge.inria.fr/projects/sirocco

5.3. MPTK: the Matching Pursuit Toolkit

Participants: Rémi Gribonval, Ronan Le Boulch.
The Matching Pursuit ToolKit (MPTK) is a fast and flexible implementation of the Matching Pursuit algorithm for sparse decomposition of monophonic as well as multichannel (audio) signals. MPTK is written in C++ and runs on Windows, MacOS and Unix platforms. It is distributed under a free software license model (GNU General Public License) and comprises a library, some standalone command line utilities and scripts to plot the results under Matlab.

MPTK has been entirely developed within the METISS group mainly to overcome limitations of existing Matching Pursuit implementations in terms of ease of maintainability, memory footprint or computation speed. One of the aims is to be able to process in reasonable time large audio files to explore the new possibilities which Matching Pursuit can offer in speech signal processing. With the new implementation, it is now possible indeed to process a one hour audio signal in as little as twenty minutes.

Thanks to an INRIA software development operation (Opération de Développement Logiciel, ODL) started in September 2006, METISS efforts have been targeted at easing the distribution of MPTK by improving its portability to different platforms and simplifying its developers’ API. Besides pure software engineering improvements, this implied setting up a new website with an FAQ, developing new interfaces between MPTK and Matlab and Python, writing a portable Graphical User Interface to complement command line utilities, strengthening the robustness of the input/output using XML where possible, and most importantly setting up a whole new plugin API to decouple the core of the library from possible third party contributions.

Collaboration : Laboratoire d’Acoustique Musicale (University of Paris VII, Jussieu).

Contact : remi.gribonval@irisa.fr


5.4. FASST

Participants: Emmanuel Vincent [correspondant], Alexey Ozerov, Frédéric Bimbot.

FASST is a Flexible Audio Source Separation Toolbox in Matlab, designed to speed up the conception and automate the implementation of new model-based audio source separation algorithms.
5. Software

5.1. HPTS++: Hierarchical Parallel Transition System ++

Participants: Stéphane Donikian [contact], Fabrice Lamarche [contact].

HPTS++ is a platform independent toolkit to describe and handle the execution of multi-agent systems. It provides a specific object oriented language encapsulating C++ code for interfacing facilities and a runtime kernel providing automatic synchronization and adaptation facilities.

The language provides functionalities to describe state machines (states and transitions) and to inform them with user specific C++ code to call at a given point during execution. This language is object oriented and supports concepts such as polymorphism and inheritance (state machines and user defined C++ classes). The compilation phase translates a state machine in a C++ class that can be compiled separately and linked through static or dynamic libraries. The runtime kernel includes a scheduler that handles parallel state machines execution and that provides synchronization facilities such as mutual exclusion on resources, dead lock avoidance, notions of priorities and execution adaptation in accordance with resources availability.

HPTS++ also provides a task model. Thanks to this model, the user can describe primitive behaviors through atomic tasks and combine them with operators (sequence, parallelism, loops, alternatives...). These operators are fully dynamic. Hence they can be used at runtime to rapidly create complex behaviors.

5.2. MKM: Manageable Kinematic Motions

Participants: Richard Kulpa [contact], Franck Multon.

We have developed a framework for animating human-like figures in real-time, based on captured motions. This work was carried-out in collaboration with the M2S Laboratory (Mouvement, Sport, Santé) of the University Rennes 2.

In this software, we propose a morphology-independent representation of the motion that is based on a simplified skeleton which normalizes the global postural informations. This formalism is not linked to morphology and allows very fast motion retargetting and adaptation to geometric constraints that can change in real-time. This approach dramatically reduces the post production time and allows the animators to handle a general motion library instead of one library per avatar.

The framework provides an animation library which uses the motions either obtained from our off-line tool (that transforms standard formats into our morphology-independent representation) or parameterized models in order to create complete animation in real-time. Several models are proposed such as grasping, orientation of the head toward a target. We have also included a new locomotion model that allows to control the character directly using a motion database.

In order to create realistic and smooth animations, MKM uses motion synchronization, blending and adaptation to skeletons and to external constraints. All those processes are performed in real-time in an environment that can change at any time, unpredictably.

All these features have been used to anticipate and control the placement of footprints depending on high level parameters. This link between control and behavior levels will be used for reactive navigation in order to have realistic motion adaptations as well as to deal with constrained environments.

5.3. TopoPlan: Topological Planner and Behaviour Library

Participant: Fabrice Lamarche [contact].
TopoPlan (Topological Planner) is a toolkit dedicated to the analysis of a 3D environment geometry in order to generate suitable data structures for path finding and navigation. This toolkit provides a two step process: an off-line computation of spatial representation and a library providing on-line processes dedicated to path planning, environmental requests...

TopoPlan is based on an exact 3D spatial subdivision that accurately identifies floor and ceiling constraints for each point of the environment. Thanks to this spatial subdivision and some humanoid characteristics, an environment topology is computed. This topology accurately identifies navigable zones by connecting 3D cells of the spatial subdivision. Based on this topology several maps representing the environment are extracted. Those maps identify obstacle and step borders as well as bottlenecks. TopoPlan also provides a runtime library enabling the on-line exploitation of the spatial representation. This library provides several algorithms including roadmap-based path-planning, trajectory optimization, footprint generation, reactive navigation and spatial requests through customizable spatial selectors.

TopoPlan behavior is a library built on top of TopoPlan and MKM providing several behaviors described thanks to the HPTS++ task model. Its goal is to provide a high level interface handling navigation and posture adaptation within TopoPlan environments. Provided behaviors include:

- A behavior handling fully planned navigation toward an arbitrary destination. This behavior precisely handles footprint generation within constrained environments such as stairs for instance.
- A behavior controlling an MKM humanoid to follow a trajectory specified by the user.
- A behavior controlling MKM to follow a list of footprints given by the user.
- A behavior adapting the humanoid posture to avoid collision with ceiling. This behavior runs in parallel of all other behaviors and adapts humanoid motion when needed without any user intervention.
- A behavior handling reactive navigation of virtual humans. This behavior plan a path to a given target and follows the path while avoiding collisions with other navigating entities.

Those behaviors have been built using the HPTS++ task model. Thus, they can be easily combined together or with other described behaviors through task operators.
MINT Team

5. Software

5.1. LibGINA

**Participant:** Laurent Grisoni [correspondant].

This library has been developed within the context of the ADT GINA, for one of the installations that have been made in collaboration with Le Fresnoy national studio (Damassama, Léonore Mercier). This library is currently being posted as APP, and has been used by Idées-3com small company, in the context of our joint I-lab program. This library allows for use of gesture for command, and is able to handle strong variability into recognized patterns.

Current version: version 1.0

**Software characterization:** A-2 SO-3 SM-2-up EM-3 SDL-3 OC-DA4-CD4-MS2-TPM4

5.2. 3D interaction using mobile phone

**Participants:** Samuel Degrande [correspondant], Laurent Grisoni.

This work has been achieved in the context of the Idées-3com I-lab. In this context a module, that allows to use any android based smartphone to control an Explorer module for navigation and interaction with VRML-based content. This module was used as a basis by Idées-3com in their commercial product this year.

Current version: version 1.0

**Software characterization:** A-2 SO-3 SM-2-up EM-2-up SDL-3 OC-DA4-CD4-MS2-TPM4

5.3. tIO (tactile input & output)

**Participants:** Paolo Olivo, Nicolas Roussel [correspondant].

tIO is a library designed to facilitate the implementation of doubly tactile interaction techniques (tactile input coupled with tactile feedback) based on the STIMTAC technology. Supporting all current STIMTAC prototypes, it makes it easy to move the system pointer of the host computer according to motions detected on them and adapt their vibration amplitude based on the color of the pointed pixel or the nature of the pointed object. The library includes a set of Qt demo applications that illustrate these two different approaches and makes it easy to “augment” existing Qt applications with tactile feedback. It also makes it possible to supplement or substitute tactile feedback with basic auditory feedback synthesized using portaudio (friction level is linearly mapped to the frequency of a sine wave). This not only facilitates the development and documentation of tactile-enhanced applications but also makes it easier to demonstrate them to a large audience.

Current version: 0.1 - June 2011 (IDDN.FR.001.270005.000.S.P.2011.000.10000)

**Software characterization:** A2, SO3-up, SM-2, EM2, SDL1.

5.4. libpointing

**Participants:** Géry Casiez [correspondant], Damien Marchal, Nicolas Roussel.
Libpointing is a software toolkit that provides direct access to HID pointing devices and supports the design and evaluation of pointing transfer functions [16]. The toolkit provides resolution and frequency information for the available pointing and display devices and makes it easy to choose between them at run-time through the use of URIs. It allows to bypass the system’s transfer functions to receive raw asynchronous events from one or more pointing devices. It replicates as faithfully as possible the transfer functions used by Microsoft Windows, Apple OS X and Xorg (the X.Org Foundation server). Running on these three platforms, it makes it possible to compare the replicated functions to the genuine ones as well as custom ones. The toolkit is written in C++ with Python and Java bindings available. It is scheduled to be publicly released in 2012, the licence remaining to be decided.

Web site: http://libpointing.org/

Software characterization: A3, SO3, SM-2, EM2, SDL4
5. Software

5.1. Platforms

5.1.1. The Grimage platform

The Grimage platform is an experimental multi-camera platform dedicated to spatio-temporal modeling including immersive and interactive applications. It hosts a multiple-camera system connected to a PC cluster, as well as visualization facilities including head mounted displays. This platform is shared by several research groups, most prominently MOAIS, MORPHEO and PERCEPTION. In particular, Grimage allows challenging real-time immersive applications based on computer vision and interactions between real and virtual objects, Figure 1.

5.1.2. Virtualization Gate

Vgate is an immersive environment that allows full-body immersion and interaction with virtual worlds. It is a joint initiative of computer scientists from computer vision, parallel computing and computer graphics from several research groups at INRIA Grenoble Rhône-Alpes, and in collaboration with the company 4D View Solutions. The MORPHEO team is leading this project.

![Grimage acquisition](image1)
![Vgate immersive environment](image2)

Figure 1. Platforms: on the left the Grimage acquisition; on the right the vgate immersive environment.

5.1.3. Multicamera platform for video analysis of mice behavior

This project is a follow-up of the experimental set-up developed for a CNES project with Mathieu Beraneck from the CESeM laboratory (centre for the study of sensorimotor control, CNRS UMR 8194) at the Paris-Descartes University. The goal of this project was to analyze the 3D body postures of mice with various vestibular deficiencies in low gravity condition (3D posturography) during a parabolic flight campaign. The set-up has been now adapted for new experiments on motor-control disorders for other mice models. This experimental platform is currently under development for a broader deployment for high throughput phenotyping with the technology transfer project ETHOMICE. This project involves a closed relationship with the CESeM laboratory and the European Mouse Clinical Institute in Strasbourg (Institut Clinique de la Souris, ICS).
5.2. Software packages

5.2.1. LucyViewer

Lucy Viewer [http://4drepository.inrialpes.fr/lucy_viewer/](http://4drepository.inrialpes.fr/lucy_viewer/) is an interactive viewing software for 4D models, i.e., dynamic three-dimensional scenes that evolve over time. Each 4D model is a sequence of meshes with associated texture information, in terms of images captured from multiple cameras at each frame. Such data is available from various websites around the world including the 4D repository website hosted by INRIA Grenoble [http://4drepository.inrialpes.fr/](http://4drepository.inrialpes.fr/). The software was developed in the context of the European project iGlance, it is available as open-source software under the GNU LGP Licence.

5.3. Databases

5.3.1. 4D repository [http://4drepository.inrialpes.fr/](http://4drepository.inrialpes.fr/)

This website hosts dynamic mesh sequences reconstructed from images captured using a multi-camera set up. Such mesh-sequences offer a new promising vision of virtual reality, by capturing real actors and their interactions. The texture information is trivially mapped to the reconstructed geometry, by back-projecting from the images. These sequences can be seen from arbitrary viewing angles as the user navigates in 4D (3D geometry + time). Different sequences of human / non-human interaction can be browsed and downloaded from the data section. A software to visualize and navigate these sequences is also available for download.
5. Software

5.1. FXP

**Participants:** Joachim Niehren [correspondant], Denis Debarbieux, Tom Sebastian.

Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4

The FXP language is a temporal logic for a fragment of Forward XPath that is suitable for querying XML streams. The FXP library of the Mostrare project of INRIA Lille provides a compiler of the FXP library to nested word automata, efficient query answering algorithm for nested word automata on XML streams, and thus for FXP queries.

FXP is developed in the INRIA transfer project QuiXProc in cooperation with Innovimax. Both a professional and a free version are available. The owner is INRIA.

