Activity Report 2011

Section Scientific Foundations
**Algorithmics, Programming, Software and Architecture**

1. ATEAMS Project-Team ......................................................... 4  
2. DART Project-Team .......................................................... 8

**Applied Mathematics, Computation and Simulation**

3. DOLPHIN Project-Team ...................................................... 28  
4. MODAL Team ................................................................. 34  
5. NON-A Team ................................................................. 35  
6. SEQUEL Project-Team ....................................................... 42  
7. SIMPAF Project-Team ....................................................... 49

**Computational Sciences for Biology, Medicine and the Environment**

8. BONSAI Project-Team ....................................................... 57  
9. S.H.A.M.A.N Team .......................................................... 58

**Networks, Systems and Services, Distributed Computing**

10. ADAM Project-Team ....................................................... 62  
11. POPS Project-Team ......................................................... 65  
12. RMOD Project-Team ....................................................... 67

**Perception, Cognition, Interaction**

13. MINT Team ................................................................. 72  
14. MOSTRARE Project-Team ................................................. 74
3. Scientific Foundations

3.1. Research method

We are inspired by formal methods and logic to construct new tools for software analysis, transformation and generation. We try and proof the correctness of new algorithms using any means necessary.

Nevertheless we mainly focus on the study of existing (large) software artifacts to validate the effectiveness of new tools. We apply the scientific method. To (in)validate our hypothesis we often use detailed manual source code analysis, or we use software metrics, and we have started to use more human subjects (programmers).

Note that we maintain ties with the CWI spinoff “Software Improvement Group” which services most of the Dutch software industry and government and many European companies as well. This provides access to software systems and information about software systems that is valuable in our research.

3.2. Software analysis

This research focuses on source code; to analyze it and to transform it. Each analysis or transformation begins with fact extraction. After the we may analyze specific software systems or large bodies of software systems. Our goal is to improve software systems by learning to understand the causes of complexity.

The mother and father of fact extraction techniques are probably Lex, a scanner generator, and AWK, a language intended for fact extraction from textual records and report generation. Lex is intended to read a file character-by-character and produce output when certain regular expressions (for identifiers, floating point constants, keywords) are recognized. AWK reads its input line-by-line and regular expression matches are applied to each line to extract facts. User-defined actions (in particular print statements) can be associated with each successful match. This approach based on regular expressions is in wide use for solving many problems such as data collection, data mining, fact extraction, consistency checking, and system administration. This same approach is used in languages like Perl, Python, and Ruby. Murphy and Notkin have specialized the AWK-approach for the domain of fact extraction from source code. The key idea is to extend the expressivity of regular expressions by adding context information, in such a way that, for instance, the begin and end of a procedure declaration can be recognized. This approach has, for instance, been used for call graph extraction but becomes cumbersome when more complex context information has to be taken into account such as scope information, variable qualification, or nested language constructs. This suggests using grammar-based approaches as will be pursued in the proposed project. Another line of research is the explicit instrumentation of existing compilers with fact extraction capabilities. Examples are: the GNU C compiler GCC, the CPPX C++ compiler, and the Columbus C/C++ analysis framework. The Rigi system provides several fixed fact extractors for a number of languages. The extracted facts are represented as tuples (see below). The CodeSurfer source code analysis tool extracts a standard collection of facts that can be further analyzed with built-in tools or user-defined programs written in Scheme. In all these cases the programming language as well as the set of extracted facts are fixed thus limiting the range of problems that can be solved.

The approach we want to explore is the use of syntax-related program patterns for fact extraction. An early proposal for such a pattern-based approach is described in: a fixed base language (either C or PL/1 variant) is extended with pattern matching primitives. In our own previous work on RScript we have already proposed a query algebra to express direct queries on the syntax tree. It also allows the querying of information that is attached to the syntax tree via annotations. A unifying view is to consider the syntax tree itself as “facts” and to represent it as a relation. This idea is already quite old. For instance, Linton proposes to represent all syntactic as well as semantic aspects of a program as relations and to use SQL to query them. Due to the lack of expressiveness of SQL (notably the lack of transitive closures) and the performance problems encountered, this approach has not seen wider use.
Another approach is proposed by de Moor and colleagues and uses path expressions on the syntax tree to extract program facts and formulate queries on them. This approach builds on the work of Paige and attempts to solve a classic problem: how to incrementally update extracted program facts (relations) after the application of a program transformation.

Parsing is a fundamental tool for fact extraction for source code. Our group has longstanding contributions in the field of Generalized LR parsing and Scannerless parsing. Such generalized parsing techniques enable generation of parsers for a wide range of real (legacy) programming languages, which is highly relevant for experimental research and validation.

3.2.1. Goals

The main goal is to replace labour-intensive manual programming of fact extractors by automatic generation from annotated grammars or other concise and formal notation. There is a wide open scientific challenge here: to create a uniform and generic framework for fact extraction that is superior to current more ad-hoc approaches. We expect to develop new ideas and techniques for generic (language-parametric) fact extraction from source code and other software artifacts.

Given the advances made in fact extraction we are starting to apply our techniques to observe source code and analyze it in detail.

3.3. Relational paradigm

For any source code analysis or transformation, after fact extraction comes elaboration, aggregation or other further analyses of these facts. For fact analysis, we base our entire research on the simple formal concept of a "relation".

There are at least three lines of research that have explored the use of relations. First, in SQL, n-ary relations are used as basic data type and queries can be formulated to operate on them. SQL is widely used in database applications and a vast literature on query optimization is available. There are, however, some problems with SQL in the applications we envisage: (a) Representing facts about programs requires storing program fragments (e.g., tree-structured data) and that is not easy given the limited built-in datatypes of SQL; (b) SQL does not provide transitive closures, which are essential for computing many forms of derived information; (c) More generally, SQL does not provide fixed-point computations that help to solve sets of equations. Second, in Prolog, Horn clauses can be used to represent relational facts and inference rules for deriving new facts. Although the basic paradigm of Prolog is purely declarative, actual Prolog implementations add imperative features that increase the efficiency of Prolog programs but hide the declarative nature of the language. Extensions of Prolog with recursion have resulted in Datalog in many variations [AHV95]. In F(p)–L a Prolog database and a special-purpose language are used to represent and query program facts.

Third, in SETL, the basic data type was the set. Since relations can easily be represented as sets of tuples, relation-based computations can, in principle, be expressed in SETL. However, SETL as a language was very complicated and has not survived. However, work on programming with sets, bags and lists has continued well into the 90’s and has found a renewed interested with the revival of Lisp dialects in 2008 and 2009.

We have already mentioned the relational program representation by Linton. In Rigi, a tuple format (RSF) is introduced to represent untyped relations and a language (RCL) to manipulate them. Relational algebra is used in GROK, Crocopat and Relation Partition Algebra (RPA) to represent basic facts about software systems and to query them. In GUPRO graphs are used to represent programs and to query them. Relations have also been proposed for software manufacture, software knowledge management, and program slicing. Sometimes, set constraints are used for program analysis and type inference. More recently, we have carried out promising experiments in which the relational approach is applied to problems in software analysis and feature analysis. Typed relations can be used to decouple extraction, analysis and visualization of source code artifacts. These experiments confirm the relevance and viability of the relational approach to software analysis, and also indicate a certain urgency of the research direction of this team.
3.3.1. Goals

- New ideas and techniques for the efficient implementation of a relation-based specification formalism.
- Design and prototype implementation of a relation-based specification language that supports the use of extracted facts (Rascal).
- We target at uniform reformulations of existing techniques and algorithms for software analysis as well as the development of new techniques using the relational paradigm.
- We apply the above in the reformulation of refactorings for Java and domain specific languages.

3.4. Refactoring and Transformation

The final goal, to be able to safely refactor or transform source code can be realized in strong collaboration with extraction and analysis.

Software refactoring is usually understood as changing software with the purpose of increasing its readability and maintainability rather than changing its external behavior. Refactoring is an essential tool in all agile software engineering methodologies. Refactoring is usually supported by an interactive refactoring tool and consists of the following steps:

- Select a code fragment to refactor.
- Select a refactoring to apply to it.
- Optionally, provide extra parameter needed by the refactoring (e.g., a new name in a renaming).

The refactoring tool will now test whether the preconditions for the refactoring are satisfied. Note that this requires fact extraction from the source code. If this fails the user is informed. The refactoring tool shows the effects of the refactoring before effectuating them. This gives the user the opportunity to disable the refactoring in specific cases. The refactoring tool applies the refactoring for all enabled cases. Note that this implies a transformation of the source code. Some refactorings can be applied to any programming language (e.g., rename) and others are language specific (e.g., Pull Up Method). At http://www.refactoring.com an extensive list of refactorings can be found.

There is hardly any general and pragmatic theory for refactoring, since each refactoring requires different static analysis techniques to be able to check the preconditions. Full blown semantic specification of programming languages have turned out to be infeasible, let alone easily adaptable to small changes in language semantics. On the other hand, each refactoring is an instance of the extract, analyze and transform paradigm. Software transformation regards more general changes such as adding functionality and improving non-functional properties like performance and reliability. It also includes transformation from/to the same language (source-to-source translation) and transformation between different languages (conversion, translation). The underlying techniques for refactoring and transformation are mostly the same. We base our source code transformation techniques on the classical concept of term rewriting, or aspects thereof. It offers simple but powerful pattern matching and pattern construction features (list matching, AC Matching), and type-safe heterogenous data-structure traversal methods that are certainly applicable for source code transformation.

