AUDIO, SPEECH, AND LANGUAGE PROCESSING
1. ALPAGE Project-Team ................................................................. 5
2. METISS Project-Team ............................................................... 10
3. PAROLE Project-Team .............................................................. 12
4. Sémagramme Team ................................................................. 16
5. TALARIS Project-Team .............................................................. 19

INTERACTION AND VISUALIZATION
6. ALICE Project-Team ................................................................. 22
7. ARTIS Project-Team ................................................................. 24
8. AVIZ Project-Team ................................................................. 28
9. EVASION Project-Team ......................................................... 32
10. IN-SITU Project-Team ............................................................ 35
11. IPARLA Project-Team ............................................................ 44
12. MIMETIC Team ................................................................. 46
13. MINT Team ................................................................. 48
14. REVES Project-Team ........................................................... 50
15. VR4I Team ................................................................. 52

KNOWLEDGE AND DATA REPRESENTATION AND MANAGEMENT
16. AXIS Project-Team ................................................................. 54
17. DAHU Project-Team (section vide) ........................................ 58
18. DREAM Project-Team ........................................................... 59
19. EDELWEISS Project-Team .................................................. 61
20. EXMO Project-Team ............................................................ 65
21. GRAPHIK Project-Team ....................................................... 67
22. LEO Team ................................................................. 68
23. MAIA Project-Team .............................................................. 71
24. MOSTRARE Project-Team ................................................... 72
25. ORPAILLEUR Project-Team .................................................. 73
26. SMIS Project-Team ............................................................ 77
27. WAM Project-Team ............................................................. 78
28. ZENITH Team ................................................................. 81

ROBOTICS
29. AROBAS Project-Team ........................................................ 83
30. COPRIN Project-Team ........................................................ 85
31. E-MOTION Project-Team ...................................................... 87
32. FLOWERS Project-Team ..................................................... 92
33. IMARA Project-Team ......................................................... 109
34. LAGADIC Project-Team ....................................................... 111

VISION, PERCEPTION AND MULTIMEDIA UNDERSTANDING
35. ARIANA Project-Team ........................................................ 115
36. IMEDIA Project-Team ......................................................... 116
<table>
<thead>
<tr>
<th></th>
<th>Project-Team</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>37.</td>
<td>LEAR Project-Team</td>
<td>117</td>
</tr>
<tr>
<td>38.</td>
<td>MAGRIT Project-Team</td>
<td>118</td>
</tr>
<tr>
<td>39.</td>
<td>Morphee Team</td>
<td>119</td>
</tr>
<tr>
<td>40.</td>
<td>PERCEPTION Project-Team</td>
<td>121</td>
</tr>
<tr>
<td>41.</td>
<td>PRIMA Project-Team</td>
<td>122</td>
</tr>
<tr>
<td>42.</td>
<td>PULSAR Project-Team</td>
<td>127</td>
</tr>
<tr>
<td>43.</td>
<td>TEMICS Project-Team</td>
<td>132</td>
</tr>
<tr>
<td>44.</td>
<td>TExMEx Project-Team</td>
<td>135</td>
</tr>
<tr>
<td>45.</td>
<td>WILLOW Project-Team</td>
<td>140</td>
</tr>
</tbody>
</table>
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 intel (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 interleaver 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 (mysq1, 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 Vallée), 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/](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](http://alpage.inria.fr/statgram/frdep/fr_stat_dep_parsing.html)

Alpahage 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.


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, SXLFG);
- 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/alexima/.

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 Lefff 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 Leffel), 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](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 Guérin 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.
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 the 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 footage 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.
PAROLE Project-Team

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 Smaïli (smaili@loria.fr).

We published in 2007 a method which allows to align sub-titles comparable corpora [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 ².

5.3. SELORIA

contact : Odile Mella (Odile.Mella@loria.fr).

SELORIA is a toolbox for speaker diarization.

² 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 facilities.
• 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 alignment 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-alignment 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. 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 a counting 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 saturate 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.

1Available 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.
TALARIS Project-Team

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 and hand, 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://talc.loria.fr:8081/EmoDial](http://talc.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. Graphite

Participants: Phuon 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.
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 Novelitis, 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 4! 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 (i.e. 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/OBWBTOS8
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 crowded 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.
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 [2] 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.

Figure 3. Screenshot the Diffamation system during a transition: (a) the document view, (b) the overview scrollbar and (c) the timeline.

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/](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](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](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/.
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. An original journal paper must be submitted soon. MyCF browser is now available on line: [http://www.mycorporisfabrica.org/](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.](image)

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.

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 subclass 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

5.1. jBricks

Participants: Stéphane Huot, Emmanuel Pietriga [correspondant], 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 x 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 x 43 200 pixels) augmented with country borders ESRI shapefiles. (b) Panning and zooming in Spitzer’s Infrared Milky Way (396 032 x 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 [correspondant], 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 (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.

---

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).

![Image of a numeric text field and a semi-transparent menu](image)

*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](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).

Touchstone is available at http://code.google.com/p/touchstone-platforms/ under a BSD License.