See also the web page [http://fxp.lille.inria.fr/](http://fxp.lille.inria.fr/).

- Version: 0-9-2011-03-25

5.2. QuixPath

**Participants:** Joachim Niehren [correspondant], Denis Debarbieux, Tom Sebastian.

Software Self-Assessment: A-3, SO-4, SM-3, EM-3, SDL-4

The QuiXPath language is a large fragment of Forward XPath with full support for the XML data model. The QuiXPath library provides a compiler from QuiXPath to FXP. Thereby, the efficient query answering algorithms for FXP are lifted to a fragment of Forward XPath. QuiXPath is developed in the INRIA transfer project QuiXProc in cooperation with Innovimax. Both, a free open source and a professional version are available. The ownership of QuiXPath is shared between INRIA and Innovimax. The main application of QuiXPath is its usage in QuiXProc, an professional implementation of the W3C pipeline language XProc owne by Innovimax.

See also the web page [http://fxp.lille.inria.fr/](http://fxp.lille.inria.fr/).

- Version: QuixPath v1.0.0

5.3. VOLATA

**Participant:** Fabien Torre [correspondant].


VOLATA provides several machine learning algorithms for attribute-value inference, grammatical inference and inductive logic programming.


- ACM: I.2.6
5. Software

5.1. Generic Symbolic KDD Systems

5.1.1. The Coron Platform

Participants: Mehdi Kaytoue [contact person], Amedeo Napoli, Yannick Toussaint.

The Coron platform [118], [95] is a KDD toolkit organized around three main components: (1) Coron-base, (2) AssRuleX, and (3) pre- and post-processing modules. The software was registered at the "Agence pour la Protection des Programmes" (APP) and is freely available. The Coron-base component includes a complete collection of data mining algorithms for extracting itemsets such as frequent itemsets, frequent closed itemsets, frequent generators. In this collection we can find APriori, Close, Pascal, Eclat, Charm, and, as well, original algorithms such as ZART, Snow, Touch, and Talky-G. The Coron-base component contains also algorithms for extracting rare itemsets and rare association rules, e.g. APriori-rare, MRG-EXP, ARIMA, and BTB. AssRuleX generates different sets of association rules (from itemsets), such as minimal non-redundant association rules, generic basis, and informative basis. In addition, the Coron system supports the whole life-cycle of a data mining task and proposes modules for cleaning the input dataset, and for reducing its size if necessary. The Coron toolkit is developed in Java, is operational, and was already used in several research projects.

5.1.2. Orion: Skycube Computation Software

Participant: Chedy Raïssi [contact person].

This program implements the algorithms described in a research paper published last year at VLDB 2010 [113]. The software provides a list of four algorithms discussed in the paper in order to compute skycubes. This is the most efficient—in term of space usage and runtime—implementation for skycube computation (see https://github.com/leander256/Orion).

5.2. Stochastic systems for knowledge discovery and simulation

5.2.1. The CarottAge system

Participants: Florence Le Ber, Jean-François Mari [contact person].

CarottAge is a data mining system, freely available (GPL license) and based on Hidden Markov Models of second order. It provides a synthetic representation of temporal and spatial data. CarottAge is currently used by INRA researchers interested in mining the changes in territories related to the loss of biodiversity (projects ANR BiodivAgrim and ACI Ecoger) and/or water contamination.

In these practical applications, the system aims at building a partition—called the hidden partition— in which the inherent noise of the data is withdrawn as much as possible. The CarottAge system takes into account: (i) the various shapes of the territories that are not represented by square matrices of pixels, (ii) the use of pixels of different size with composite attributes representing the agricultural pieces and their attributes, (iii) the irregular neighborhood relation between those pixels, (iv) the use of shape files to facilitate the interaction with GIS (geographical information system).

CarottAge has been used for mining hydromorphological data. Actually a comparison was performed with three other algorithms classically used for the delineation of river continuums and CarottAge proved to give very interesting results for that purpose [73].

1 http://coron.loria.fr
2 http://www.loria.fr/~jfmari/App/
5.2.2. The ARPEnTAge system

Participants: Florence Le Ber, Jean-François Mari [contact person].

ARPEnTAge \(^3\) (for Analyse de Régularités dans les Paysages: Environnement, Territoires, Agronomie) is a software based on stochastic models (HMM2 and Markov Field) for analyzing spatiotemporal data-bases \([73]\). ARPEnTAge is built on top of the CarottAge system to fully take into account the spatial dimension of input sequences. It takes as input an array of discrete data in which the columns contain the annual land-uses and the rows are regularly spaced locations of the studied landscape. Displaying tools and the generation of shape files have also been defined.

We model the spatial structure of the landscape by a Markov Random Field (MRF) whose sites are random Land Uses (LUS) located in the parcels. The dynamics of these LUS are modelled by a temporal HMM2. This leads to the definition of a MRF where the underlying mean field is approximated by a HMM2 that processes a Hilbert-Peano fractal curve spanning the image. This MRF is used to segment the landscape into patches, each of them being characterized by a temporal HMM2. The patch labels, together with the geographic coordinates, determine a clustered image of the landscape that can be coded within an ESRI shapefile.

ARPEnTAge is freely available (GPL license pending) and is currently used by INRA researchers interested in mining the changes in territories related to the loss of biodiversity (projects ANR BiodivAgrim and ACI Ecoger) and/or water contamination.

5.2.3. GenExp-LandSiTes: KDD and simulation

Participants: Sébastien Da Silva, Florence Le Ber [contact person], Jean-François Mari.

In the framework of the project “Impact des OGM” initiated by the French ministry of research, we have developed a software called GenExp-LandSiTes for simulating bidimensional random landscapes, and then studying the dissemination of vegetable transgenes. The GenExp-LandSiTes system is linked to the CarottAge system, and is based on computational geometry and spatial statistics. The simulated landscapes are given as input for programs such as “Mapod-Maïs” or “GeneSys-Colza” for studying the transgene diffusion. Other landscape models based on tessellation methods are under studies. The last version of GenExp allows an interaction with R and deals with several geographical data formats.

This work is now part of an INRA-INRIA project about landscape modeling, PAYOTE (2009–2011), that gathers eleven research teams of agronomists, ecologists, statisticians, and computer scientists. The PAYOTE project is now focusing on the comparison of various methods for analyzing and building temporal and spatial landscape structures. Sébastien da Silva is preparing his PhD thesis within this framework and is conducted both by Claire Lavigne (DR in ecology, INRA Avignon) and Florence Le Ber \([62]\). Florence Le Ber is also involved within a new INRA project on virtual landscape modelling.

5.3. KDD in Systems Biology

5.3.1. IntelliGO online

The IntelliGO measure computes semantic similarity between terms from a structured vocabulary (Gene Ontology: GO) and uses these values for computing functional similarity between genes annotated by sets of GO terms \([82]\). The IntelliGO measure is made available online (http://plateforme-mbi.loria.fr/intelligo/) to be used by members of the community for exploitation and evaluation purposes. It is possible to compute the functional similarity between two genes, the intra-set similarity value in a given set of genes, and the inter-set similarity value for two given sets of genes.

5.3.2. WAFOBI : KNIME nodes for relational mining of biological data

---

\(^3\) http://www.loria.fr/~jfmari/App/
KNIME (for “Konstanz Information Miner”) is an open-source visual programming environment for data integration, processing, and analysis. KNIME has been developed using rigorous software engineering practices and is used by professionals in both industry and academia. The KNIME environment includes a rich library of data manipulation tools (import, export) and several mining algorithms which operate on a single data matrix (decision trees, clustering, frequent itemsets, association rules...). The KNIME platform aims at facilitating the data mining experiment settings as many tests are required for tuning the mining algorithms. The evaluation of the mining results is also an important issue and its configuration is made easier.

A position of engineer (“Ingénieur Jeune Diplomé INRIA”) was granted to the Orpailleur team to develop some extra KNIME nodes for relational data mining using the ALEPH program (http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl). The developed KNIME nodes include a data preparation node for defining a set of first-order predicates from a set of relation schemas and then a set of facts from the corresponding data tables (learning set). A specific node allows to configure and run the ALEPH program to build a set of rules. Subsequent nodes allow to test the first-order rules on a test set and to perform configurable cross validations. An INRIA APP procedure is currently pending.

5.3.3. MODel-driven Data Integration for Mining (MODIM)

Participants: Marie-Dominique Devignes [contact person], Birama Ndiayé, Malika Smaïl-Tabbone.

The MODIM software (MOdel-driven Data Integration for Mining) is a user-friendly data integration tool which can be summarized along three functions: (i) building a data model taking into account mining requirements and existing resources; (ii) specifying a workflow for collecting data, leading to the specification of wrappers for populating a target database; (iii) defining views on the data model for identified mining scenarios. A steady-version of the software has been deposited through INRIA APP procedure in December, 2010.

Although MODIM is domain independent, it was used so far for biological data integration in various internal research studies. A poster was presented at the last JOBIM conference (Paris, June 2011). Recently, MODIM was used by colleagues from the LIFL for organizing data about non ribosomal peptide syntheses. Feedback from users led to extensions of the software. The sources can be downloaded at https://gforge.inria.fr/projects/modim/.

5.4. Knowledge-Based Systems and Semantic Web Systems

5.4.1. The Kasimir System for Decision Knowledge Management

Participants: Nicolas Jay, Jean Lieber [contact person], Amedeo Napoli, Thomas Meilender.

The objective of the Kasimir system is decision support and knowledge management for the treatment of cancer. A number of modules have been developed within the Kasimir system for editing of treatment protocols, visualization, and maintenance. Kasimir is developed within a semantic portal, based on OWL. KatexOWL (Kasimir Toolkit for Exploiting OWL Ontologies, http://katexowl.loria.fr) has been developed in a generic way and is applied to Kasimir. In particular, the user interface EdHibou of KatexOWL is used for querying the protocols represented within the Kasimir system.

The software CabamakA (case base mining for adaptation knowledge acquisition) is a module of the Kasimir system. This system performs case base mining for adaptation knowledge acquisition and provides information units to be used for building adaptation rules [123]. Actually, the mining process in CabamakA is implemented thanks to a frequent close itemset extraction module of the Coron platform (see § 5.1.1). A semantic wiki for the collaborative edition of decision protocols was developed and is going to be deployed.

5.4.2. Taaable: a system for retrieving and creating new cooking recipes by adaptation

Participants: Julien Cojan, Valmi Dufour-Lussier, Inaki Fernandez, Emmanuelle Gaillard, Laura Infante-Blanco, Florence Le Ber, Jean Lieber, Amedeo Napoli, Emmanuel Nauer [contact person], Yannick Toussaint.
Taaable is a system whose objectives are to retrieve textual cooking recipes and to adapt these retrieved recipes whenever needed. Suppose that someone is looking for a “leek pie” but has only an “onion pie” recipe: how can the onion pie recipe be adapted?

The Taaable system combines principles, methods, and technologies of knowledge engineering, namely case-based reasoning (CBR), ontology engineering, text mining, text annotation, knowledge representation, and hierarchical classification. Ontologies for representing knowledge about the cooking domain, and a terminological base for binding texts and ontology concepts, have been built from textual web resources. These resources are used by an annotation process for building a formal representation of textual recipes. A CBR engine considers each recipe as a case, and uses domain knowledge for reasoning, especially for adapting an existing recipe w.r.t. constraints provided by the user, holding on ingredients and dish types.

The Taaable system is available online at http://taaable.fr. After being ranked twice second, in the 2008 and 2009 “Computer Cooking Contests” organized during the ICCBR conference, Taaable won the first price and the adaptation challenge, in 2010. In 2011, no contest was organized but the system has, however, been extended by two new features, both concerning knowledge acquisition using FCA [42]. The first feature uses FCA in order to enrich the domain ontology (especially the ingredient hierarchy), making the case retrieval more progressive and more precise [45]. The second feature uses FCA for extracting adaptation knowledge, in order to be able to better adapt a recipe to given constraints [47]. Current ongoing work on the Taaable project also includes formal representation of preparations [63].
5. Software

5.1. WinSnoori

contact : Yves Laprie (Yves.Laprie@loria.fr)

WinSnoori is a speech analysis software that we have been developing for 15 years. It is intended to facilitate the work of the scientist in automatic speech recognition, phonetics or speech signal processing. Basic functions of Snorri enable several types of spectrograms to be calculated and the fine edition of speech signals (cut, paste, and a number of filters) as the spectrogram allows the acoustical consequences of all the modifications to be evaluated. Beside this set of basic functions, there are various functionalities to annotate phonetically or orthographically speech files, to extract fundamental frequency, to pilot the Klatt synthesizer and to utilize PSOLA resynthesis.