3.4.1. Goals

Our goal is to integrate the techniques from program transformation completely with relational queries. Refactoring and transformation form the Achilles Heel of any effort to change and improve software. Our innovation is in the strict language-parametric approach that may yield a library of generic analyses and transformations that can be reused across a wide range of programming and application languages. The challenge is to make this approach scale to large bodies of source code and rapid response times for precondition checking.
3.5. The Rascal Meta-programming language

The Rascal Domain Specific Language for Source code analysis and Transformation is developed by ATeams. It is a language specifically designed for any kind of meta programming.

Meta programming is a large and diverse area both conceptually and technologically. There are plentiful libraries, tools and languages available but integrated facilities that combine both source code analysis and source code transformation are scarce. Both domains depend on a wide range of concepts such as grammars and parsing, abstract syntax trees, pattern matching, generalized tree traversal, constraint solving, type inference, high fidelity transformations, slicing, abstract interpretation, model checking, and abstract state machines. Examples of tools that implement some of these concepts are ANTLR, ASF+SDF, CodeSurfer, Crocopat, DMS, Grok, Stratego, TOM and TXL. These tools either specialize in analysis or in transformation, but not in both. As a result, combinations of analysis and transformation tools are used to get the job done. For instance, ASF+SDF relies on RScript for querying and TXL interfaces with databases or query tools. In other approaches, analysis and transformation are implemented from scratch, as done in the Eclipse JDT. The TOM tool adds transformation primitives to Java, such that libraries for analysis can be used directly. In either approach, the job of integrating analysis with transformation has to be done over and over again for each application and this requires a significant investment.

We propose a more radical solution by completely merging the set of concepts for analysis and transformation of source code into a single language called Rascal. This language covers the range of applications from pure analyses to pure transformations and everything in between. Our contribution does not consist of new concepts or language features per se, but rather the careful collaboration, integration and cross-fertilization of existing concepts and language features.

3.5.1. Goals

The goals of Rascal are: (a) to remove the cognitive and computational overhead of integrating analysis and transformation tools, (b) to provide a safe and interactive environment for constructing and experimenting with large and complicated source code analyses and transformations such as, for instance, needed for refactorings, and (c) to be easily understandable by a large group of computer programming experts. Rascal is not limited to one particular object programming language, but is generically applicable. Reusable, language specific, functionality is realized as libraries.
3. Scientific Foundations

3.1. Introduction

The main research topic of the DaRT team-project concerns the hardware/software codesign of embedded systems with high performance processing units like DSP or SIMD processors. A special focus is put on multiprocessor architectures on a single chip (System-on-Chip). The contribution of DaRT is organized around the following items:

Co-modeling for High Performance SoC design: We define our own metamodels to specify application, architecture, and (software hardware) association. These metamodels present new characteristics as high level data parallel constructions, iterative dependency expression, data flow and control flow mixing, hierarchical and repetitive application and architecture models. All these metamodels are implemented with respect to the MARTE standard profile of the OMG group, which is dedicated to the modeling of embedded and real-time systems.

Model-based optimization and compilation techniques: We develop automatic transformations of data parallel constructions. They are used to map and to schedule an application on a particular architecture. This architecture is by nature heterogeneous and appropriate techniques used in the high performance community can be adapted. We developed new heuristics to minimize the power consumption. This new objective implies to specify multi criteria optimization techniques to achieve the mapping and the scheduling.

SoC simulation, verification and synthesis: We develop a SystemC based simulation environment at different abstraction levels for accurate performance estimation and for fast simulation. To address an architecture and the applications mapped on it, we simulate in SystemC at different abstraction levels the result of the SoC design. This simulation allows us to verify the adequacy of the mapping and the schedule, e.g., communication delay, load balancing, memory allocation. We also support IP (Intellectual Property) integration with different levels of specification. On the other hand, we use formal verification techniques in order to ensure the correctness of designed systems by particularly considering the synchronous approach. Finally, we transform MARTE models of data intensive algorithms in VHDL, in order to synthesize a hardware implementation.

3.2. Co-modeling for HP-SoC design

The main research objective is to build a set of metamodels (application, hardware architecture, association, deployment and platform specific metamodels) to support a design flow for SoC design. We use a MDE (Model Driven Engineering) based approach.

3.2.1. Foundations

3.2.1.1. System-on-Chip Design

SoC (System-on-Chip) can be considered as a particular case of embedded systems. SoC design covers a lot of different viewpoints including the application modeling by the aggregation of functional components, the assembly of existing physical components, the verification and the simulation of the modeled system, and the synthesis of a complete end-product integrated into a single chip.

The model driven engineering is appropriate to deal with the multiple abstraction levels. Indeed, a model allows several viewpoints on information defined only once and the links or transformation rules between the abstraction levels permit the re-use of the concepts for a different purpose.
3.2.1.2. Model-driven engineering

Model Driven Engineering (MDE) \cite{121} is now recognized as a good approach for dealing with System on Chip design issues such as the quick evolution of the architectures or always growing complexity. MDE relies on the model paradigm where a model represents an abstract view of the reality. The abstraction mechanism avoids dealing with details and eases reusability.

A common MDE development process is to start from a high level of abstraction and to go to a targeted model by flowing through intermediate levels of abstraction. Usually, high level models contain only domain specific concepts, while technological concepts are introduced smoothly in the intermediate levels. The targeted levels are used for different purposes: code generation, simulation, verification, or as inputs to produce other models, etc. The clear separation between the high level models and the technological models makes it easy to switch to a new technology while re-using the previous high level designs. Transformations allow to go from one model at a given abstraction level to another model at another level, and to keep the different models synchronized.

In an MDE approach, a SoC designer can use the same language to design application and architecture. Indeed, MDE is based on proved standards: UML 2 \cite{65} for modeling, the MOF (Meta Object Facilities \cite{110}) for metamodel expression and QVT \cite{111} for transformation specifications. Some profiles, i.e. UML extensions, have been defined in order to express the specificities of a particular domain. In the context of embedded system, the MARTE profile in which we contribute follows the OMG standardization process.

3.2.1.3. Models of computation

We briefly present our reference models of computation that consist of the Array-OL language and the synchronous model. The former allows us to express the parallelism in applications while the latter favors the formal validation of the design.

Array-OL. The Array-OL language \cite{90}, \cite{91}, \cite{83}, \cite{81} is a mixed graphical-textual specification language dedicated to express multidimensional intensive signal processing applications. It focuses on expressing all the potential parallelism in the applications by providing concepts to express data-parallel access in multidimensional arrays by regular tilings. It is a single assignment first-order functional language whose data structures are multidimensional arrays with potentially cyclic access.

The synchronous model. The synchronous approach \cite{79} proposes formal concepts that favor the trusted design of embedded real-time systems. Its basic assumption is that computation and communication are instantaneous (referred to as “synchrony hypothesis”). The execution of a system is seen through the chronology and simultaneity of observed events. This is a main difference from visions where the system execution is rather considered under its chronometric aspect (i.e., duration has a significant role). There are different synchronous languages with strong mathematical foundations. These languages are associated with tool-sets that have been successfully used in several critical domains, e.g. avionics, nuclear power plants.

In the context of the DaRT project, we consider declarative languages such as Lustre \cite{85} and Signal \cite{104} to model various refinements of Array-OL descriptions in order to deal with the control aspect as well as the temporal aspect present in target applications. The first aspect is typically addressed by using concepts such as mode automata, which are proposed as an extension mechanism in synchronous declarative languages. The second aspect is studied by considering temporal projections of array dimensions in synchronous languages based on clock notion. The resulting synchronous models are analyzable using the formal techniques and tools provided by the synchronous technology.

3.2.2. Contributions of the team

Our proposal is partially based upon the concepts of the “Y-chart” \cite{97}. The MDE contributes to express the model transformations which correspond to successive refinements between the abstraction levels. Metamodeling brings a set of tools which enable us to specify our application and hardware architecture models using UML tools, to reuse functional and physical IPs, to ensure refinements between abstraction levels via mapping rules, to initiate interoperability between the different abstraction levels used in a same codesign, and to ensure the opening to other tools, like verification tools, thought the use of standards.
The application and the hardware architecture are modeled separately using similar concepts inspired by Array-OL to express the parallelism. The placement and scheduling of the application on the hardware architecture is then expressed in an association model.

All the previously defined models, application, architecture and association, are platform independent and they conform to the MARTE OMG Profil (figure 1). No component is associated with an execution, simulation or synthesis technology. Such an association targets a given technology (OpenMP, OpenCL, SystemC/PA, VHDL, etc.). Once all the components are associated with some IPs of the GasparLib library, the deployment is fully realized. This result can be transformed to further abstraction level models via some model transformations (figure 2).