- 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 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
Figure 7. VCR-like interface, session overview and sample plots of mouse trajectories (black) and mouse clicks (red)

- License: GPL
- Type of human computer interaction: Deamon and Graphique
- OS/Middleware: X Window (deamon) and Java (VCR interface)
- Required library or software: all X libraries (daemon) and Java (VCR interface)
- Programming language: * C and Java

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 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. In particular 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. 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...). Theses 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.
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
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 Photoshop, 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. Imerase: Inria Multi-Environment Realistic Simulation Engine

Participants: Adrien David, George Drettakis.

In the context of the ADT Interact3D and the ARC NIEVE, we developed Imerase, a middleware to be used as a VR engine, helping in the implementation of realistic simulations for immersive installations. Imerase 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 Imerase 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. OpenMASK: Open-Source platform for Virtual Reality

**Participants:** Alain Chauffaut [contact], Ronan Gaugne [contact], Georges Dumont, Thierry Duval, Laurent Aguerreche, 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 frequential 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
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 Télis 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-corrrespondant], 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 CCClust [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, CCClust. 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 registrated 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

**Participant:** 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 [correspondant], 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** method have been developed in Java:

- **SMDS** compares the sequences to each other 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 Rafaele 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/ 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/ .
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
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.
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 KnP 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

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.

![Figure 1. ISICIL Platform Architecture](image)

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 SIOC\(^3\), SCOT, and Newman’s Tag Ontology\(^4\) 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](http://sioc-project.org)
4 [http://www.holygoat.co.uk/owl/redwood/0.1/tags](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 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
EXMO Project-Team

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 [ ]. 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 owns.

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. Cogui

**Participants:** Alain Gutierrez, Michel Chein, Michel Leclère, Marie-Laure Mugnier, Madalina Croitoru.

*Cogui* ([http://www.lirmm.fr/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 URIrefs 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. AlignViz

Name: AlignViz
Contact: Fayçal Hamdi (hamdi@lri.fr)
Other contacts: Brigitte Safar (safar@lri.fr) and Chantal Reynaud (chantal.reynaud@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), Fatiha 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-adaptative 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)
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

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
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.
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/.

- 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/.

- 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.

See also the web page http://www.grappa.univ-lille3.fr/~torre/Recherche/Softwares/volata/.

- 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\(^1\). 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\(^2\) 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/~jfmar/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 Diplômé 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. 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 ubiqui-
tuous 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 per-
sonal 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.

WAM Project-Team

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 worldwide 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 Layaida.

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 interpolatively 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. 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 platform. 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.
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
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.
4. Software

4.1. PROTEUS

Participants: Amaury Nègre, Juan Lahera-Perez.

This toolkit offers an automatic mobile robot driver, some sensor 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.

- Version: 2.0
- APP: IDDN.FR.001.510040.000.S.P.2005.000.10000
- Programming language: C+++, Lua

4.2. AROSDYN

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
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.

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. 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 mumur2), 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/](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.
Figure 2. An example of an experiment in RLPark. Zephyr displays two views of a learned weight vector, an animation of the problem, the current policy distribution learned by the algorithm and the reward obtained by the algorithm.

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 Nonnegative 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.

![ProcBox2D Simulation](image)

**Figure 3.** A JBox2D simulation rendered with Processing using ProcBox2D. A robotic arm is interacting with 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 one, 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 begun 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 (formally 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 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 software is 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 units. 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 is 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**

This UObject wraps the API of the librhoban (see the previous chapter). This tool is especially developed to control Bioloid motors in high frequency. With that software, we can create instances of motors scanned and directly read and write features like position, torque, load, speed.

---

1Broadly used servo motor product from Robotis [http://www.robotis.com](http://www.robotis.com)
2Electronic 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, Mathieu 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.

![Stem, Behaviour, Scene components](image)

**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 stylised interaction. Repeated interactions lead to the development of a common repertoire of words for naming objects. More precisely, object 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 hole 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.

![Figure 8. A virtual visitor interact with a virtual grid](image)

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 freenect\(^3\) 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.

---

\(^3\)Kinect 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 [correspondant], 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.

### 5.5.2. Flowers Field/Robot Lamps

**Participants:** Jérôme BÉCHU [correspondant], Pierre-Yves OUDEYER, 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.

![Figure 14. Flowers Fields/Robot Lamps](image)

### 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 versions 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. 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
LAGADIC Project-Team

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.

![Figure 2. Lagadic robotics platforms for vision-based manipulation](image)

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 Hippocrate 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
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.
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/cbir-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. 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.
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
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.

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/ 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 over the world including the 4D repository website hosted by INRIA Grenoble http://4drepository.inrialpes.fr/ . The software was developed in the context of the European project iGlance, it is available as an open-source software under the GNU LGPL Licence.

5.3. Databases

5.3.1. 4D repository (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. 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 [25]). 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, Loïc 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 preparation 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
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 auto-stereoscopic 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].
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, Qualcom 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/.

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/.

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/.

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 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*

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) implementations 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:

1. **I-DESCRIPTION** (APP deposit number: IDDN.FR.001.270047.000.S.P.2003.000.21000),

**A-SARES**, is a symbolic machine learning system that automatically infers, from descriptions of pairs of linguistic 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),

**ANAMORPHO**, 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](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.
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 (ℓ_1/ℓ_2, ℓ_1/ℓ_∞, sparse group lasso, tree-structured regularization, structured sparsity with overlapping groups,...).


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/](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](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](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](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](http://www.di.ens.fr/~joulin/code/clusterpath_norm_Inf.zip).