The main improvement concerns automatic formant tracking which is now available with other tools for copy synthesis. It is now possible to determine parameters for the formant synthesizer of Klatt quite automatically. The first step is formant tracking, then the determination of F0 parameters and finally the adjustment of formant amplitudes for the parallel branch of the Klatt synthesizer enable a synthetic speech signal to be generated. The automatic formant tracking that has been implemented is an improved version of the concurrent curve formant tracking [60]. One key point of this tracking algorithm is the construction of initial rough estimates of formant trajectories. The previous algorithm used a mobile average applied onto LPC roots. The window is sufficiently large (200 ms) to remove fast varying variations due to the detection of spurious roots. The counterpart of this long duration is that the mobile average prevents formants fairly far from the mobile average to be kept. This is particularly sensitive in the case of F2 which presents low frequency values for back vowels. A simple algorithm to detect back vowels from the overall spectral shape and particularly energy levels has been added in order to keep extreme values of F2 which are relevant.

Together with other improvements reported during the last years, formant tracking enables copy synthesis. The current version of WinSnoori is available on http://www.winsnoori.fr.

5.2. SUBWEB

contacts : David Langlois (langlois@loria.fr) and Kamel Smaili (smaili@loria.fr).

We published in 2007 a method which allows to align sub-titles comparable copora [61]. In 2009, we proposed an alignment web tool based on the developed algorithm. It allows to: upload a source and a target files, obtain an alignment at a sub-title level with a verbose option, and a graphical representation of the course of the algorithm. This work has been supported by CPER/TALC/SUBWEB.\footnote{\url{http://wikitalc.loria.fr/dokuwiki/doku.php?id=operations:subweb}}

5.3. SELORIA

contact : Odile Mella (Odile.Mella@loria.fr).

SELORIA is a toolbox for speaker diarization.

\footnote{\url{http://wikitalc.loria.fr/dokuwiki/doku.php?id=operations:subweb}}
The system contains the following steps:

- **Speaker change detection**: to find points in the audio stream which are candidates for speaker change points, a distance is computed between two Gaussian modeling data of two adjacent given-length windows. By sliding both windows on the whole audio stream, a distance curve is obtained. A peak in this curve is thus considered as a speaker change point.

- **Segment recombination**: too many speaker turn points detected during the previous step results in a lot of false alarms. A segment recombination using BIC is needed to recombine adjacent segments uttered by the same speaker.

- **Speaker clustering**: in this step, speech segments of the same speaker are clustered. Top-down clustering techniques or bottom-up hierarchical clustering techniques using BIC can be used.

- **Viterbi re-segmentation**: the previous clustering step provides enough data for every speaker to estimate multi-gaussian speaker models. These models are used by a Viterbi algorithm to refine the boundaries between speakers.

- **Second speaker clustering step (called cluster recombination)**: This step uses Universal Background Models (UBM) and the Normalized Cross Likelihood Ratio (NCLR) measure.

This toolbox is derived from mClust designed by LIUM.

### 5.4. ANTS

**Contact**: Dominique Fohr (fohr@loria.fr).

The aim of the Automatic News Transcription System (ANTS) is to transcribe radio broadcast news. ANTS is composed of five stages: broad-band/narrow-band speech segmentation, speech/music classification, speaker segmentation and clustering, detection of silences/breathing segments and large vocabulary speech recognition. The three first stages split the audio stream into homogeneous segments with a manageable size and allow the use of specific algorithms or models according to the nature of the segment.

Speech recognition is based on the Julius engine and operates in two passes: in the first pass, a frame-synchronous beam search algorithm is applied on a tree-structured lexicon assigned with bigram language model probabilities. The output of this pass is a word-lattice. In the second pass, a stack decoding algorithm using a trigram language model gives the N-best recognition sentences.

A real time version of ANTS has been developed. The transcription is done in real time on a quad-core PC.

### 5.5. JSafran

**Contact**: Christophe Cerisara (Christophe.Cerisara@loria.fr).

J-Safran is the “Java Syntaxico-semantic French Analyser”. Its development has started in June 2009 from the collaboration between Parole and Talaris in the context of the RAPSODIS project. It is an open-source dependency parsing platform that is dedicated to oral speech. Its main interesting features, as compared to other similar software, are:

- It is designed for both manual and semi-automatic edition of dependency graphs, as well as for fully automatic parsing. To this end, it integrates two of the best state-of-the-art automatic parsers of the literature, the Malt Parser and the MATE parser, as well as a third experimental Maximum Entropy Markov Model-based parser developed from November 2011 in the team. It further integrates three automatic Part-of-speech taggers: the TreeTagger, the OpenNLP and MATE taggers.

- It is smoothly interfaced with the JTrans platform, thus enabling the user to directly listen to the aligned speech segments when annotating, which is an important added value to help disambiguation. The interface between both software goes well beyond simple method calls, as they both share for instance parts of the tokenization process and access a common immutable text source from the disk or on the Web.
• It supports multi-layer annotations, such as dependency relations, semantic role labeling, named entities and coreference links for instance, as well as inter-layer projection facilites.
• It offers a powerful rule-based search and tree manipulation language to transform for instance the annotation schema of a large corpus with a few commands only.
• As it is written in pure Java, it can run on any modern computer, either as a standalone application or embedded in a web page.

A description of JSafran is published in [16]. JSafran is distributed under the Cecill-C licence, and can be downloaded at http://synalp.loria.fr/?n=Research.Software

5.6. JTrans

Contact: Christophe Cerisara (Christophe.Cerisara@loria.fr).

JTrans is an open-source software for semi-automatic alignement of speech and textual corpus. It is written 100% in JAVA and exploits libraries developed since several years in our team. Two algorithms are available for automatic alignment: a block-viterbi and standard forced-alignement Viterbi. The latter is used when manual anchors are defined, while the former is used for long audio files that do not fit in memory. It is designed to be intuitive and easy to use, with a focus on GUI design. The rationale behind JTrans is to let the user control and check on-the-fly the automatic alignment algorithms. It is bundled for now with a French phonetic lexicon and French models.

Recent improvements include its integration within the JSafran platform and its release as a Java applet that can be demonstrated on web pages. During the last three months, JTrans has been downloaded about 120 times and seven users of JTrans, outside LORIA, have directly contacted the team for requests about JTrans.

JTrans is developed in the context of the CPER MISN TALC project, in collaboration between the Parole and Talaris INRIA teams, and CNRS researchers from the ATILF laboratory. It is distributed under the Cecill-C licence, and can be downloaded at http://synalp.loria.fr/?n=Research.Software

5.7. STARAP

contact: Dominique Fohr (fohr@loria.fr).

STARAP (Sous-Titrage Aidé par la Reconnaissance Automatique de la Parole) is a toolkit to help the making of sub-titles for TV shows. This toolkit performs:

• Parameterization of speech data;
• Clustering of parameterized data;
• Gaussian Mixture Models (GMM) training;
• Viterbi recognition.

This toolkit was realised in the framework of the STORECO contract and the formats of the input and output files are compatible with HTK toolkit.

5.8. TTS SoJA

contact: Vincent Colotte (Vincent.Colotte@loria.fr).

TTS SoJA (Speech synthesis platform in Java) is a software of text-to-speech synthesis system. The aim of this software is to provide a toolkit to test some steps of natural language processing and to provide a whole system of TTS based on non uniform unit selection algorithm. The software performs all steps from text to the speech signal. Moreover, it provides a set of tools to elaborate a corpus for a TTS system (transcription alignment, ... ). Currently, the corpus contains 1800 sentences (about 3 hours of speech) recorded by a female speaker.
Most of the modules are developed in Java. Some modules are in C. The platform is designed to make easy the addition of new modules. The software runs under Windows and Linux (tested on Mandriva, Ubuntu). It can be launched with a graphical user interface or directly integrated in a Java code or by following the client-server paradigm.

The software license should easily allow associations of impaired people to use the software. A demo web site has been built: http://soja-tts.loria.fr

5.9. Corpus Recorder

Contact: Vincent Colotte (Vincent.Colotte@loria.fr).

Corpus Recorder is a software for the recording of audio corpora. It provides an easy tool to record with a microphone. The gain of the audio input is controlled during the recording. From a list of sentences, the output is a set of wav files automatically renamed with textual information given in input (nationality, speaker language, gender...). An easy syntactic tagging allows to display a textual context of the sentence to pronounce. This software is suitable for recording sentences with information to guide the speaker.

The software is developed in Tcl/Tk (tested under Windows and Linux). It was used for the recording of sentences for the TTS system SOJA and during the Intonale Project (Prosody Modeling).
5. Software

5.1. Mixed camera platform

We started to develop a multiple camera platform composed of both high-definition color cameras and low-resolution depth cameras. This platform combines the advantages of the two camera types. On one side, depth (time-of-flight) cameras provide relatively accurate 3D scene information. On the other side, color cameras provide information allowing for high-quality rendering. The software package developed during the year 2011 contains the calibration of TOF cameras, alignment between TOF and color cameras, and image-based rendering. These software developments are performed in collaboration with the Samsung Advanced Institute of Technology. The multi-camera platform and the basic software modules are products of 4D Views Solutions SAS, a start-up company issued from the PERCEPTION group.

5.2. Audiovisual robot head

We have developed two audiovisual (AV) robot heads: the POPEYE head and the NAO stereo head. Both are equipped with a binocular vision system and four microphones. The software modules comprise stereo matching and reconstruction, sound-source localization and audio-visual fusion. POPEYE has been developed within the European project POP (http://perception.inrialpes.fr/POP) in collaboration with the project-team MISTIS and with two other POP partners: the Speech and Hearing group of the University of Sheffield and the Institute for Systems and Robotics of the University of Coimbra. The NAO stereo head is being developed under the European project HUMAVIPS (http://humavips.inrialpes.fr) in collaboration with Aldebaran Robotics (which manufactures the humanoid robot NAO) and with the University of Bielefeld, the Czech Technical Institute, and IDIAP. The software modules that we develop are compatible with both these robot heads.
5. Software

5.1. OMiSCID Middleware for Distributed Multi-Modal Perception

Participants: Patrick Reignier, Dominique Vaufreydaz [correspondant], Amaury Negre, Remi Barraquand.

OMiSCID is new lightweight middleware for dynamic integration of perceptual services in interactive environments. This middleware abstracts network communications and provides service introspection and discovery using DNS-SD ([DNS-based Service Discovery](#)). Services can declare simplex or duplex communication channels and variables. The middleware supports the low-latency, high-bandwidth communications required in interactive perceptual applications. It is designed to allow independently developed perceptual components to be integrated to construct user services. Thus our system has been designed to be cross-language, cross-platform, and easy to learn. It provides low latency communications suitable for audio and visual perception for interactive services.

OMiSCID has been designed to be easy to learn in order to stimulate software reuse in research teams and is revealing to have a high adoption rate. To maximize this adoption and have it usable in projects involving external partners, the OMiSCID middleware has been released under an open source licence. To maximize its target audience, OMiSCID is available from a wide variety of programming languages: C++, Java, Python and Matlab. A website containing informations and documentations about OMiSCID has been set up to improve the visibility and promote the use of this middleware.

The OMiSCID graphical user interface (GUI) is an extensible graphical application that facilitates analysis and debugging of service oriented applications. The core functionality of this GUI is to list running services, their communication channels and their variables. This GUI is highly extensible and many modules (i.e. plugins) have been created by different members of the team: figure 4 shows an example of some of these modules. OMiSCID GUI is based on the Netbeans platform and thus inherits from its dynamic installation and update of modules.

5.2. 3D Bayesian Tracker

Participants: James Crowley [correspondant], Amaury Negre, Lukas Rummelhard.

The 2DBT and 3DBT tracking systems are autonomic perceptual components originally created for the IST CAVIAR project and the IST CHIL projects. Both systems are autonomous perceptual components managed by an autonomic supervisor. The Autonomic supervisor provides self monitoring, self repair, self configuration, auto-regulation of parameters and self-description.

The INRIA 3D Bayesian body tracker is used to detect, locate and track multiple 3D entities in real time. It is configured and optimized for detecting and tracking people within rooms using multiple calibrated cameras. The system currently uses corner mounted cartesian cameras, ceiling mounted cameras with wide angle lenses and panoramic cameras placed on tables. Cameras may be connected and disconnected while the component is running, but they must be pre-calibrated to a common room reference frame. We are currently experimenting with techniques for Bayesian estimation of camera parameters for auto-calibration.