The simulation results can lead to a refinement of the initial application, hardware architecture, association and deployment models. We propose a methodology to work with all these different models. The design steps are:

1. Separation of application and hardware architecture modeling.
2. Association with semi-automatic mapping and scheduling.
3. Selection of IPs from libraries for each element of application/architecture models, to achieve the deployment.
4. Automatic generation of the various platform specific simulation or execution models.
5. Automatic simulation or execution code generation with calls to the IPs.
6. Refinement at the highest level taking account of the simulation results.

3.2.2.1. High-level modeling in Gaspard2

In Gaspard2, models are described by using the recent OMG standard MARTE profile combined with a few native UML concepts and some extensions.

The new release of Gaspard2 uses different packages of MARTE for UML modeling. The Hardware Resource Model (HRM) concepts of MARTE enable to describe the hardware part of a system. The Repetitive Structure Modeling (RSM) concepts allow one to describe repetitive structures (DaRT team was the main contributor of this MARTE package definition). Finally, the Generic Component Modeling (GCM) concepts are used as the base for component modeling.

The above concepts are expressive enough to permit the modeling of different aspects of an embedded system:

- functionality (or applicative part): the focus is mainly put on the expression of data dependencies between components in order to describe an algorithm. Here, the manipulated data are mainly multidimensional arrays. Furthermore, a form of reactive control can be described in modeled applications via the notion of execution modes. This last aspect is modeled with the help of some native UML notions in addition to MARTE.

- hardware architecture: similar mechanisms are also used here to describe regular architectures in a compact way. Regular parallel computation units are more and more present in embedded systems, especially in SoCs. HRM is fully used to model these concepts. Some extensions are proposed for NoC design and FPGA specifications. The GPU have a particular memory hierarchy. In order to model the memory details, we extend the MARTE metamodel to describe low level characteristics of the memory.

- association of functionality with hardware architecture: the main issues concern the allocation of the applicative part of a system onto the available computation resources, and the scheduling. Here also, the allocation model takes advantage of the repetitive and hierarchical representation offered by MARTE to enable the association at different granularity levels, in a factorized way.
In addition to the above usual design aspects, Gaspard2 also defines a notion of deployment specification (see Figure 1) in order to select compilable IPs from libraries, at this time models can produce codes. The corresponding package defines concepts that (i) enable to describe the relation between a MARTE representation of an elementary component (a box with ports) to a text-based code (and Intellectual Property - IP, or a function with arguments), and (ii) allow one to inform the Gaspard2 transformations of specific behaviors of each component (such as average execution time, power consumption...) in order to generate a high abstraction level simulation in adequacy with the real system. Recently this package was extended to design reconfigurable systems using dynamical deployment.

![Figure 1. Overview of the design concepts.](image)

### 3.2.2.2. Intermediate concept modeling and transformations

Gaspard2 targets different technologies for various purposes: formal verification, high-performance computing, simulation and hardware synthesis (Figure 1). This is achieved via model transformations that relate intermediate representations towards the final target representations.

- A metamodel for procedural language with OpenMP (OpenMP in Figure 1). It is inspired by the ANSI C and Fortran grammars and extended by OpenMP statements [68]. The aim of this metamodel is to use the same model to represent Fortran and C code. Thus, from an OpenMP model, it is possible to generate OpenMP/Fortran or OpenMP/C. The generated code includes parallelism directives and control loops to distribute task (IPs code) repetitions over processors [124].

- A VHDL metamodel (VHDL in Figure 1). It gathers the necessary concepts to describe hardware accelerators at the RTL (Register Transfer Level) level, which allows the hardware execution of applications. This metamodel introduces, e.g., the notions of clock and register in order to manipulate some of the usual hardware design concepts. It is precise enough to enable the generation of synthetizable HDL code [103].
The two metamodels SystemC and Pthread was redefined to implement both a multi-thread execution model. These are described in the "New results" part.

- Synchronous metamodel (Synchronous Equational). It was used to benefit of the verification tools of synchronous languages. It is not yet maintained in the new release of Gaspard2.

The transformation scheme. In order to target these metamodels, several transformations have been developed (Figure 2). MartePortInstance introduces into the MARTE metamodel the concept of PortInstance corresponding to an instance of port associated to a part. The ExplicitAllocation transformation explicits the association of each application part on the processing units, according to the association of other elements in the application hierarchy. The LinkTopologyTask transformation replaces the connectors between a component and an inner repeated part by a task managing the data (TilerTask). The scheduling of the application tasks is decomposed into three transformations, Synchronisation that associates, to each application component, a local graph of tasks corresponding to its parts; GlobalSynchronization that computes a global graph of tasks for the complete application from the local graphs of tasks; and Scheduling that schedules the tasks from the global graph. TilerMapping maps the TilerTasks onto processors. The management of the data in the memory is performed through two transformations. MemoryMapping maps the data into memory i.e. creates the variables and allocates address spaces. AddressComputation computes addresses for each variable. Finally, some transformations are dedicated to targets: Functional introduces the concepts relative to procedural languages. pThread transforms MARTE elementary tasks into threads and the connectors into buffers. SystemC traduces the MARTE architecture into concepts of the SystemC language.

3.2.2.3. An operational semantics for RSM

The Repetitive Structure Modeling (RSM) package of the UML MARTE profile is used to describe repetitive computations and topologies (e.g., data-parallel algorithms, grid of processing units) in an embedded system. In Gaspard2, the concepts provided by this package are of prime importance for the specification of data-intensive applications. A formal semantics [82] has been previously defined for the Array-OL language, which is the basis for the definition of RSM. We proposed an new formal semantics for RSM, which is operational unlike [82]. Execution semantic descriptions are rarely taken into account in the definition of UML profiles. This raises several serious correctness issues about the manipulation of models defined with these profiles. The aim of our new semantics [100] is to answer this demand by proposing a help for understanding the behavior and execution of models specified with RSM concepts in UML MARTE.

3.2.2.4. Clock-based modeling of embedded system behavior

The concepts defined in the RSM package of MARTE allow one to suitably describe the data intensive algorithms [70][69]. In order to add more details about the system functional behavior, logical clocks are associated with components to describe the expected rates at which data should be processed. The Time sub-profile of MARTE is used to model this rate information. It offers a rich expressivity for describing both logical and physical time aspects [74]. The rate constraints are expressed using the CCSL package of MARTE in the form of clock constraints. We refer to this clock constraints as functional clock properties.

The physical resources that implement the data intensive algorithms are specified in MARTE. For each resource, hardware IPs are deployed in order to refine the models towards a specific technology. At this level, we extract information concerning the processors speed represented by its frequency. We synthesize new clocks that represent the periods of the clock cycles for each processor involved in the execution. All clocks are related to an ideal clock. The occurrence of the instants of the ideal clock are fast enough to capture any instant of the processors clocks. We refer to these clock specifications as physical clock properties.

Since application functionality and hardware architecture are modeled independently in Gaspard2, the allocation phase bridges these two different views in order to map functionality on their associated physical resources. In terms of clocks, this allocation is expressed as the mapping of functional clock properties onto physical clock properties, according to a particular mapping algorithm. The result of such allocation is a new set of clocks reflecting the simulation of the temporal behavior of the system during execution. We refer to these clock description as simulation clock properties. They are usable for a very relevant system analysis.
Figure 2. Overview of the transformation chains.
3.2.2.5. High-level modeling and exploration of non functional properties

We have proposed an approach for high-level modeling and exploration of non functional properties. Our work proposed a Model Driven Engineering (MDE)-based approach to integrate non functional requirements for systems on chip and defined metamodels that allow the integration of external optimization tools in the Gaspard2 environment. The designer creates the application and architecture models at a high level. The designer should then take the decision to allocate application functions on hardware components. This decision depends essentially on the non functional properties of both of the software and hardware components. For this reason, it is necessary to express these requirements. The proposed methodology uses models enriched with non-functional properties to drive the optimization of resource allocation.

3.2.2.6. HPF towards Marte

Concerning the power of expression of the MARTE RSM subprofile that we have defined, we have studied the data and computation distribution capabilities. We have proved that the MARTE «distribute» stereotype is at least as expressive as the well known High Performance Fortran data distribution. The proof is constructive: starting from an ALIGN and a DISTRIBUTE HPF directive, we build a MARTE «distribute».

3.2.2.7. MARTE extensions for reconfigurable based systems

Reconfigurable FPGA based Systems-on-Chip (SoC) architectures are increasingly becoming the preferred solution for implementing modern embedded systems. However due to the tremendous amount of hardware resources available in these systems, new design methodologies and tools are required to reduce their design complexity.

In previous work, we provided an initial contribution to the modeling of these systems by extending MARTE profile to incorporate significant design criteria such as power consumption.

In its current version, MARTE lacks dynamic reconfiguration concepts. Even these later are necessary to model and implement rapid prototypes for complex systems.

Our objective is to define all necessary concepts for dynamic reconfiguration issues regarding configuration latency, resources number, etc. Afterwards, these concepts will be integrated to MARTE to obtain an extended and complete profile, which can be called Reconfigurable MARTE (RecoMARTE).

Our current proposals permit us to model fine grain reconfigurable FPGA architectures with an initial extension of the MARTE profile to model Dynamic Reconfiguration at a high-level description.

Since a controller is essential for managing a dynamically reconfigurable region, we modeled a state machine at high abstraction levels using UML state machine diagrams. This state machine is responsible for switching between the available configurations.

As a future work, we will analyze the reconfigurable design flow of Xilinx from the design partitioning to the bitstream generation stage. It is a starting point for understanding how to generate configuration files. Then, we will extract relevant data to define our own design flow.

3.2.2.8. Traceability

We use the transformation mechanism to assist a tester in the mutation analysis process dedicated to model transformations. The mutation analysis aims to qualify a test model set. More precisely, errors are voluntary injected in transformation and the ability of the test models set to highlight these errors is analyzed. If the number of highlighted errors, i.e. if the test model set is not enough qualified, new models have to be added in order to raise the set quality [108]. Our approach relies on the hypothesis that it is easier to modify an existing model than to create a new one from scratch. The local trace, coupled to a mutation matrix, helps the tester to identify adequate test models and their relevant parts to modify in order to improve the test data set.

We propose a semi-automation approach that can automatically generate new test model in some cases and efficiently assist the testers in others cases [77].
3.2.2.9. Transformation migration after metamodel evolution

Metamodels evolve because of several reasons such as design refinement and software requirement changes. When this happens, transformations defined in terms of those metamodels might become inconsistent and migration would be necessary. Due to the lack of methodology support, transformation migration is mostly ad hoc and manually performed. Besides, the growing complexity and size of transformations make this task difficult and error prone. We started works in this domain area. More specifically, on the one hand, we specify transformation consistency by defining the relationship between transformation and metamodels, we called it domain conformance. On the other hand, we propose a transformation migration process which describes the set of tasks that should be completed in order to re-establish consistency after metamodel evolution [107], [116].

3.2.2.10. Model transformation towards SystemC-PA

The buffered strategy developed for the transformation chain towards pThread has been kept to simulate the behavior of the application for the SystemC-PA simulation target. Mapped tasks are associated to threads themselves run on SystemC processing modules. Most of the thread contents (concepts, transformation and code generator) were reused and coupled with the SystemC contents dedicated to the architecture. A new model transformation has been developed to map the threads related to the application to the SystemC elements related to the architecture. The data accesses in the new SystemC-PA target are triggered off when the buffers (Pthread mechanisms) are requested. Those accesses are forwarded to the architecture through the TLM2 communication channels of the processors running the thread. The resulting transformation chain is available in the on-line Gaspard version (http://www.gaspard2.org).

3.2.3. Gaspard2 for avionic hybrid test platform design

The emergence and the maturity of FPGA circuits for distributed and reconfigurable architectures offer the opportunity to explore real time problems in the field of avionic systems. FPGA becomes de facto a major processing element as same as general CPUs. As of now, the FPGA is widely used in the field of I/O component in order to connect the real equipment with the CPU host. Among the main features mapped into the FPGA in the original architecture, we quote the fast serial link and RAM IPs (Intellectual property) which are needed to ensure communication between CPU and FPGA. Additionally, the Base Time IP is needed for the global system synchronization. This minimal configuration based on FPGA can be duplicated several times and connected together to build bigger test system or a complete simulator. Eurocopter expectation for the above-described architecture is to prototype some models which can be eligible and relocated in the FPGA. The objective is to increase the performances of these models and to reduce the communication latencies by the means of embedding the different parts in the same chip. To do so, we studied in this first year a real avionic test loop in order to extract the complex models that will be implemented in the FPGA. Different hardware model configurations have been explored to reach an optimal well-balanced global system using the ML403 Virtex-4 Xilinx board. Different tradeoffs in terms of performance and resource occupation in the FPGA are obtained. Later, these results will be used for dynamically adapt the system functioning according to the available resources and performance requirements.

As a second part, we used the MARTE profile to represent an hybrid system (CPU/FPGA). In the MARTE specification, an application is a set of tasks connected through ports. Tasks are considered as mathematical functions reading data from their input ports and writing data on their output ports. This specification has been used to model the avionic test loop. In addition, MARTE allows describing the hardware architecture in a structural way. Typical components such as HwProcessor, HwFPGA and HwRAM can be specified with their non-functional properties. We used this subset of MARTE in order to represent an hybrid multiprocessor architecture. The main component of this architecture is composed of the Xeon-X3370 processor (multicore CPU) and the Virtex-4 Xilinx FPGA. Furthermore, MARTE provides the Allocate concept as well as the concept specially crafted for repetitive structures Distribute. This latter concept gives a way to express regular distribution of tasks onto a set of processors or FPGA resources. The mapping step relies on two types of distribution (timeScheduling and spatialDistribution) depending on the target hardware platform (CPU/FPGA). The different models of our avionic test loop can be mapped onto the host multicore processor, the embedded processor (Microblaze) or the hardware resources in the FPGA.
3.3. Model-based optimization and compilation techniques

3.3.1. Foundations

3.3.1.1. Optimization for parallelism

We study optimization techniques to produce “good” schedules and mappings of a given application onto a hardware SoC architecture. These heuristic techniques aim at fulfilling the requirements of the application, whether they be real time, memory usage or power consumption constraints. These techniques are thus multi-objective and target heterogeneous architectures.

We aim at taking advantage of the parallelism (both data-parallelism and task parallelism) expressed in the application models in order to build efficient heuristics.

Our application model has some good properties that can be exploited by the compiler: it expresses all the potential parallelism of the application, it is an expression of data dependencies—so no dependence analysis is needed—, it is in a single assignment form and unifies the temporal and spatial dimensions of the arrays. This gives to the optimizing compiler all the information it needs and in a readily usable form.

3.3.1.2. Transformation and traceability

Model to model transformations are at the heart of the MDE approach. Anyone wishing to use MDE in its projects is sooner or later facing the question: how to perform the model transformations? The standardization process of Query View Transformation [111] was the opportunity for the development of transformation engines as Viatra, MoFln or Sitra. However, since the standard has been published, only few of investigating tools, such as ATL\(^1\) (a transformation dedicated tool) or Kermeta\(^2\) (a generalist tool with facilities to manipulate models) are powerful enough to execute large and complex transformations such as in the Gaspard2 framework. None of these engine is fully compliant with the QVT standard. To solve this issue, new engine relying on a subset of the standard recently emerged such as QVTO\(^3\) and smartQVT. These engines implement the QVT Operational language.

Traceability may be used for different purposes such as understanding, capturing, tracking and verification on software artifacts during the development life cycle [98]. MDE has as main principle that everything is a model, so trace information is mainly stored as models. Solutions are proposed to keep the trace information in the initials models source or target [125]. The major drawbacks of this solution are that it pollutes the models with additional information and it requires adaptation of the metamodels in order to take into account traceability. Using a separate trace model with a specific semantics has the advantage of keeping trace information independent of initial models [102].

3.3.2. Contributions of the team

3.3.2.1. Data-parallel code transformations

We have studied Array-OL to Array-OL code transformations [83], [122], [93], [92], [94], [101]. Array-OL allows a powerful expression of the data access patterns in such applications and a complete parallelism expression. It is at the heart of our metamodel of application, hardware architecture and association.

The code transformations that have been proposed are related to loop fusion, loop distribution or tiling but they take into account the particularities of the application domain such as the presence of modulo operators to deal with cyclic frequency domains or cyclic space dimensions (as hydrophones around a submarine for example).

We pursue the study of such transformations with two objectives:

- Propose utilization strategies of such transformations in order to optimize some criteria such as memory usage, minimization of redundant computations or adaptation to a target hardware architecture.
- Stretch their application domain to our more general application model (instead of just Array-OL).

---

\(^1\) [http://www.eclipse.org/m2m/atl]
\(^2\) [http://www.kermeta.org]
\(^3\) [http://www.eclipse.org/m2m/qvto/doc]
In 2009 the study on the interaction between the high-level data-parallel transformations and the inter-repetition dependencies (allowing the specification of uniform dependencies) was achieved. Because the ODT formalism behind the Array-OL transformations cannot express dependencies between the elements of the same multidimensional space, in order to take into account the uniform dependencies we proposed and proved an algorithm that, starting from the hierarchical distribution of repetition before and after a transformation, is capable to compute the new uniform dependencies that express the same exact dependencies as before the transformations. It all comes down to solving an (in)equations system, interpreting the solutions and translating them into new uniform dependencies.

The algorithm was implemented and integrated into the refactoring toolbox and enables the use of the transformations on models containing inter-repetition dependencies.

In order to validate the theoretical work around the high-level Array-OL refactoring based on the data-parallel transformations, together with Eric Lenormand and Michel Barreteau from THALES Research & Technology we worked on a study on optimization techniques in the context of an industrial radar application. We have proposed a strategy to use the refactoring toolbox to help explore the design space, illustrated on the radar application modeled using the Modeling and Analysis of Real-time and Embedded systems (MARTE) UML profile.