This perceptual component can be configured to monitor and track the activity within a smart space. The tracker receives its observations from 2D detection process that can use any available pixel level detection algorithm. The tracker currently integrates information from adaptive background subtraction, motion detection, skin color detection, and local appearance using scale normalised Gaussian derivatives. A common scenario is to use the motion to detect and initialise tracking, adaptive background subtraction to track 3D bodies, and skin color to track hands and faces. Cameras may be connected dynamically.
Figure 4. OMiSCID GUI showing a list of running services and some modules for service interconnections, variable plotting, live video stream display and variable control
Figure 5. The 3D Bayesian tracker integrates observations from multiple sensors
This work is currently supported by ICT Labs thematic actions on Smart Spaces and Smart Energy systems. The original system 3DBT has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.490023.000.S.P.2006.000.10000. A revised declaration for the latest version of the system is currently being prepared.

5.3. Stereo Viewfinder

Participants: Frédéric Devernay [correspondant], Elise Mansilla, Loic Lefort, Sergi Pujades.

This software has been filed with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370083.000.S.P.2007.000.10000

5.4. Tracking Focus of Attention for Large Screen Interaction

Participants: Claudine Combe, John Alexandre Ruiz Hernandez, Varun Jain, James Crowley [correspondant].

Large multi-touch screens may potentially provide a revolution in the way people can interact with information in public spaces. Technologies now exist to allow inexpensive interactive displays to be installed in shopping areas, subways and urban areas. Thesis displays can provide location aware access to information including maps and navigation guidance, information about local businesses and and commercial activities. While location information is an important component of a users context, information about the age and gender of a user, as well as information about the number of users present can greatly enhance the value of such interaction for both the user and for local commerce and other activities.

The objective of this task is to leverage recent technological advances in real time face detection developed for cell phones and mobile computing to provide a low-cost real time visual sensor for observing users of large multi-touch interactive displays installed in public spaces. The initial requirements for this system were expressed by the recent INRIA start-up HiLabs, created in 2008. By the end of 2010, HiLabs had installed over 100 interactive displays in public spaces, mostly in the form of interactive shop windows for travel agents, real-estate agents and banks. HiLabs customers indicated a potential important gain in market if such displays could be made aware of the number, gender and age of users.

The software developed for this activity builds on face detections software that has recently been developed by INRIA for the French OSEO project MinImage. MinImage was a five year, multi-million euro project to develop next generation technologies for integrated digital imaging devices to be used in cellphones, mobile and lap-top computing devices, and digital cameras, that has begun in February of 2007. The project scope included research on new forms of retinas, integrated optics, image formation and embedded image processing. INRIA was responsible for embedded algorithms for real time applications of computer vision.

Within MinImage, INRIA developed embedded image analysis algorithms using image descriptors that are invariant to position, orientation and scale and robust to changes in viewing angle and illumination intensity. INRIA proposed use of a simple hardware circuit to compute a scale invariant Gaussian pyramid as images acquired by the retina. Sums and differences of image samples from the pyramid provide invariant image descriptors that can be used for a wide variety of computer vision applications including detection, tracking and recognition of visual landmarks, physical objects, commercial logos, human bodies and human faces. Detection and tracking of human faces was selected as benchmark test case. This work has been continued with support from EIT ICT Labs, to provide context information for interaction with large multi-touch interactive displays installed in public spaces.

Multitouch interactive displays are increasingly used in outdoor and public spaces. This objective of this task is to provide a visual observation system that can detect and count users of a multitouch display and to estimate information such as the gender, and age category of each user, us rendering the system sensitive to environmental context.
SuiviDeCiblesCouleur locates individuals in a scene for video communications. FaceStabilisationSystem renormalises the position and scale of images to provide a stabilised video stream. SuiviDeCiblesCouleur has been declared with the APP "Agence pour la Protection des Programmes" under the Interdeposit Digital number IDDN.FR.001.370003.000.S.P.2007.000.21000.

A revised APP is under preperation for new versions of this software for face detection, face tracking, gender and age estimation, and orientation estimation.
5. Software

5.1. SUP

SUP is a Scene Understanding Software Platform written in C and C++ (see Figure 2). SUP is the continuation of the VSIP platform. SUP is splitting the workflow of a video processing into several modules, such as acquisition, segmentation, etc., until scenario recognition. Each module has a precise interface, and different plugins implementing these interfaces can be used for each step of the video processing. This generic architecture is designed to facilitate:

1. integration of new algorithms in SUP;
2. sharing of the algorithms among the team.

Currently, 15 plugins are available, covering the whole processing chain. Several plugins are using the Genius platform, an industrial platform based on VSIP and exploited by Keeneo, the Orion/Pulsar spin off created in July 2005.

Goals of SUP are twofolds:

1. From a video understanding point of view, to allow the researchers of the Pulsar team can share the implementations of their researches through this platform.
2. From a software engineering point of view, to integrate the results of the dynamic management of the applications when applied to video surveillance.

5.2. ViSEval

ViSEval is a software dedicated to the evaluation and visualization of video processing algorithm outputs. The evaluation of video processing algorithm results is an important step in video analysis research. In video processing, we identify 4 different tasks to evaluate: detection of physical objects of interest, classification of physical objects of interest, tracking of physical objects of interest and event recognition.
The proposed evaluation tool (ViSEvAl, visualization and evaluation) respects three important properties:

- To be able to visualize the algorithm results.
- To be able to visualize the metrics and evaluation results.
- For users to easily add new metrics.

The ViSEvAl tool is composed of two parts: a GUI to visualize results of the video processing algorithms and metrics results, and an evaluation program to evaluate automatically algorithm outputs on large amount of data. An XML format is defined for the different input files (detected objects from one or several cameras, ground-truth and events). XSD files and associated classes are used to check, read and write automatically the different XML files. The design of the software is based on a system of interfaces-plugins. This architecture allows the user to develop specific treatments according to her/his application (e.g. metrics). There are 6 interfaces:

1. The video interface defines the way to load the images in the interface. For instance the user can develop her/his plugin based on her/his own video format. The tool is delivered with a plugin to load JPEG image, and ASF video.
2. The object filter selects which objects (e.g. objects far from the camera) are processed to compute the evaluation. The tool is delivered with 3 filters.
3. The distance interface defines how the detected objects match the ground-truth objects based on their bounding box. The tool is delivered with 3 plugins comparing 2D bounding boxes and 3 plugins comparing 3D bounding boxes.
4. The frame metric interface implements metrics (e.g. detection metric, classification metric, ...) which can be computed on each frame of the video. The tool is delivered with 5 frame metrics.
5. The temporal metric interface implements metrics (tracking metric,...) which are computed on the whole video sequence. The tool is delivered with 3 temporal metrics.
6. The event metric interface implements metrics to evaluate the recognized events. The tool is delivered with 4 metrics.

The GUI is composed of different parts (see Figure 3):

- Window 1: the video part displays the current image and information about the detected and ground-truth objects (bounding-boxes, identifier, type,...).
- Window 2: the 3D virtual scene displays a 3D view of the scene (3D avatars for the detected and ground-truth objects, context, ...).
- Window 3: the temporal information about the detected and ground truth objects, and about the recognized and ground-truth events.
- Window 4: the description part gives detailed information about the objects and the events.
- Window 5: the metric part shows the evaluation results of the frame metrics.
- The object window enables the user to choose the object to be displayed (see Figure 4).
- The multi-view window displays the different points of view of the scene (see Figure 5).

The evaluation program saves, in a text file, the evaluation results of all the metrics for each frame (whenever it is appropriate), for all video sequences and for each object of the ground truth.

The ViSEvAl software was tested and validated into the framework of the Cofriend project through its partners (Akka,...). The tool is also used by IMRA, Nice hospital, Institute for Infocomm Research (Singapore),... The software version 1.0 was delivered to APP (French Program Protection Agency) on August 2010. ViSEvAl is under GNU Affero General Public License AGPL (http://www.gnu.org/licenses/) since July 2011. The tool is available on the web page: http://www-sop.inria.fr/teams/pulsar/EvaluationTool/ViSEvAl_Description.html
Figure 3. GUI of the ViSEvAl software

Figure 4. The object windows enables users to choose the object to display
5.3. Pegase

Since September 1996, the Orion team (and now the Pulsar team) distributes the program supervision engine PEGASE, based on the LAMA platform. The Lisp version has been used at Maryland University and at Genset (Paris). The C++ version (PEGASE+) is now available and is operational at ENSI Tunis (Tunisia) and at CEMAGREF, Lyon (France).

5.4. Clem

The Clem Toolkit [61] (see Figure 6) is a set of tools devoted to design, simulate, verify and generate code for LE [17] [71] programs. This latter is a synchronous language supporting a modular compilation. The language also supports automata possibly designed with a dedicated graphical editor. The Clem toolkit comes with a simulation tool. Hardware description (Vhdl) and software code (C) are generated for LE programs. Moreover, we also generate files to feed the NuSMV model checker [57] in order to perform validation of program behaviors.
Figure 6. The Clem Toolkit
4. Software

4.1. RID: Rich Intrinsic Decomposer

Participants: Pierre-Yves Laffont, Adrien Bousseau, George Drettakis.

We developed a software platform to perform rich intrinsic decomposition methods from photographs of outdoor scenes, as described in [21] and in an article currently submitted for publication. It includes main scripts and functions in Matlab for treatment of the input data, interfaces to software for multi-view reconstruction (Bundler, PMVS) and meshing from point clouds (method developed by Julie Digne, a postdoc in the Geometrica team). We then interface software for image matting using the Matting Laplacian, and User-Assisted Intrinsic Images. The system also includes an interface with Adobe Photoshops, for visualization and demonstration of our results in end-user image editing software. The method performs the computation of sun, sky and indirect lighting received at 3D points of an automatically reconstructed scene, using a modified version of the PBRT stochastic raytracer. Finally, there is a scene calibration module and an OpenGL viewer.

4.2. Imere: Inria Multi-Environment Realistic Simulation Engine

Participants: Adrien David, George Drettakis.

In the context of the ADT Interact3D and the ARC NIEVE, we developed Imere, a middleware to be used as a VR engine, helping in the implementation of realistic simulations for immersive installations. Imere provides a wrapper to OSG’s (OpenSceneGraph) deep scene graph and its traversals abilities into an abstracted collection of high level objects which directly represent realistic entities (such as indoor elements, machines and realistic characters). It provides capacities such as skeletal animations or spatialized audio by interfacing with APF, while its clear composite pattern allows implementing more behaviors easily.

Finally, a generic design based on triggers and functors lets the final user implement complex scenarios of VR applications with the feeling of writing a script in C++. Applications developed on top of Imere plug transparently into osgVR developed in the DREAM group (i.e., the research support development group of our INRIA center). We are using osgVR to render OSG’s scene graph in a distributed manner, since rendering clusters are available in an increasing number of installations. osgVR is a software layer developed by the DREAM research support group, ensuring synchronization and events/inputs distribution among a list of rendering slaves. These two libraries are available on GForge.

4.3. APF: state-of-the-art 3D audio library

Participants: Adrien David, George Drettakis.

This work was performed in collaboration with Jean-Christophe Lombardo of the DREAM research engineer service at INRIA Sophia-Antipolis Méditerranée. REVES has several audio research publications over the last 10 years, which correspond to a class of functionalities. The first component is the masking or culling algorithm, which aims at removing all the inaudible audio sources from a virtual scene based on perceptual metrics. The second component, called clustering, aims at grouping audio sources that are spatially close to each other and premix them to a representative cluster source, so that all spatialization related processing can be applied only on the representative premixed source [9]. Other audio topics were also considered and developed, like progressive and scalable frequency domain mixing, sound propagation, scalable reverberation, modal sound synthesis and contact sounds generation [1].
In order to maintain all the knowledge in the group and re-use these technologies in the Immersive Space, a previous young engineer, a previous engineer (David Grelaud) wrote a fully documented audio library (APF) which gathers about 10 audio publications and 1 US patent. APF is a cross-platform, object oriented C++ API available on GForge. All the code has been re-implemented and a completely new software architecture resulted in a twofold increase in the speed of our algorithms. APF runs in the Immersive Space and uses the tracking system to spatialize virtual audio sources around the listener. It can also exploit personal Head Related Transfer Functions (HRTF).

We have implemented a network communications layer to create an audio rendering server on a separate machine, and the library is fully integrated into the osgVR platform.

APF has also been critical in establishing collaborations in the context of various grant proposals (EU and national).

4.4. GaborNoise Software

Participants: Ares Lagae, George Drettakis.

We proposed a new procedural noise function last year, Gabor noise [6]. In the context of this project, we have developed a software package, which includes a CPU reference implementation of the 2D noise, and a complete GPU implementation of the 2D noise, surface noise, and 3D noise. This software package has been filed for APP protection and is in the process of being transferred to industrial partners.

This work is a collaboration with Sylvain Lefebvre, former member of the team, now at INRIA Nancy.
5. Software

5.1. LEOPAR

Participants: Bruno Guillaume [correspondant], Guy Perrier, Mathieu Morey, Paul Masson.

5.1.1. Software description

LEOPAR is a parser for natural languages which is based on the formalism of Interaction Grammars [35]. It uses a parsing principle, called “electrostatic parsing” which consists in neutralizing opposite polarities. A positive polarity corresponds to an available linguistic feature and a negative one to an expected feature.

Parsing a sentence with an Interaction Grammar consists in first selecting a lexical entry for each of its words. A lexical entry is an underspecified syntactic tree, a tree description in other words. Then, all selected tree descriptions are combined by partial superposition guided by the aim of neutralizing polarities: two opposite polarities are neutralized by merging their support nodes. Parsing succeeds if the process ends with a minimal and neutral tree. As IGs are based on polarities and under-specified trees, LEOPAR uses some specific and non-trivial data-structures and algorithms.

The electrostatic principle has been intensively considered in LEOPAR. The theoretical problem of parsing IGs is NP-complete; the nondeterminism usually associated to NP-completeness is present at two levels: when a description for each word is selected from the lexicon, and when a choice of which nodes to merge is made. Polarities have shown their efficiency in pruning the search tree:

- In the first step (tagging the words of the sentence with tree descriptions), we forget the structure of descriptions, and only keep the bag of their features. In this case, parsing inside the formalism is greatly simplified because composition rules reduce to the neutralization of a negative feature-value pair \( f \leftarrow v \) by a dual positive feature-value pair \( f \rightarrow v \). As a consequence, parsing reduces to counting the number of positive and negative polarities present in the selected tagging for every pair \((f, v)\): every positive occurrence counts for +1 and every negative occurrence for -1, the sum must be 0.
- Again in the tagging step, original methods were developed to filter out bad taggings. Each unsaturated polarity \( p \) in the grammar induces constraints on the set of contexts in which it can be used: the unsaturated polarity \( p \) must find a companion (i.e., a tree description able to saturated it); and the set of companions for the polarity \( p \) can be computed statically from the grammar. Each lexical selection which contains an unsaturated polarity without one of its companions can be safely removed.
- In the next step (node-merging phase), polarities are used to cut off parsing branches when their trees contain too many non-neutral polarities.

5.1.2. Current state of the implementation

LEOPAR is presented and documented at http://leopar.loria.fr; an online demonstration page can be found at http://leopar.loria.fr/demo.

It is open-source (under the CECILL License http://www.cecill.info) and it is developed using the InriaGforge platform (http://gforge.inria.fr/projects/semagramme/).

The main features of current software are:

- automatic parsing of a sentence or a set of sentences,
- dependency and parse-tree representation of sentences,
- interactive parsing (the user chooses the couple of nodes to merge),
- visualization of grammars produced by XMG or of sets of description trees associated to some word in the linguistic resources,
During 2011, with the help of an engineer, the LEOPAR software was improved in several ways:

- A new graphical interface (using GTK) was designed
- New algorithms for the super-tagging step of the parsing process were implemented. These algorithms are described in [9].

5.2. ACG Development Toolkit

In order to support the theoretical work on ACG, we have been developing a support system. The objectives of such a system are twofold:

1. to make possible to implement and experiment grammars the modeling of linguistic phenomena;
2. to make possible to implement and experiment results related to the ACG formalisms. Such results can concern parsing algorithms, type extensions, language extensions, etc.

The current version of the ACG development toolkit prototype issues from a first release published in October 2008. Further releases have been published before the ESSLLI 2009 course on ACG. It focuses on providing facilities to develop grammars. To this end, the type system currently implemented is the linear core system plus the (non-linear) intuitionistic implication, and a special attention has been paid to type error management. As a major limitation, this version only considers transformation from abstract terms to object terms, and not the other way around.

Enabling transformation from the object terms to the abstract terms is the first step of future development for the ACG support system. A parsing algorithm based on [37]’s methods is being implemented for second-order ACGs. It is based on a translation of ACG grammars into Datalog programs and is well-suited to fine-grained optimization. A summer internship from ENS Cachan, Clovis Eberhart (L3) has been implementing the translation from the higher-order signatures and terms data structures to the Datalog clauses data structures. It still remains to be integrated to the main branch.

In order to allow for a larger character set as input, another extension implemented this summer by another internship from École des Mines de Nancy, Grégoire Brenon (M1) was to extend the lexer and the parser for the data files with UTF-8 capabilities (OCaml lacks such a built-in capability).

However, since we’re interested not only by recognizability (hence whether some fact is provable) but also by the parsing structure (hence the proof), the Datalog solver requires further adaptations. Note however that in the general case, the decidability of translating an object term to an abstract one is still an open problem.

5.3. GREW

Participants: Bruno Guillaume [correspondant], Guy Perrier, Mathieu Morey, Paul Masson.

Grew is a Graph Rewriting tools dedicated to applications in NLP. It was developed as a support tool during the PhD thesis of Mathieu Morey.

It is freely-available (from the page http://wikilligramme.loria.fr/doku.php?id=grew:grew ) and it is developed using the InriaGforge platform (http://gforge.inria.fr/projects/semagramme/)

We list below some of the major specificities of the GREW software.

- Graph structures can use a build-in notion of feature structures.
- The left-hand side of a rule is described by a graph called a pattern; injective graph morphisms are used in the pattern matching algorithm.
- Negative pattern can be used for a finer control on the left-hand side of rules.
- The right-hand side or rules is described by a sequence of atomic commands that describe how the graph should be modified during the rule application.

1 Available at http://acg.gforge.inria.fr with a CeCILL license.
• Subset of rules are grouped in modules; the full rewriting process being a sequence of module applications.
• The GREW software has support both for confluent and non-confluent modules; when a non-confluent modules is used, all normal forms are returned and then ambiguity is handled in a natural way.
• GREW can be used on Corpus mode with statistics about rules usage or with an a Graphical User Interface which can show all intermediate graphs used during the rewriting process (useful either to debug rewriting system of for demonstrations).

During the last 18 months, the GREW software were used for several kind of applications manipulating syntactic and/or semantic graph representations:
• to build DMRS semantic representation from syntactic dependency trees ( [ 26 ], [ 14 ], [ 9 ]); 
• to enrich surface syntactic structures ([ 13 ], [ 9 ]); 
• to detect annotation errors in the French Treebank.

5.4. Other developments

Participants: Bruno Guillaume [correspondant], Paul Masson.

Other peripheral developments of the team are available either as web service of as downloadable code:
• A concordancer named CONDOR. The main features of this tool are:
  – It is usable online: http://condor.loria.fr ;
  – It is possible to search for all inflexions (given by a lexicon) of some words;
  – It is possible to combine two searches and to search for a couple of words to find collocations.

• A program (named DEP2PICT) to build graphical representations of dependency structures.
  – it is presented and documented at: http://dep2pict.loria.fr ;
  – it is usable online at http://dep2pict.loria.fr/demo ;
  – it can produce PNG, SVG and PDF output formats;
  – it can be used to represented dependency structures with chunks;
  – it support CONLL input format.
5. Software

5.1. Introduction

In our research domain, developing software prototypes is mandatory to validate research solutions and is an important vector for publications, demonstrations at conferences and exhibitions as well as for cooperations with industry. This prototyping task is however difficult because it requires specialized hardware platforms (e.g., new generations of smart tokens), themselves sometimes at an early stage of development.

For a decade, we have developed successive prototypes addressing different application domains, introducing different technical challenges and relying on different hardware platforms. PicoDBMS was our first attempt to design a full-fledged DBMS embedded in a smart card [9] [27]. Chip-Secured Data Access (C-SDA) embedded a reduced SQL query engine and access right controller in a secure chip and acted as an incorruptible mediator between a client and an untrusted server hosting encrypted data [33]. Chip-Secured XML Access (C-SXA) was an XML-based access rights controller embedded in a smart card [8]. Prototypes of C-SXA have been the recipient of the e-gate open 2004 Silver Award and SIMagine 2005 Gold award, two renowned international software contests. The next subsections details the two prototypes we are focusing on today.

5.2. PlugDB engine

Participant: Nicolas Anciaux.

More than a stand-alone prototype, PlugDB is part of a complete architecture dedicated to a secure and ubiquitous management of personal data. PlugDB aims at providing an alternative to a systematic centralization of personal data. To meet this objective, the PlugDB architecture lies on a new kind of hardware device called Secure Portable Token (SPT). Roughly speaking, a SPT combines a secure microcontroller (similar to a smart card chip) with a large external Flash memory (Gigabyte sized). The SPT can host data on Flash (e.g., a personal folder) and safely run code embedded in the secure microcontroller. PlugDB engine is the cornerstone of this embedded code. PlugDB engine manages the database on Flash (tackling the peculiarities of NAND Flash storage), enforces the access control policy defined on this database, protects the data at rest against piracy and tampering, executes queries (tackling low RAM constraint) and ensures transaction atomicity. Part of the on-board data can be replicated on a server (then synchronized) and shared among a restricted circle of trusted parties through crypto-protected interactions. PlugDB engine has been registered at APP (Agence de Protection des Programmes) in 2009 [29] and its Flash-based indexing system has been patented by INRIA and Gemalto [37]. It has been demonstrated in a dozen of national and international events including JavaOne and SIGMOD. It is being experimented in the field to implement a secure and portable medical-social folder helping the coordination of medical care and social services provided at home to dependent people.


5.3. uFLIP Benchmark

Participant: Luc Bouganim.

It is amazingly easy to produce meaningless results when measuring flash devices, partly because of the peculiarity of flash memory, but primarily because their behavior is determined by layers of complex, proprietary, and undocumented software and hardware. uFLIP is a component benchmark for measuring the response time distribution of flash IO patterns, defined as the distribution of IOs in space and time. uFLIP includes a benchmarking methodology which takes into account the particular characteristics of flash devices. The source code of uFLIP, available on the web (700 downloads, 4000 distinct visitors), was registered at APP in 2009 [31]. It has been demonstrated at SIGMOD.

5. Software

5.1. GenI

Participants: Claire Gardent [correspondent], Eric Kow [developer], Carlos Areces [developer].

GenI is a surface realiser that generates sentences from first order logical formulae. It is implemented in Haskell and uses the Glasgow Haskell compiler to obtain executable code for Windows, Solaris, Linux and Mac OS X. GENI is compatible with both a grammar for French (SEMTAG) and for English (SEMXTAG), both grammars being produced using the XMG MetaGrammar Compiler. SEMTAG covers the basic syntactic structures of French as described in Anne Abeillé’s book “An Electronic Grammar for French”. SEMXTAG has a coverage similar to that of XTAG, the TAG grammar for English developed by the University of Pennsylvania. GenI is under GPL License. See also the web page http://talc.loria.fr/GenI-un-realisateur-de-surface.html.

- Version: 0.20.1

5.2. Web Service for the Multilingual-Assisted Chat Interface

Participant: Samuel Cruz-Lara [correspondent].

The Web Service for the Multilingual-Assisted Chat Interface program (WSMACI) is a linguistic assistant for virtual worlds. Its first version is dedicated to English assistance in such worlds. It has been developed in the context of the Metaverse1 project. It provides the end-users with MLIF-based provision of sentence analysis and word information (synonyms, definitions, translations) based on Google Translate, WordNet and the Brown Corpus.

- Version: 0.2

5.3. Emotion detection from textual information

Participant: Samuel Cruz-Lara [correspondent].

The 4 Layers Emotion Detection program (4LED) is an emotion detection tool. The emotions are extracted from texts in particular, from chat interfaces in virtual worlds. It has been developed in the context of the Metaverse1 project. The emotion detection process is based on SMILEY detection using WordNet-Domains and Tree-Tagger-based rules, WordNet-Affect, and keywords. http://talc.loria.fr/~metaverse/web_test/emotions/filterDetection/corpusCreation.php.

- Version: 0.2

5.4. Second Life Magic Carpet

Participant: Samuel Cruz-Lara [correspondent].

The Second Life Magic Carpet program (SLMC) is an assistant whose role is to guide people through virtual worlds with textual instructions. It has been developed in the context of the Metaverse1 project. It analyses the instructions of the visitors in order to find where they want to go, using web services for the analysis, for synonyms retrieving and for path finding.

- Version: 0.2

5.5. WikiAnalyzer

Participant: Alexandre Denis [correspondent].
The WikiAnalyzer is a tool developed in the CCCP-Prosodie project that aims to describe participants of Wikipedia projects. It provides a range of linguistic and structural analyses of Wikipedia discussion pages. The tool performs pages retrieval and automatic annotation of markers to build interactive profiles of participants. These profiles include information such as their level of expertise in the domain, the use of subjective elements in their contributions, the connotation of the terms they use and enable to describe participants relative to their degree of conflictuality in the discussion. The structural analyses are parallel analyses on the structure of messages, enabling to categorize participants with regards to the type of contribution (starting a thread, participants they answer to, etc.). The tool has been developed in Java, and will be released as an online web application to the other members of the CCCP-Prosodie project.