3.3.2.2. Multi-objective hierarchical scheduling heuristics

When dealing with complex heterogeneous hardware architectures, the scheduling heuristics usually take a task dependence graph as input. Both our application and hardware architecture models are hierarchical and allow repetitive expressions. We propose a Globally Irregular, Locally Regular (GILR) combination of heuristics to allow to take advantage of both task and data parallelism [105] and have started evaluating multi-objective evolutionary meta-heuristics in this context. These evolutionary meta-heuristics deal with the irregular (task parallelism) part of the design [80] while we have proposed a heuristic to deal with the regular part (data parallelism) [106].

Furthermore, local optimizations (contained inside a hierarchical level) decrease the communication overhead and allow for a more efficient usage of the memory hierarchy. We aim at combining the data-parallel code transformations presented before and the GILR heuristics in order to deal efficiently with the data-parallelism of the application by using repetitive parts of the hardware architecture.

The introduction of uniform inter-repetition dependencies in the data-parallel tasks of Gaspard2 has had several consequences. Aside the modification of the refactoring (see section 3.3.2.1), we have studied the compilation of such tasks. This compilation involves the scheduling of such repetitions on repetitive grids of processors and the code generation. This scheduling problem is NP-complete and we have proposed a heuristic based on the automatic parallelization techniques to compute a good (efficient both in time and code size) schedule in the case when all loop bounds and processor array shapes are known.

3.3.2.3. Transformation techniques

In the previous version of Gaspard2, model transformations were complex and monolithic. They were thus hardly evolvable, reusable and maintainable. We thus proposed to decompose complex transformations into smaller ones jointly working in order to build a single output model [96]. These transformations involve different parts of the same input metamodel (e.g. the MARTE metamodel); their application field is localized. The localization of the transformation was ensured by the definition of the intermediary metamodels as delta. The delta metamodel only contains the few concepts involved in the transformation (i.e. modified, or read). The specification of the transformations only uses the concepts of these deltas. We defined the Extend operator to build the complete metamodel from the delta and transposed the corresponding transformations. The complete metamodel corresponds to the merge between the delta and the MARTE metamodel or an intermediary metamodel. The transformation then becomes the chaining of metamodel shifts and the localized transformation. This way to define the model transformations has been used in the Gaspard2 environment. It allowed a better modularity and thus also reusability between the various chains.
3.3.2.4. Traceability

Our traceability solution relies on two models the Local and the Global Trace metamodels. The former is used to capture the traces between the inputs and the outputs of one transformation. The Global Trace metamodel is used to link Local Traces according to the transformation chain. The local trace also proposes an alternative “view” to the common traceability mechanism that does not refers to the execution trace of the transformation engine. It can be used whatever the used transformation language and can easily complete an existing traceability mechanism by providing a more finer grain traceability [75].

Furthermore, based on our trace metamodels, we developed algorithms to ease the model transformation debug. Based on the trace, the localization of an error is eased by reducing the search field to the sequence of the transformation rule calls [76].

3.3.2.5. Verifying conformance and semantics-preserving model transformations

We give formal executable semantics to the notions of conformance and of semantics-preserving model transformations in the model-driven engineering framework [119]. Our approach consists in translating models and meta-models (possibly enriched with OCL invariants) into specifications in Membership Equational Logic, an expressive logic implemented in the Maude tool. Conformance between a model and a meta-model is represented by the validity of a certain theory interpretation, of the specification representing the meta-model, in the specification representing the model. Model transformations between origin and destination meta-models are mappings between the sets of models that conform to the those meta-models, respectively, and can be represented by rewrite rules in Rewriting Logic, a superset of Membership Equational Logic also implemented in Maude. When the meta-models involved in a transformation are endowed with dynamic semantics, the transformations between them are also typically required to preserve those semantical aspects. We propose to represent the notion of dynamic semantics preservation by means of algebraic simulations expressed in Membership Equational Logic. Maude can then be used for automatically verifying conformance, and for automatically verifying dynamic semantics preservation up to a bounded steps of the dynamic semantics. These works lead to better understood meta-models and models, and to model transformations containing fewer errors.

3.3.2.6. Modeling for GPU

The model described in UML with Marte profile model is chained in several inout transformations that adds and/or transforms elements in the model. For adding memory allocation concepts to the model, a QVT transformation based on «Memory Allocation Metamodel» provides information to facilitate and optimize the code generation. Then a model to text transformation allows to generate the C code for GPU architecture. Before the standard releases, Acceleo is appropriate to get many aspects from the application and architecture model and transform it in CUDA (cu, .cpp, .c, .h, Makefile) and OpenCL (.cl, .cpp, .c, .h, Makefile) files. For the code generation, it’s required to take into account intrinsic characteristics of the GPUs like data distribution, contiguous memory allocation, kernels and host programs, blocks of threads, barriers and atomic functions.

3.3.2.7. Clock-based design space exploration for SoCs

We have previously proposed an abstract clock-based modeling of data-intensive SoCs behaviors within the Gaspard2 framework [70] [69]. Both application functionality and hardware architecture are characterized in terms of clocks. Then, their allocation is also expressed as a projection of functional clock properties onto physical clock properties, according to a mapping choice. The result of such allocation is a new set of clocks reflecting the simulation of the temporal behavior of the system during execution.

This year, this approach has been applied to the design of the H.264 encoder on a multiprocessor hardware architecture using the standard MARTE profile [71]. The obtained model has been analyzed by considering abstract clocks. In particular, it has been shown that such clocks help to tackle design space exploration issues via a relevant modeling of different hardware/software mappings. The trade-off about processor frequency scaling, system functional properties and energy consumption has been addressed, via different hardware IP choices. This has been achieved via a qualitative reasoning on traces resulting from a scheduling of logical clocks, capturing functional properties, on physical clocks derived from processors frequency.
3.3.2.8. Optimized code generation from UML/MARTE models

Starting from the observation that some semantics (and thus some optimization possibilities) are lost when generating code in a programming language from a UML/MARTE model, the contribution of a thesis co-directed with the CEA LIST is an optimization at the model level followed by a translation to the GENERIC intermediate representation of the gcc compilation framework in order to allow more optimization, for the moment focusing on code size optimization.

3.3.2.9. Architecture exploration based on meta-heuristics

Some progress has been made on the proposal of meta-heuristics use for multi-objective mapping and scheduling. In collaboration with the Dolphin project-team of INRIA Lille - Nord Europe and LIFL we have modeled the association process of Gaspard2 as an optimization problem in order to solve it with a genetic algorithm based heuristic that has been implemented in the ParadisEO optimization framework. This new heuristics is currently being integrated in the Gaspard2 tool. Another work comparing heuristics based on the particle swarm and genetic algorithm meta heuristics has been proposed in collaboration with the computer science laboratory of Oran, Algeria, in continuation of our collaboration.

3.3.2.10. Architecture exploration for efficient data transfer and storage

A major point in embedded system design today is the optimization of communication structures, memory hierarchy and global synchronizations. Such an optimization is a time consuming and error-prone process, that requires a suitable automatic approach. We proposed an electronic system level framework to explore the data transfer storage micro-architecture and the synchronization of iterative data-parallel applications [ 88 ]. The aim is to define a methodology that can be a front-end for loop-based high level synthesis or interconnect hardware IPs in order to realize memory-centric MPSoCs. In Gaspard2, this will enable to assess various mappings of Array-OL models onto different kinds of target architectures.

Our solution starts from a canonical Array-OL representation and apply a set of transformations in order to infer an Application Specific architecture that masks the times to transfer data with the time to perform the computations. A customizable model of the target architecture including FIFO queues and double buffering mechanism is proposed. The mapping of a given image processing application onto this architecture is performed through a flow of Array-OL transformations aimed to improve the parallelism level and to reduce the size of the used internal memories. A method based on an integer partition is considered to reduce the space of explored transformations.

3.3.2.11. Multi-objective mapping and scheduling heuristics

Mohamed Akli Redjedal, univ. Lille 1 master, co-directed with Laetitia Jourdan form the Dolphin project-team of INRIA Lille - Nord Europe and LIFL. The work of Mohamed Redjedal has consisted in modeling the association process of Gaspard2 as an optimization problem in order to solve it with a genetic algorithm based heuristic. He has indeed modeled this multi-objective mapping and scheduling problem, proposed a heuristic and its implementation in the ParadisEO optimization framework. A 1st year master student from the univ. of Brussels has worked 6 weeks on the model driven export from Gaspard2 to the optimization heuristics proposed by Mohamed Redjedal

3.3.2.12. GPGPU code production

The solution of large, sparse systems of linear equations « Ax=b » presents a bottleneck in sequential code executing on CPU. To solve a system bound to Maxwell’s equations on Finite Element Method (FEM), a version of conjugate gradient iterative method was implemented in CUDA and OpenCL as well. The aim is to accelerate and verify the parallel code on GPUs. The first results showed a speedup around 6 times against sequential code on CPU. Another approach uses an algorithm that explores the sparse matrix storage format (by rows and by columns). This one did not increase the speedup but it allows to evaluate the impact of the access to the memory.
3.3.2.13. From MARTE to OpenCL.