- Version: 0.8

5.6. Emospeech Dialogue Toolkit

**Participant:** Lina-Maria Rojas Barahona [correspondent].

The Emospeech Dialogue Toolkit is a multi-agent architecture for developing man/machine dialog systems in the context of a video game. It includes the following agents:

- Midiki Dialogue Manager: We extended and improved the open source MIDIKI (MITRE Dialogue Toolkit) software to support the multi-agent architecture and the configuration from a relational database.
- Wizard of Oz: We implemented two Wizard of OZ interfaces which allow a human to interact with other agents in the dialogue architecture. The free-wizard acts as a dialogue manager and permits a chat between two humans the player and the Wizard while simultaneously storing all interactions in a database. In contrast, the semi-automatic wizard connects the Wizard with Midiki, whereby the Wizard interprets and adjusts Midiki generation.
- Answer Selection: We trained a classifier with Conditional Random Fields that chooses the most plausible response to a player utterance.

In addition, we trained a Logistic Regression Classifier for the interpretation agent that communicates with MIDIKI.

The dialogue agents communicate with the Game Agent, Speech Recognition and/or Chatbox agents developed by the Parole team. The Wizard of Oz, in which a human simulates a dialogue system, is used to collect dialogue data which can be used for training the interpreter and/or the Answer Selection Classifier. Moreover, a Dialogue Configuration Tool has been implemented for the configuration of several dialogues for different game scenarios by configuring the characters and goals in the game and the goals to be discussed in each dialogue. (See [http://tulc.loria.fr:8081/EmoDial](http://tulc.loria.fr:8081/EmoDial)).

- Version: 1.0

5.7. IGNG-Fv2

**Participant:** Jean-Charles Lamirel [correspondent].

The IGNG-Fv2 program implements a new incremental clustering algorithm whose main domain of application is the statistical analysis of continuous flow of evolving textual data, as well as the one of static textual data. It has been developed in the context of the CPER TALC (McFiID action). It is based on a generic adaptation of the classical neural-based clustering approaches relying on gas of neurons with free topology. The IGNG-Fv2 approach exploits a combination of distance based and cluster data feature maximization criteria. This approach has been proved more efficient than the usual techniques for the analysis all kinds of static textual datasets. Considering its incremental character, it can also provide the information analysts with precise online detection of topic changes in the curse of a textual information flow.
5.8. C-Quality

**Participant:** Jean-Charles Lamirel [correspondent].

The C-Quality toolkit provides method-independent clustering quality measures and cluster labeling techniques specifically adapted to the interpretation of data analysis performed on textual data. The toolkit relies on an evaluation approach based on the exploitation of the maximized features of the data associated to each cluster after the clustering process without prior consideration of clusters profiles. The toolkit basic role is to act as an overall clustering quality evaluation tool. In a complementary way toolkit’s clusters labeling functionalities can be used altogether for visualizing or synthesizing clustering results, for optimizing learning of a clustering method, for validating cluster content and act as efficient variable selection methods in the framework of supervised or semi-supervised learning tasks.

5.9. tl_dv2_ladl, a subcategorisation lexicon for French verbs.

**Participant:** Ingrid Falk [correspondent].

tl_dv2_ladl is a subcategorisation lexicon for French verbs produced by merging three lexicons which were built or validated manually: Dicovalence (version 2), TreeLex and the LADL tables. tl_dv2_ladl lists subcategorisation frames for 5918 French verbs. An entry in the lexicon consists of a verb and an associated subcategorisation frame whereby each subcategorisation frame describes a set of syntactic arguments with each argument being described by a grammatical function and a syntactic category. Each entry also gives the original lexical resource the information was extracted from. tl_dv2_ladl can be downloaded from [http://talc.loria.fr/tl_dv2_ladl-a-subcategorisation.html](http://talc.loria.fr/tl_dv2_ladl-a-subcategorisation.html).

- Version: 0.1
5. Software

5.1. Oriented wavelet based image codec

Participant: Christine Guillemot [contact person].

This still image codec is based on oriented wavelet transforms developed in the team. The transform is based on wavelet lifting locally oriented according to multiresolution image geometry information. The lifting steps of a 1D wavelet are applied along a discrete set of local orientations defined on a quincunx sampling grid. To maximize energy compaction, the orientation minimizing the prediction error is chosen adaptively. This image codec outperforms JPEG-2000 for lossy compression. This software has been registered at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.260024.000.S.P.2008.000.21000.

5.2. M3DPlayer: 3D video player

Participant: Vincent Jantet [contact person].

A 3D player - named M3DPlayer - supporting rendering of a 3D scene and navigation within the scene has been developed. It integrates as a plug-in the 3D model-based video codec of the team. From a video sequence of a static scene viewed by a monocular moving camera, the 3D model-based video codec allows the automatic construction of a representation of a video sequence as a stream of textured 3D models. 3D models are extracted using stereovision and dense matching maps estimation techniques. A virtual sequence is reconstructed by projecting the textured 3D models on image planes. This representation enables 3D functionalities such as synthetic objects insertion, lightning modification, stereoscopic visualization or interactive navigation. The codec allows compression at very low bit-rates (16 to 256 kb/s in 25Hz CIF format) with a satisfactory visual quality. It also supports scalable coding of both geometry and texture information. The first version of the software was registered at the Agency for the Protection of Programmes (APP) under the number IDDN.FR.001.130017.000S.P.2003.000.41200.

A second version of the player has been registered at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.090023.000.S.P.2008.000.21000. In 2009-2010, we focused on improving the rendering engine, based on recent OpenGL extensions, to be able to render the viewed scenes on an autostereoscopic display with low-end graphic cards. In our case, auto-stereoscopic display requires the rendering of eight 1920x1200 frames instead of just one for a standard display. This player is also used to render LDI (Layered Depth Images) and LDV (Layered Depth Videos) and to visualize 3D scenes on autostereoscopic displays taking multiple input views rendered from the LDI representation.

5.3. Depth maps extractor in mono-view (M3dAnalyzer2)

Participant: Josselin Gauthier [contact person].

This software estimates depth maps from a video captured by a unique camera moving in a static 3D environment with Lambertian surfaces. These sequences are of interest to specialized applications such as augmented reality, remote-controlled robots operating in hazardous environments or remote exploration by drones. This software has been filed at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.110031.000.S.P.2010.000.31235.

5.4. Depth maps extractor in multi-view (MV2MVD)

Participant: Josselin Gauthier [contact person].
This software estimates depth maps from multi-view videos, to provide Multi-View plus Depth (MVD) videos. MVD videos can be used to synthesize virtual views of the scene, or to render a different number of views than captured in the original video, for instance on an auto-stereoscopic display. This software produces depth maps of higher quality than those generated by the Depth Estimation Reference Software from the MPEG-3DV group, in terms of virtual views synthesis quality. This software has been filed at the APP (Agence de Protection des Programmes) under the number IDDN.FR.001.110034.000.S.P.2010.000.31235.

5.5. LDI builder

**Participant:** Vincent Jantet [contact person].

This software constructs a Layered Depth Image (LDI) representation of un-rectified Multi-View + Depth (MVD) sequences. The Incremental construction scheme reduces inter-layer correlation. The generated I-LDI is compatible with the M3DPlayer, permitting 3D visualisation and free viewpoint rendering of the 3D scene. The software also implements a virtual-view rendering technique which significantly reduces ghosting artefacts by eliminating untrusted texture boundaries detected in depth maps, as well as cracking artefacts thanks to an epipolar geometry aided inpainting method.

5.6. ADT PICOVIN

**Participants:** Ronan Boitard, Laurent Guillo [contact person], Thomas Guionnet, Tangi Poirier.

The ADT Picovin is a technological development action, which works closely with the project-team TEMICS. This is a development structure which gives its support to the project-team to integrate new and relevant algorithms into the state-of-the-art codec and to take part in standardization.

The ITU-T Study Group 16 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) have created in 2010 the Joint Collaborative Team on Video Coding (JCT-VC) in order to develop a new generation video coding standard that will further reduce by 50
In 2011, the ADT mainly focused on the improvement and integration of algorithms dedicated to intra prediction. A part of our work was integrated in the HM 1.0 and then submitted and presented as a proposal in Daegu during the 4th JCT-VC meeting in January 2011. Bit rate gains that we obtained were significant but to the detriment of encoding and decoding times. That is why, all along this year, the ADT tried to reach the best tradeoff between performances and encoding/decoding times. Our solution based on linear combination of template matching predictors has been improved by taking into account several shapes of template. The very last integration in the HM (HM4.0) shows that it performs well especially for the class “screen content”. A new very promising intra prediction method, Short Distance Intra Prediction (SDIP) is being tested as part of a Core Experiment in the HM 4.0. Once it is associated to our approach, bit save gains are additive. These results will be presented during the 7th JCT-VC meeting in Geneva in November 2011.

During 2011, the ADT also took part in cross checks which aims at evaluating and testing tools studied in core experiments. As part of cross checks the ADT has run 9 tests jointly with companies such as Technicolor, Mitsubushi, Huawei, Qualcomm and Canon.

This ADT started in October 2008. It will go on for one more year through the ADT PICOVIN-P. During this year, one permanent engineer from the SED Rennes (development and experimentation department of INRIA Rennes) and one senior engineer specialized in video compression are involved in the ADT. It is supported by the technological development department of INRIA.
5. Software

5.1. Software

5.1.1. New Software

5.1.1.1. Babaz

**Participants:** Jonathan Delhumeau, Guillaume Gravier, Hervé Jégou [correspondent].

*The deposit of this software at APP is currently being processed (submitted). The software is available from its homepage, namely [http://babaz.gforge.inria.fr/](http://babaz.gforge.inria.fr/).*

Babaz is a audio database management system with an audio-based search function, which is intended for audio-based search in video archives.

It is licensed under the terms of the GNU General Public License v3.0.

5.1.1.2. Bag-of-colors

**Participants:** Sébastien Campion [correspondent], Hervé Jégou.

*Joint work with Christian Wengert (Kooba Inc.) and Matthijs Douze (INRIA LEAR and SED project-teams)*

This package implements the color descriptor proposed in our ACM Multimedia paper [48], which improves the previous color histogram representation.

The bag-of-colors software corresponds to two packages:

- The (reference) Matlab package, which was co-developed with Christian Wengert and Matthijs Douze;
- The python package (translated) was translated from Matlab by Sébastien Campion.

The Matlab version of this package is available on Github at [https://github.com/kooaba/bag-of-color/](https://github.com/kooaba/bag-of-color/).

The python version is available on the gforge INRIA server, and might be available on request.

5.1.1.3. BonzaiBoost

**Participant:** Christian Raymond [correspondent].

*The software homepage is available at [http://bonzaiboost.gforge.inria.fr/](http://bonzaiboost.gforge.inria.fr/).*

Bonzaiboost stands for boosting over small decisions trees. bonzaiboost is a general purpose machine-learning program based on decision tree and boosting for building a classifier from text and/or attribute-value data. Currently one configuration of bonzaiboost is ranked first on [http://mlcomp.org](http://mlcomp.org) a website which propose to compare several classification algorithms on many different datasets.

5.1.1.4. Don Quixotte

**Participant:** Teddy Furon [correspondent].

*This software was developed in collaboration with project-team TEMICS (P. Meerwald)*

Don Quixotte a software suite in C for Tardos Fingerprinting code (Code generation, collusion, and accusation with single and/or joint decoding).

5.1.1.5. Rare Event

**Participant:** Teddy Furon [correspondent].

*This software was developed in collaboration with project-team ASPI (F. Cérou, A. Guyader)*

Rare Event is a Matlab package for rare event probabilities and extreme quantiles estimations.
5.1.2. Most active software started before 2011

5.1.2.1. Bigimbaz

**Participant:** Hervé Jégou [correspondent].

This software is jointly maintained by Matthijs Douze, from INRIA Grenoble.

Bigimbaz is a platform originally developed in the LEAR project-team, and now co-maintained by TEXMEX. It integrates several contributions on image description and large-scale indexing: detectors, descriptors, retrieval using bag-of-words and inverted files, and geometric verification.

5.1.2.2. kertrack

**Participant:** Sébastien Campion [correspondent].

Visual graphical interface for tracking visual targets based on particle filter tracking or based on mean-shift. The deposit of this software at APP is currently being processed.

5.1.2.3. mozaic2d

**Participant:** Sébastien Campion [correspondent].

Creation of spatio-temporal mosaic based on dominant motion compensation. It depends on the Motion2D library, which computes the dominant motion, and then adjust the images by back-warping. The deposit of this software at APP is currently being processed.

5.1.2.4. PimPy

**Participant:** Sébastien Campion [correspondent].

The software homepage is available here: [http://pim.gforge.inria.fr/pimpy/](http://pim.gforge.inria.fr/pimpy/).

First APP deposit: IDDN.FR.001.260038.000.S.P.2011.000.40000

PimPy stands for Indexing Multimedia with Python (or Platform for Indexing Multimedia with Python). The aim of this module is to provide a convenient and high level API to manage common multimedia indexing tasks. It includes several features. It is used, in particular:

- to retrieve video features, such as histogram, binarized DCT descriptor, SIFT, SURF, etc;
- to detect video cuts and dissolve (GoodShotDetector);
- for fast video frame access (pyffas);
- for raw frame extraction, or video segment extraction and re-encoding;
- to search a video segment in another video (content based retrieval);
- to perform scene clustering.

5.1.2.5. Pqcodes

**Participant:** Hervé Jégou [correspondent].

This software is jointly maintained by Matthijs Douze, from INRIA Grenoble.

First APP deposit: IDDN.FR.001.220012.000.S.P.2010.000.10000

A new version of the software at APP is currently being processed.

Pqcodes is a library which implements the approximate k nearest neighbor search method of [18]. This software has been transferred to Technicolor in August 2011.

5.1.2.6. python-geohash

**Participant:** Sébastien Campion [correspondent].

The deposit of this software at APP is currently being processed.

Implementation of the Geometric Hashing algorithm of [85] to check if geometrical consistency between pairs of images.
5.1.2.7. Samusa
Participant: Sébastien Campion [correspondent].

This software is jointly maintained with Guillaume Gravier.
Samusa enable to detect speech and/or musical segment in multimedia content.

5.1.2.8. Yael
Participant: Hervé Jégou [correspondent].

This software is jointly maintained by Matthijs Douze, from INRIA Grenoble.
APP deposit: IDDN.FR.001.220014.000.S.P.2010.000.10000
A new version of the software at APP is currently being processed.
Yael is a C/python/Matlab library providing (multi-threaded, Blas/Lapack, low level optimization) implementa-
tions of computationally demanding functions. In particular, it provides very optimized functions for
k-means clustering and exact nearest neighbor search.

5.1.2.9. TVSearch
Participant: Sébastien Campion [correspondent].

TVSearch is a content based retrieval search engine used to search and propagate manual annotation such as
advertisement in a TV corpora. Based on a binary DCT descriptor, it used GPU card to compute exhaustive
Hamming distance between the query and database. For example, a query of 11 seconds in 21 days on
television (504 hours) is done in 9 seconds. (i.e., bit-rate of 2,3 days/second) TVSearch offer a web services
API using the HTTP/REST protocol.
The deposit of this software at APP is currently being processed.

5.1.2.10. AVSST
Participant: Sébastien Campion [correspondent].

AVSST is an Automatic Video Stream Structuring Tool. First, it allows the detection of repetitions in a TV
stream. Second, a machine learning method allows the classification of programs and inter-programs such as
advertisements, trailers, etc. Finally, the electronic program guide is synchronized with the right timestamps
based on dynamic time warping. A graphical user interface is provided to manage the complete workflow.

5.1.3. Other softwares

Several software programs have been developed in the team over the years:
I-DESCRIPTION (APP deposit number: IDDN.FR.001.270047.000.S.P.2003.000.21000),
ASARES, is a symbolic machine learning system that automatically infers, from descriptions of pairs of linguis-
tic elements found in a corpus in which the components are linked by a given semantic relation, corpus-specific
morpho-syntactic and semantic patterns that convey the target relation. (IDDN.FR.001.0032.000.S.C.2005.000.20900),
ANAMORPHOS, detects morphological relations between words in many languages (IDDN.FR.001.050022.000.S.P.2008.000.20900),
DIVATEX is a audio/video frame server. (IDDN.FR.001.320006.000.S.P.2006.000.40000),
NAVITEX is a video annotation tool. (IDDN.FR.001.190034.000.S.P.2007.000.40000),
TELEMEX, is a web service that enables TV and radio stream recording.
VIDSIG computes a small and robust video signature (64 bits per image).
VIDSEG computes segmentation features such as cuts, dissolves, silences in audio track, changes of ratio
aspect, monochrome images. (IDDN.FR.001.250009.000.S.P.2009.000.40000),
ISEC, web application used as graphical interface for image searching engines based on retrieval by content.
GPU-KMEANS, implementation of k-means algorithm on graphical process unit (graphic cards)
CORRESPONDENCE ANALYSIS computes a factorial correspondence analysis (FCA) for image retrieval.
GPU CORRESPONDENCE ANALYSIS, is an implementation of the previous software Correspondence Analysis on graphical processing unit (graphical card).

CAVIZ is an interactive graphical tool that allows to display and to extract knowledge from the results of a Correspondence Analysis on images.

KIWI (standing for Keywords Extractor) is mostly dedicated to indexing and keyword extraction purposes.

TOPIC SEGMENTER, is a software dedicated to topic segmentation of texts and (automatic) transcripts.

S2E (Structuring Events Extractor) is a module which allows the automatic discovery of audiovisual structuring events in videos.

2PAC, build classes of words of similar meanings ("semantic classes") specific to the use that is made of them in that given topic. (IDDN.FR.001.470028.000.S.P.2006.000.40000)

FAESTOS, (Fully Automatic Extraction of Sets of keywords for TOpic characterization and Spotting) is a tool composed of a sequence of statistical treatments that extracts from a morpho-syntactically tagged corpus sets of keywords that characterize the main topics that corpus deals with. (IDDN.FR.001.470029.000.S.P.2006.000.40000)

FISHNET, Fishnet is an automatic web pages grabber associated with a specific theme.

MATCH MAKER, semantic relation extraction by statistical methods.

IRISA NEWS TOPIC SEGMENTER (IRINTS), automatically segments speech transcripts into topic-consistent parts.

IRISAPHON, produce phonetic words.

5.2. Demonstrations

Participants: Morgan Bréhinier, Sébastien Campion [correspondent], Guillaume Gravier.

The gradual migration of television from broadcast diffusion to Internet diffusion offers tremendous possibilities for the generation of rich navigable contents. However, it also raises numerous scientific issues regarding de-linearization of TV streams and content enrichment. In this demonstration, we illustrate how speech in TV news shows can be exploited for de-linearization of the TV stream. In this context, de-linearization consists in automatically converting a collection of video files extracted from the TV stream into a navigable portal on the Internet where users can directly access specific stories or follow their evolution in an intuitive manner.

Structuring a collection of news shows requires some level of semantic understanding of the content in order to segment shows into their successive stories and to create links between stories in the collection, or between stories and related resources on the Web. Spoken material embedded in videos, accessible by means of automatic speech recognition, is a key feature to semantic description of video contents. At IRISA/INRIA Rennes, we have developed multimedia content analysis technology combining automatic speech recognition, natural language processing and information retrieval to automatically create a fully navigable news portal from a collection of video files.

The demonstration was presented in several workshops (Quaero CTC workshop, Journée INRIA Industrie La Télévision du Futur) and a video has been made available online on the portal of the EIT ICT Labs OpenSEM project.

See the demo at http://texmix.irisa.fr .

5.3. Experimental platform

Participants: Laurent Amsaleg, Sébastien Campion [correspondent], Patrick Gros, Pascale Sébillot.
Until 2005, we used various computers to store our data and to carry out our experiments. In 2005, we began some work to specify and set-up dedicated equipment to experiment on very large collections of data. During 2006 and 2007, we specified, bought and installed our first complete platform. It is organized around a very large storage capacity (155TB), and contains 4 acquisition devices (for Digital Terrestrial TV), 3 video servers, and 15 computing servers partially included in the local cluster architecture (IGRIDA).

In 2010, we have acquired a new large memory server with 144GB of RAM which is used for memory demanding tasks, in particular to improve the speed of building index or language model. The previous server dedicated to this kind of jobs (acquired in 2008) has been upgraded to 96GB of RAM.

A dedicated website has been developed in 2009 to provide a user support. It contains useful information such as references of available and ready to use software on the cluster, list of corpus stored on the platform, pages for monitoring disk space consumption and cluster loading, tutorials for best practices and cookbooks for treatments of large datasets.

In 2008, we build up a corpus of multimedia data. It consists in a continuous recording (6 months) of two TV channels and three radios. It also includes web pages related to these contents captured on broadcaster’s website. This corpus is to be used for different studies like the treatment of news along the time and to provide sub-corpus like TV news within the Quaero project (see below). The manual annotation of all the TV programs is under progress.

This platform is funded by a joint effort of INRIA, INSA Rennes and University of Rennes 1.
VR4I Team

5. Software

5.1. OpenMASK: Open-Source platform for Virtual Reality

Participants: Alain Chauffaut [contact], Ronan Gaugne [contact], Georges Dumont, Thierry Duval, Laurent Aguereche, Florian Nouviale.

OpenMASK (Open Modular Animation and Simulation Kit) is a federative platform for research developments in the VR4i team. Technology transfer is a significant goal of our team so this platform is available as OpenSource software (http://www.openmask.org).

OpenMASK is a C++ software platform for the development and execution of modular applications in the fields of animation, simulation and virtual reality. The main unit of modularity is the simulated object (OSO) which can be viewed as frequentational or reactive motors. It can be used to describe the behavior or motion control of a virtual object as well as input devices control like haptic interfaces. Two OSO communicate with synchronous data flows or with asynchronous events.

We provide Model Driven Tools to help building OpenMASK applications without tedious and repeated coding and to improve reusability. Within Eclipse environment we offer an editor and a C++ code generator to design and build objects classes. The current OpenMASK 4.2 release is now based on MPI for distribution service, Ogre3D for visualisation service. One can benefit of new interaction tools for local or remote collaborative applications.

5.2. GVT : Generic Virtual Training

Participants: Bruno Arnaldi, Valérie Gouranton [contact], Florian Nouviale, Andrès Saraos-Luna.

The aim of GVT software is to offer personalized VR training sessions for industrial equipments. The most important features are the human and equipment security in the VR training (in opposition to the real training), the optimization of the learning process, the creation of dedicated scenarios, multiple hardware configurations: laptop computer, immersion room, distribution on network, etc.

The actual kernel of GVT platform is divided into two main elements that rely on innovative models we have proposed: LORA and STORM models.

- A Behavior Engine. The virtual world is composed of behavioral objects modeled with STORM (Simulation and Training Object-Relation Model).
- A Scenario Engine. This engine is used to determine the next steps of the procedure for a trainee, and its state evolves as the trainee achieves actions. The scenario is written in the LORA language (Language for Object-Relation Application).

A commercialized version of GVT, which includes a pedagogical engine developed in CERV laboratory, proposes training on individual procedures. A prototype is also available that enables users to train on collaborative procedures with one another or with virtual humans.

In the ANR Corvette 7.1.4 and in the FUI SIFORAS 7.1.2, new features of GVT Software are proposed.

5.3. OpenViBE Software

Participants: Anatole Lécuyer [contact], Laurent Bonnet, Jozef Legény, Yann Renard.

OpenViBE is a free and open-source software devoted to the design, test and use of Brain-Computer Interfaces.
The OpenViBE platform consists of a set of software modules that can be integrated easily and efficiently to design BCI applications. Key features of the platform are its modularity, its high-performance, its portability, its multiple-users facilities and its connection with high-end/VR displays. The "designer" of the platform enables to build complete scenarios based on existing software modules using a dedicated graphical language and a simple Graphical User Interface (GUI).

This software is available on the INRIA Forge under the terms of the LGPL-V2 licence, and it was officially released in June 2009. Since then, the OpenViBE software has already been downloaded more than 300 time, and it is used by numerous entities worldwide.

Our first international tutorial about OpenViBE was held at the International BCI Meeting in June 2010 (Monterey, US), with around 30 participants.

More information, downloads, tutorials, documentation, videos are available on OpenViBE website: http://openvibe.inria.fr
5. Software

5.1. Amaya

Participant: Vincent Quint.

Amaya is an open source web editor, i.e. a tool for creating and updating documents directly on the web. Browsing features are seamlessly integrated with editing features in a uniform environment that allows users to save files locally and on remote servers as well. This follows the original vision of the web as a space for collaboration and not just a one-way publishing medium.

Amaya started as a joint effort with W3C to showcase web technologies in a fully-featured web client. The main motivation for developing Amaya was originally to provide a framework that can integrate many web technologies during their development, with the goal of demonstrating these technologies in action while taking advantage of their combination in a single, consistent environment.