We have proposed an MDE approach to generate OpenCL code. From an abstract model defined using UML/MARTE, we generate a compilable OpenCL code and then, a functional executable application. As MDE approach, the research results provide, additionally, a tool for project reuse and fast development for not necessarily experts. This approach is an effective operational code generator for the newly released OpenCL standard. Further, although experimental examples use mono device(one GPU) example, this approach provides resources to model applications running on multi devices (homogeneously configured). Moreover, we provide two main contributions for modeling with UML profile to MARTE. On the one hand, an approach to model distributed memory simple aspects, i.e. communication and memory allocations. On the other hand, an approach for modeling the platform and execution models of OpenCL. During the development of the transformation chain, an hybrid metamodel was proposed for specifying of CPU and GPU programming models. This allows generating other target languages that conform the same memory, platform and execution models of OpenCL, such as CUDA language. Based on other created model to text templates, future works will exploit this multi language aspect. Additionally, intelligent transformations can determine optimization levels in data communication and data access. Several studies show that these optimizations increase remarkably the application performance.

3.3.2.14. Formal techniques for construction, compilation and analysis of domain-specific languages

The increasing complexity of software development requires rigorously defined domain specific modelling languages (DSML). Model-driven engineering (MDE) allows users to define their language’s syntax in terms of metamodels. Several approaches for defining operational semantics of DSML have also been proposed [123], [89], [73], [84], [115]. We have also proposed one such approach, based on representing models and metamodels as algebraic specifications, and operational semantics as rewrite rules over those specifications [95], [120]. These approaches allow, in principle, for model execution and for formal analyses of the DSML. However, most of the time, the executions/analyses are performed via transformations to other languages: code generation, resp. translation to the input language of a model checker. The consequence is that the results (e.g., a program crash log, or a counterexample returned by a model checker) may not be straightforward to interpret by the users of a DSML. We have proposed in [118] a formal and operational framework for tracing such results back to the original DSML’s syntax and operational semantics, and have illustrated it on SPEM, a language for timed process management.

3.3.3. Electromagnetic modeling

The Finite Integration Technique (F.I.T) is used to compute the phenomena. This technique is efficient if the mesh is generated by a regular hexahedron. Moreover the matrix system, obtained from a regular mesh can be exploited to use the parallel direct solver. In fact, in reordering the unknowns by the nested dissection method, it is possible to construct directly the lower triangular matrix with many processors without assembling the matrix system. During this year, we have used our parallel direct solver as a preconditionner for a sparse linear system coming from a FEM problem with a good efficiency.

3.4. HP-SoC simulation, verification and synthesis

Many simulations at different levels of abstraction are the key of an efficient design of embedded systems. The different levels include a functional (and possibly distributed) validation of the application, a functional validation of the application and an architecture co-model, and a validation of a heterogeneous specification of an embedded system (a specification integrating modules provided at different abstraction levels).

SoCs are more and more complex and integrate software parts as well as specific hardware parts (IPs, Intellectual Properties). Generally before obtaining a SoC on silicium, a system is specified at several abstraction levels. Any system design flow consists in refining, more or less automatically, each model to obtain another, starting from a functional model to reach a Register Transfer Level model. One of the biggest design challenges is the development of a strong, low cost and fast simulation tool for system verification and simulation.
The DaRT project is concerned by the simulation at different levels of abstraction (SystemC, VHDL) of the application/architecture co-model and of the mapping/schedule produced by the optimization phase.

3.4.1. Foundations

3.4.1.1. Abstraction levels and Transaction Level Modeling

Currently, Transaction Level Modeling, TLM, is being used in the industry to solve a variety of practical problems during the design, development and deployment of electronic systems.

The TLM 2.0 standard appeared during the very few last years. It consists in describing systems according to the specifications of the TLM abstraction levels. At these levels, function calls simulate the behavior of the communications between architecture components.

Nowadays, this modeling style is widely used for verification and it is starting to be used for design at many major electronic companies. Recently, many actions and challenges have been started in order to help to proliferate TLM. Thus, several teams are working to furnish to designers standard TLM APIs and guidelines, TLM platform IP and tools supports. SystemC is the first system description language adopting TLM specifications. Thus, several standardization APIs have been proposed to the OSCI by all the major EDA and IP vendors. This standardization effort is being generalized now by the OSCI / OCP-IP TLM standardization alliance, to build on a common TLM API foundation. One of the most important TLM API proposals is the one from Cadence, distributed to OSCI and OCP-IP. It is intended as common foundation for OSCI and OCP-IP allowing protocol-specific APIs (e.g. AMBA, OCP) and describing a wide range of abstraction levels for fast and efficient simulations.

In order to keep our design flow coherent, we choose to use two significant simulation levels. Each of them has special advantages.

The main objectives of the PVT level are fast verification of system functionalities and monitoring of the contentions in the interconnection network. Complementary to this level, the CABA level is used to accurately estimate the execution time and power consumption. At the PVT level, details related to the computation and communication resources are omitted. The software application is executed by an instruction-accurate Instruction Set Simulator. Transactions are performed through channels instead of signals. At the CABA level, hardware components are implemented at the cycle accurate level for both processing and communication parts. Communication protocol and arbitration strategy are specified as well. Simulation at the PVT level permits a rapid exploration of a large solution space by eliminating non interesting regions from the DSE process. The solutions selected at this level are then forwarded to a new exploration at the CABA level. At each level, the exploration is based on developed performance and power estimation tools. Code generation at both of those levels needs parameter specifications for execution time, power estimation, and platform configurations. These parameters are specified at the deployment phase.

Due to all TLM’s benefits, we defined a TLM metamodel as a top level point for automatic transformations to both simulation and synthesis platforms. Our TLM metamodel contains the main concepts needed for verification and design following the Cadence API proposal. But, as we are targeting multi-language simulation platforms, the meta model is completely independent from the SystemC syntax. It is composed mainly by two parts: architecture and application. This clear separation between SW and HW parts permits easy extensions and updates of the meta model.

- The architecture part contains all necessary concepts to describe HW elements of systems at TLM levels. The SW part is mainly composed of computation tasks. They should be hierarchical and repetitive. A set of parameters could be attached to each task in order to specify the scheduling dependently of the used computation model.
- Thus this metamodel keeps hierarchies and repetitions of both the application and the architecture. This permits to still benefit from the data parallelism as far as possible in the design (simulation and synthesis flow). In fact, the designer can choose to eliminate hierarchies when transforming the TLM model into a simulation model, and to keep it when transforming into a synthesis model.
3.4.1.2. Dynamic reconfiguration - FPGA

Current FPGAs support the notion of Partial Dynamic Reconfiguration which allows part of the FPGA to be reconfigured on the fly hence introducing the idea of virtual hardware. Partial Reconfiguration allows swapping of tasks (mutually exclusive) depending upon user requirements and Quality of service needs. Using such a technology permits to optimize energy consumption and the area in the system. It allows also to have very flexible systems, adaptable for large application classes.

3.4.1.3. Verification

Our privileged basis for verification is the reactive synchronous domain. Over the last two decades several formal verification technologies have been provided by a very active research community in this domain. Among the available tools, we can mention efficient compilers that act more than usual compilers in that they address more static analysis issues. There are also various model-checkers that use both symbolic representations and non symbolic ones. Some of these model-checkers offer facilities that go beyond verification by enabling the synthesis of (discrete) controllers. Finally, these synchronous technologies give the opportunity in some cases to perform a functional simulation of the described systems.

3.4.2. Contributions of the team

The results of DaRT simulation package concerns mainly the PVT and the CABA levels. We also propose techniques to interact with IPs specified at other level of abstraction (mainly RTL).

3.4.2.1. Co-simulation in SystemC

From the association model, the Gaspard2 environment is able to automatically produce SystemC simulation code. The MDE techniques offer the transformation of the association model to the SystemC model. During this transformation the data parallel components are unrolled and the data dependencies between elementary tasks become synchronization primitive calls relying on a buffered strategy.

The SoC architecture is produced from the architecture model coupled with a ready-to-use component library. A processing module in SystemC simulates the behavior of tasks mapped to a particular processor.

Other modules contain the data parallel structures and are able to answer to any read/write requests. The communications between tasks and between tasks and memories are simulated via communication modules in SystemC. These last modules produce interesting results concerning the simultaneous network conflicts and the capacity of this network for this application.

A transformation chain within Gaspard2 ensures the code generation from the input model. The produced simulation code is based on SystemC IPs assembling. These IPs are available in the Gaspard2 library in both TLM and CABA levels. They represent all the usual architecture components such as processors (ARM, MIPS, ..etc), memories, caches, buses, NoCs, etc.