Amaya now implements a number of web technologies, such as HTML and the XHTML family, CSS style sheets, generic XML, MathML (for mathematical expressions), and SVG (for vector graphics). It allows all those document formats to be edited simultaneously in compound documents. It also includes a collaborative annotation application based on RDF, XLink, and XPointer.

It is a unique tool for manipulating simultaneously different kinds of content through a formatted representation of documents, while closely following standard formats. Developed jointly with W3C, the software is distributed world wide through the W3C servers and many mirrors. It is also part of several Linux distributions.

Amaya is also used as a platform for experimenting and distributing new editing techniques and document formats developed in WAM. It provides a full implementation of the XTiger language and its constraint-driven editing feature. It also helps users to create their own document types defined as XTiger templates.

5.2. XML Reasoning Solver

Participants: Pierre Genevès, Nabil Layaïda.

The XML Reasoning Solver is a tool for the static analysis of XPath queries and XML schemas based on the latest theoretical advances [13]. It allows automated verification of properties that are expressed as logical formulas over trees. A logical formula may for instance express structural constraints or navigation properties (like e.g. path existence and node selection) in finite trees.

The tool can solve many fundamental XML problems such as satisfiability of XPath expressions in the presence of XML schemas, containment and equivalence of XPath expressions, and many other problems that can be formulated with XPath expressions and schemas (DTDs, XML Schemas, Relax-NG).

The system is implemented in Java and uses symbolic techniques (binary decision diagrams) in order to enhance its performance. It is capable of comparing path expressions in the presence of real-world DTDs (such as the W3C SMIL and XHTML language recommendations, for instance). The cost ranges from several milliseconds for comparison of XPath queries without tree types, to several seconds for queries under very large, heavily recursive, type constraints, such as the XHTML DTD. These measurements shed light for the first time on the cost of solving static analysis problems in practice. Furthermore, the analyzer generates XML counter-examples that allow program defects to be reproduced independently from the analyzer.

5.3. Timesheets Library

Participants: Fabien Cazenave, Cécile Roisin.
The goal of the Timesheets library is to synchronize HTML5 content using declarative synchronization languages defined by W3C standards (namely, SMIL Timing and Synchronization and SMIL Timesheets).

With the raise of HTML5 which natively supports continuous content (audio, video) there is a dramatic need for handling synchronization, animation and user interaction in an efficient and homogeneous way. As web browsers do not support SMIL, except for SVG Animation (which rely on the SMIL BasicAnimation module), multimedia web authoring remains difficult and relies on code-based, non-standard solutions.

Therefore we are developing a generic, cross-browser JavaScript implementation for scheduling the dynamic behavior of HTML5 content that can be described with declarative SMIL markup. Using a declarative language makes sense for the most common tasks, that currently require JavaScript programming:

- it is much easier for web authors and for web authoring tool developers;
- it is a much better way to achieve good accessibility and indexability;
- it is easier to maintain, since no specific JavaScript code is used.

5.4. Mobile Audio Language

Participants: Yohan Lasorsa, Jacques Lemordant.

5.4.1. MAUDL library

The MAUDL library (Mobile AUDio Language) [15] is an evolution of the ARIA library whose primary target was games on mobile.

Augmented Reality Audio applications use sound objects to create a soundscape. A sound object is a time structure of audio chunks whose duration is on the time scale of 100 ms to several seconds. These sound objects have heterogeneous and time-varying properties. In order to describe Interactive Audio (IA) contents, we created MAUDL, an XML language inspired by iXMF that is well adapted to the design of dynamic soundtracks for navigation systems.

MAUDL prevents audio information overwhelming through categorization at the declarative level and the use of priority queues at the execution level. This allows to take account of speed when walking, and rapid hand gestures when interrogating the environment for example. MAUDL can be used as an authoring time interchange file format for interactive mobile applications or as a runtime file format that is actually loaded through the web and played directly in the mobile. MAUDL is a cue-oriented interactive audio system, audio services being requested using named events and the systems response to each event being determined by the audio artist. The current version of the API supports iOS and further support for other mobile platforms (Android) is planned.

5.4.2. 3D Audio Pointer

A virtual 3D audio pointer provides an intuitive guide to the user of a mobile application, reducing the need for cognitive work when compared to vocal instructions. We have built such a pointer using the MAUDL language. It gives the user the azimuth using HRTF spatialized audio cues, with additional hints taking the form of variations in the sound used. It allows to superpose other kinds of audio contents, such as voice while the pointer is active, to indicate distance for example. This audio object is suitable for different sorts of navigation systems, such as POIs browsers, self-guided audio tours, or predefined route following applications.

5.5. Mixed Reality Browser

Participants: Audrey Colbrant, Yohan Lasorsa, Jacques Lemordant, David Liodenot, Mathieu Razafimahazo.

The Mixed Reality Browser (MRB) is a geolocalized web browser running on mobile devices.
The concept of Mixed Reality comes from the fact that the real/virtual dichotomy is not sharp, but interpo-
latively smooth over a virtuality continuum. Idealized notions of reality and virtuality can be thought of as
endpoints on a continuum, an instance of the former approach corresponding for example to a see-through
display with natural sounds, an instance of the latter to texture-mapped image-based rendering (panoramas)
with synthetic sound objects.

Augmented Reality (AR) mode refers to all cases in which the auditory or visual display of an otherwise real
environment is augmented by means of virtual sound or graphic objects. The converse case on the virtuality
continuum is Augmented Virtuality (AV), where a virtual world, one that is generated primarily by computer,
like with synthetic 3D graphic or synthetic panoramic, is being augmented with the audio-visual content of
points of interest (POIs).

The introduction of mobile augmented reality browsers has forced a rethink on what kind of reality should
be offered. Mobility induces a need for telepresence and simulation to free the user or the developer of
the necessity to go every time in the real world. Mobility is the main reason behind the concept of Mixed
Reality Browsers. By its intrinsic characteristics, MRB supports advance MR applications like mobile remote
maintenance and assisted navigation.
5. Software

5.1. SPArse Modeling Software (SPAMS)
SPAMS v2.1 was released as open-source software in June 2011 (v1.0 was released in September 2009 and v2.0 in November 2010). It is an optimization toolbox implementing algorithms to address various machine learning and signal processing problems involving

- Dictionary learning and matrix factorization (NMF, sparse PCA, ...)
- Solving sparse decomposition problems with LARS, coordinate descent, OMP, SOMP, proximal methods
- Solving structured sparse decomposition problems ($\ell_1/\ell_2$, $\ell_1/\ell_\infty$, sparse group lasso, tree-structured regularization, structured sparsity with overlapping groups, ...).

The software and its documentation are available at http://www.di.ens.fr/willow/SPAMS/.

5.2. Non-uniform Deblurring for Shaken and Partially Saturated Images
This is a package of Matlab code for non-blind removal of non-uniform camera shake blur from a single blurry image. The package explicitly deals with images containing some saturated pixels. The algorithm is described in [19]. The package is publicly available at http://www.di.ens.fr/willow/research/saturation/.

5.3. Local dense and sparse space-time features
This is a package with Linux binaries implementing extraction of local space-time features in video. The package was updated in January 2011. The code supports feature extraction at Harris3D points, on a dense space-time grid as well as at user-supplied space-time locations. The package is publicly available at http://www.di.ens.fr/~laptev/download/stip-2.0-linux.zip.

5.4. Segmenting Scenes by Matching Image Composites
This is a package of Matlab code implementing unsupervised data-driven scene segmentation as described in (Russell et al. NIPS 2009). The package was created in June 2011 and is available at http://www.cs.washington.edu/homes/bcr/projects/SceneComposites/index.html.

5.5. Discriminative Clustering for Image Co-segmentation
This is a package of Matlab code implementing unsupervised discriminative clustering for co-segmenting multiple images described in (Joulin et al. CVPR 2010) and (Joulin et al. NIPS 2010). The aim is to segment a given set of images containing objects from the same category, simultaneously and without prior information. The package was last updated in October 2011 and is available at http://www.di.ens.fr/~joulin/code/coseg.zip.

5.6. Clustering with Convex Fusion Penalties
This is a package of Matlab code implementing a hierarchical clustering with convex fusion penalties described in (Hocking et al. ICML 2011 [10]). The package is available at http://www.di.ens.fr/~joulin/code/clusterpath_norm_Inf.zip.
5. Software

5.1. WebSmatch (Web Schema Matching)

Participants: Zohra Bellahsène, Emmanuel Castanier, Rémi Coletta, Duy Hoa Ngo, Patrick Valduriez [contact].

URL: http://websmatch.gforge.inria.fr/

In the context of the Action de Développement Technologique (ADT) started in October 2010, WebSmatch is a flexible, open environment for discovering and matching complex schemas from many heterogeneous data sources over the Web. It provides three basic functions: (1) metadata extraction from data sources; (2) schema matching (both 2-way and n-way schema matching), (3) schema clustering to group similar schemas together. WebSmatch is being delivered through Web services, to be used directly by data integrators or other tools, with RIA clients. Implemented in Java, delivered as Open Source Software (under LGPL) and protected by a deposit at APP (Agence de Protection des Programmes). WebSmatch is being used by Datapublica and CIRAD to integrate public data sources.

5.2. SON (Shared-data Overlay Network)

Participants: Ayoub Ait Lahcen, Fady Draidi, Esther Pacitti, Didier Parigot [contact], Patrick Valduriez, Guillaume Verger.

URL: http://www-sop.inria.fr/teams/zenith/SON

SON is an open source development platform for P2P networks using web services, JXTA and OSGi. SON combines three powerful paradigms: components, SOA and P2P. Components communicate by asynchronous message passing to provide weak coupling between system entities. To scale up and ease deployment, we rely on a decentralized organization based on a DHT for publishing and discovering services or data. In terms of communication, the infrastructure is based on JXTA virtual communication pipes, a technology that has been extensively used within the Grid community. Using SON, the development of a P2P application is done through the design and implementation of a set of components. Each component includes a technical code that provides the component services and a code component that provides the component logic (in Java). The complex aspects of asynchronous distributed programming (technical code) are separated from code components and automatically generated from an abstract description of services (provided or required) for each component by the component generator.

5.3. P2Prec (P2P recommendation service)

Participants: Fady Draidi, Esther Pacitti [contact], Didier Parigot, Guillaume Verger.

URL: http://p2prec.gforge.inria.fr

P2Prec is recommendation service for P2P content sharing systems that exploits users social data. To manage users social data, we rely on Friend-Of-A-Friend (FOAF) descriptions. P2Prec has a hybrid P2P architecture to work on top of any P2P content sharing system. It combines efficient DHT indexing to manage the users FOAF files with gossip robustness to disseminate the topics of expertise between friends. P2Prec is implemented in Java using the Data-Shared Overlay Network (SON) infrastructure which is the basis for the ANR DataRing project.

5.4. ProbDB (Probabilistic Database)

Participants: Reza Akbarinia [contact], Patrick Valduriez, Guillaume Verger.

URL: http://probdb.gforge.inria.fr
ProbDB is a probabilistic data management system to manage uncertain data on top of relational DBMSs. One of the main features of the prototype is its portability; that means with a minimum effort it can be implemented over any DBMS. In ProbDB, we take advantage of the functionalities provided by almost all DBMSs, particularly the query processing functions. It is implemented in Java on top of PostgreSQL.

5.5. SnoopIm

Participants: Julien Champ [contact], Alexis Joly.

SnoopIm is a content-based search engine allowing to retrieve small visual patterns or objects in large collections of pictures (such as logos on clothes, road signs in the background, paintings on walls, etc.) and to derive statistics from them (frequency, visual cover, size variations, etc.). Query objects to be searched can be either selected from the collection of photos or from an external picture (by simply providing its URL). The web application allows online search of multiple users and has a cache feature to speed-up the processing of seen queries. It is implemented in Javascript on top of a C++ library developed in collaboration with INA'sup (http://www.ina-sup.com/).

5.6. SimJoin (Distributed Approximate Similarity Join)

Participant: Alexis Joly [contact].

SimJoin is a distributed software for the efficient computation of the full approximate k-nn graph of large collections of high-dimensional features. It is developed within a MapReduce framework and is therefore easily portable to large cloud computing plateform. It is based on recent theoretic contributions related to locality preserving hash functions [34]. Its first main feature is to allow splitting a large collection of high-dimensional features into highly balanced pages that preserve locality according to any given similarity kernel. Its second main feature is to build in $O(n^{1+\gamma})$ operations a candidate set of item pairs that approximate the theoretic knn-graph with high recall. This software is developed in collaboration with INRIA Imedia.