3.4.2.2. Model transformation towards Pthreads

The strategy in previous version of the Gaspard2 framework imposed a global synchronization mechanism between all the tasks of the application. This mechanism does not allow one to reach an optimal execution. We have investigated a new strategy to overcome this problem, based on fine grain synchronizations between the different tasks of the modeled application. For this new strategy, we use the pthread API. Each task of the UML application model is transformed into a thread. The data exchanges between the tasks are ensured by a buffer-based strategy. The best compromise between the memory used and the performance can be reached by adjusting the size of each buffer. Moreover, we have developed this strategy to facilitate its use in simulation targets such as SystemC-PA. The transformation chain towards Pthreads enabled to optimize the global synchronization mechanism between all the tasks of the application provided by the previous version of Gaspard2.
3.4.2.3. Gaspardlib extensions

The chain towards SystemC code allows simulations at the TLM-PA level. Regarding the architecture design, the process acts as a connector between existing SystemC modules. They correspond to basic components such as memories, processors, caches. They are gathered in the Gaspardlib to be included or linked at the code compilation step. On one hand, both application and architecture IPs have been modeled using UML to easily drag and drop the available components inside the user’s model. On another hand, we aimed at providing the most flexible design for the SystemC architecture.

The GaspardLib allows a high interoperability for our SystemC components with any other SystemC architecture. Consequently, additional SystemC modules have been integrated to extend the Gaspardlib. They come from other free simulation environments: ReSP,SocLib,Unisim.

3.4.2.4. Partial and Dynamic Reconfiguration (PDR) implementations

Current Gaspard2 Model transformation chain to Register Transfer Level (RTL) allows to generate two key aspects of a partial dynamically reconfigurable system: namely the dynamically reconfigurable region and the code for the reconfiguration manager that carries out the switch between the different configurations of this dynamic region. For this, the MARTE metamodel has been extended to integrate concepts of UML state machines and collaborations, which help in creation of mode automata semantics at the high abstraction levels. Integration of these concepts in the extended MARTE metamodel helps in the respective model-to-model transformations.

Moreover, the high level application model has several building blocks: the elementary components, each associated to several available intellectual properties (IPs). The current deployment level has been also extended to integrate the notion of "configurations", which are unique global implementations of the application functionality, with each configuration comprised of different combinations of IPs related to the elementary components. Using a combination of the deployment level and the introduced control semantics, it is possible for a designer to change the configuration related to an application, resulting in different results such as consumed FPGA resources, reconfiguration times, etc. We incorporate two model-to-model transformations in our flow, first the UML2MARTE transformation, with integrated state machine and configuration concepts. This transformation results in an intermediate MARTE model, which is converted into an RTL model by the MARTE2RTL transformation. The application model is converted into several implementations of a dynamically reconfigurable hardware accelerator, along with the source code for the configuration switch.

Finally, the design flow has been validated in the construction of a dynamically reconfigurable delay estimation correlation module that is part of a complex anti-collision radar detection system in collaboration with IEMN Valenciennes. The simulation results from the different configurations correspond to an initial MATLAB result, validating the different configurations. Additionally change of IPs related to a key elementary component in the module resulted in different reconfiguration times proving methodology.

3.4.2.5. IP based configurable massively parallel processing SoC

A methodology and a tool chain to design and build IP-based configurable massively parallel architectures is proposed. The defined architecture is named mppSoC, massively parallel processing System on Chip. It is a SIMD architecture composed of a number of processor elements (the PEs) working in perfect synchronization. A small amount of local and private memory is attached to each PE. Every PE is potentially connected to its neighbors via a regular network. Furthermore, each PE is connected to an entry of mpNoC, a massively parallel Network on Chip that potentially connects each PE to one another, performing efficient irregular communications. All the system is controlled by an Array Controller Unit (ACU). Our objective is to propose then a methodology to produce FPGA implementations of the mppSoC architecture.

The whole mppSoC architecture with its various components is implemented following an IP based design methodology. An implementation on FPGA, ALTERA StratixII 2s180, is proposed as a proof of feasibility. The architecture consists of general IPs (processor IPs, memory IPs, etc.) and specific IPs supplied with the mppSoC system (control IPs, etc.). Specific IPs are used as a glue to build the architecture. General IPs present a defined interface which must be respected by the designer if it wants to produce its own IP. For this kind of IPs we provide a library to alleviate their design. The designed architecture is configurable and parametric.
In fact, to construct a mppSoC system, we assemble IPs to generate a FPGA configuration. The designer has to make different choices. He has to determine the different components in his architecture, for example if it contains an irregular communication network with a defined interconnection router or a neighborhood one or both. Since we propose a parametric architecture, he has to choose also some architectural parameters such as the number of PEs, the memory size and the topology of the neighborhood network if it exists. After fixing the architecture, the designer will choose then the basic IPs which will be used such as processor IP, interconnection network IP, etc. By this way, the user can choose the most appropriate mppSoC configuration satisfying his needs. To evaluate the proposed design methodology we have implemented different sized architectures with various configurations. We have also tested some examples of data parallel applications such as FIR, reduction, matrix multiplication, image rotation and 2D convolution. Through simulation results we can choose the most appropriate mppSoC configuration with the optimal performance metrics: execution time, FPGA resources and energy consumption. As a result we have proposed an IP based methodology for the construction of mppSoC system helping the designer to choose the best configuration for a given application. It is a first step towards a mppSoC architecture exploration.

Ongoing work aims at integrating the mppSoC in a real application such a video processing framework. Future work will aim at improving the proposed IP assembling methodology to construct mppSoC systems. Our ultimate goal is to provide a completely tool to generate a mppSoC configuration in order to help the designer in a semi-automatic architecture exploration for a given application.

3.4.2.6. Caches in MPSoCs

In Multi-Processor System-on-Chip (MPSoC) architectures using shared-memory, caches plays an important impact on performance and energy consumption levels.

When the executed application depicts a high degree of reference locality, caches may reduce the amount of shared-memory accesses and data transfers on the interconnection network. Hence, execution time and energy consumption can be greatly optimized. However, caches in MPSoC architectures put forward the data coherency problem. In this context, most of the existing solutions are based either on data invalidation or data update protocols. These protocols do not consider the change in the application behavior. This paper presents a new hybrid cache-coherency protocol that is able to dynamically adapt its functioning mode according to the application needs.

An original architecture which facilitates this protocol’s implementation in Network-On-Chip based MPSoC architectures has been proposed. Performances, in terms of speed up factor and energy reduction gain of the proposed protocol, have been evaluated using a Cycle Accurate Bit Accurate (CABA) simulation platform. Experimental results in comparison with other existing solutions show that this protocol may give significant reductions in execution time and energy consumption can be achieved.

3.4.2.7. Verification

Guaranteeing the correctness of systems is a highly important issue in the Gaspard2 design methodology. This is required at least for their validation. In order to provide the designer with the required means to cope with validation, we propose to bridge the gap between the Gaspard2 design approach and validation techniques for SoCs by using the synchronous approach and test-based techniques.

We have already defined a synchronous dataflow equational model of Gaspard2 specification concepts. The resulting model is then usable to address various correctness issues: causality analysis that enables to detect erroneous data dependencies (i.e., those which lead to cycles) in specifications, clock synchronizability analysis when such a system model is to be considered on a deployment platform, etc.

Starting from the simulation clock properties of an embedded system (as described previously), we start an analysis of the system behavior. On the one hand, we verify whether or not the functional clock constraints specified by the designer in the application specification are met during the system execution on considered physical resources. When these constraints are not met, the simulation clock traces can be used to reason and find the solutions to satisfy the constraints. For instance, this may amount to decrease the speed of processors that compute data very fast or to increase the speed of processors that compute data very slowly. The modification of the processors speed by increasing or decreasing the speed should always respect the
functional constraints imposed by the designer. It appears in the simulation clock traces by determining new physical clock properties from the suitable processor frequencies. Another example of solution may consist in delaying the first activation of a faster processor until an adequate time to begin the execution. Such an activation delay could be seen as minimizing the voltage/frequency. The team examples have highlighted some needs for a better numeric verification of synchronous programs, and we also work on the amelioration of precision of the Signal analysis.

3.4.2.8. System Level Power Modeling

Due to the ongoing nano-miniaturization in chip production, estimation of power consumption is becoming a critical metric in embedded system design. In current industrial and academic practices, power estimation using low-level CAD tools is still widely adopted. These low level tools are however inconvenient to manage the architecture of modern complex embedded systems. System level power estimation is considered a vital premise to cope with the critical design constraints. The keywords in our contribution are Hybridization and decorrelation between abstraction levels. The hybridization is applied here at 2 levels: granularity of activities used to develop the power models in one side and the level of the considered abstraction on the other side. If almost of studies focus on power estimation for a given abstraction level without overcoming the wall of speed/accuracy trade-off, the idea is to build up hybrid power estimation tool that gathers different abstraction levels of the system to grab the strict relevant data depending on the power estimation process step. Thus, designers build their systems by instantiating different hardware and software IPs (Intellectual Property) from existing libraries. The granularity of the used power models should be coherent with the design approach. In this work, we develop a hybrid system level power estimator for embedded systems. First, power models relying on Functional Level Power Analysis (FLPA) methodology is developed. Secondly, we forge the whole system into a fast simulation framework in order to obtain the system's power consumption data. The combination of the above parts yields to a relatively fast and accurate power estimation. Our experimental results, performed on explicit embedded platform, show that obtained power estimation are less than 1% of error when compared to the measurements realized on the real system. In our work, we further extend the usage of higher abstraction level to speed up the estimation with the help of multigranularity of input data and phase sampling of the application. At the end, the proposed power estimation is 21 times faster than the detailed simulation with a marginal error of 1.5%.

3.4.2.9. Energy consumption driven dynamic reconfigurable execution model

As a continuation of our work on energy consumption estimation for Systems on Chip (SoC) at the Cycle Accurate Level using SystemC simulation, the aim of our current work is to ensure the adaptivity of SoCs regarding changes at run time of some operating conditions such as consumption constraints. This adaptivity is based on the reconfigurability of the SoCs implemented on FPGAs. Here, the energy consumption estimation is not done during simulation anymore but during the execution of the application on the FPGA.

In order to be adaptive to runtime changes, the system architecture has to be changed accordingly. A possible change can be, for example, to change the parallelism degree or to change a processing algorithm in order to consume less energy. The decision of reconfiguring is taken after a negotiation between consumption monitors integrated in the system. This monitors are OCP- compliant, which allows them to be easily integrated and reused for different architectures thanks to the genericity and parametrability of this standard communication protocol.

Up to now, we have started implementing simple systems on FPGA supporting the dynamic reconfiguration taking the user inputs as a criterion of reconfiguration. We also implemented some interface adapters in order to facilitate the future integration of the OCP monitors in the system. As a future work, we intend to integrate the energy consumption as a criterion of reconfiguration using monitors. These monitors are supposed to take decisions of reconfiguration after negotiating between them. Therefore, we started by studying the negotiation used on software systems such as multi-agent systems. We will adapt this for our hardware architecture on FPGA.

3.4.2.10. Partial dynamic reconfiguration
Partial dynamic reconfiguration modeling [114], [113] permits to generate two key aspects of a partial dynamically reconfigurable system from high level modeled specifications: namely the dynamically reconfigurable region and the code for the reconfiguration manager that carries out the switch between the different configurations of this dynamic region. Once these aspects are generated using the model transformations, it is possible to use commercial simulation and synthesis tools to implement dynamic reconfiguration in state of the art FPGAs [114]. Currently the intermediate model transformation chain is being updated to make use of the newly introduced intermediate metamodels and model transformations developed by the DaRT team, in order to provide a uniform design flow. Similarly, optimizations related to RTL code generation using Acceleo are also continuing.

However, the MARTE compliant high level specifications lack the means to express architectural details at high abstraction levels. For this reason, an initial exploratory analysis was carried out in [86] that expands the MARTE hardware concepts to include aspects of reconfigurable architectures, and to introduce aspects such as power consumption at these high level models. These works can be described as an initial contribution to the ANR FAMOUS project.

Similarly, MARTE has recently introduced the notion of ‘configurations’, similar to those introduced in [114]. These concepts permit to express system configuration at the MARTE UML models, but lack guidelines and precise semantics. An overview of these concepts was presented in [112], which highlights some of the shortcomings of the present concepts and provides an alternative, as described in [114].

3.4.2.11. Network on Chip synthesis

The study of Networks on Chip (NoC) is a research field that primarily addresses the global communication in Systems-on-Chip (SoC). The selected topology and the routing algorithm play a prime role in the performance of NoC architectures. In order to handle the design complexity and meet the tight time-to-market constraints, it is important to automate most of these NoC design phases. The use of MARTE in modeling such architectures may provide designers asset of high level concepts to obtain compact and reusable models in a fast way.

Thus we defined a new methodology for modeling concepts of NoC based Architectures. It aims to improve the effectiveness of the MARTE standard by clarifying some notations and extending some definitions in the standard, in order to allow modeling complex NoC architectures.

3.4.2.12. IP based configurable massively parallel processing SoC

Our mppSoC project proposed a methodology and tool chain to design and build IP-based configurable massively parallel architectures. A mppSoc architecture is a SIMD architecture composed of a number of processor elements working in perfect synchronization, the PEs. Each PE is potentially connected to its neighbors via a regular network. Furthermore, each PE is connected to an entry of mpNoC, a massively parallel Network on Chip that performs efficient irregular communications. All the system is controlled by an Array Controller Unit, the ACU.

The mppSoc project aims at the design and implementation of a given mppSoC architecture to fit the requirements of a given application. The mppSoC architecture model is then configurable and parametrizable and our chain produces FPGA implementations of the mppSoC architecture.

Our last contributions define a model-driven based generation chain integrated in the Gaspard environment. A mppSoC UML model is defined using using the MARTE profile. From this model, our chain allows the generation of the corresponding mppSoC synthesizable VHDL code that can be directly simulated or prototyped on FPGA. Targeting the DE2-70 FPGA board, we have been able to validate some mppSoC configurations running signal processing applications [ref]. This last works conclude Mouna Baklouti PhD thesis [ref].

3.5. Formal Methods for General and Domain-Specific Languages

We are working on developing and applying formal methods to the definition, analysis, and transformation of languages. These languages include general ones like C, Domain-Specific ones (DSLs) such as Kermeta [109], Signal [99], and VHDL, and Domain-Specific modelling ones (DSMLs) such as xSPEM [78].
We use rewriting techniques embodied in the K [117] and Maude [87] semantical frameworks, abstract interpretation techniques, techniques inspired from program transformation and compilation, and refinement techniques. We often use Model-Driven Engineering (MDE) as a *lingua franca* and we believe it is a useful vessel for formal methods into software engineering practice. We fruitfully collaborate with colleagues within Inria (the Triskell team at Inria Rennes-Bretagne Atlantique and the Compsys team at Inria Grenoble Rhône-Alpes), with colleagues outside Inria (David Monniaux at Verimag, Grenoble), and with foreign colleagues (the K-framework team bi-localised in Iași, Romania and in Urbana Champaign, USA; the university of Aleppo, Syria). We organise events (two workshops and one summer school in 2011), supervise PhD students (one started in the Fall 2001, co-supervision with the K team) and interns, participate in PhD committees (two in 2011) and in teaching. We have obtained financial support outside Inria from the University of Lille.
3. Scientific Foundations

3.1. Modeling and landscape analysis

The modeling of problems, the analysis of structures (landscapes) of MOPs and the performance assessment of resolution methods are significant topics in the design of optimization methods. The effectiveness of metaheuristics depends on the properties of the problem and its landscape (roughness, convexity, etc). The notion of landscape has been first described in [64] by the way of the study of species evolution. Then, this notion has been used to analyze combinatorial optimization problems.

3.1.1. Modeling of problems

Generally there are several ways of modeling a given problem. First, one has to find the most suitable model for the type of resolution he or she plans to use. The choice can be made after a theoretical analysis of the model, or after computational experiments. The choice of the model depends on the type of method used. For example, a major issue in the design of exact methods is to find tight relaxations for the problem considered.

Let us note that many combinatorial optimization problems of the literature have been studied in their mono-objective form even if a lot of them are naturally of a multi-objective nature.

Therefore, in the DOLPHIN project, we address the modeling of MOPs in two phases. The first one consists in studying the mono-objective version of the problem, where all objectives but one are considered as constraints. In the second phase, we propose methods to adapt the mono-objective models or to create hand-tailored models for the multi-objective case. The models used may come from the first phase, or from the literature.

3.1.2. Analysis of the structure of a problem

The landscape is defined by a neighborhood operator and can be represented by a graph $G = (V, E)$. The vertices represent the solutions of the problem and an edge $(e_1, e_2)$ exists if the solution $e_2$ can be obtained by an application of the neighborhood operator on the solution $e_1$. Then, considering this graph as the ground floor, we elevate each solution to an altitude equals to its cost. We obtain a surface, or landscape, made of peaks, valleys, plateaus, cliffs, etc. The problem lies in the difficulty to have a realistic view of this landscape.

Like others, we believe that the main point of interest in the domain of combinatorial optimization is not the design of the best algorithm for a large number of problems but the search for the most adapted method to an instance or a set of instances of a given problem. Therefore, we are convinced that no ideal metaheuristic, designed as a black-box, may exist.

Indeed, the first studies realized in our research group on the analysis of landscapes of different mono-objective combinatorial optimization problems (traveling salesman problem, quadratic assignment problem) have shown that not only different problems correspond to different structures but also that different instances of the same problem correspond to different structures.

For instance, we have realized a statistical study of the landscapes of the quadratic assignment problem. Some indicators that characterize the landscape of an instance have been proposed and a taxonomy of the instances including three classes has been deduced. Hence it is not enough to adapt the method to the problem under study but it is necessary to specialize it according to the type of the treated instance.

So in its studies of mono-objective problems, the DOLPHIN research group has introduced into the resolution methods some information about the problem to be solved. The landscapes of some combinatorial problems have been studied in order to investigate the intrinsic natures of their instances. The resulting information has been inserted into an optimization strategy and has allowed the design of efficient and robust hybrid methods. The extension of these studies to multi-objective problems is a part of the DOLPHIN project [62], [63].
3.1.3. Performance assessment

The DOLPHIN project is also interested in the performance assessment of multi-objective optimization methods. Nowadays, statistical techniques developed for mono-objective problems can be adapted to the multi-objective case. Nevertheless, specific tools are necessary in many situations: for example, the comparison of two different algorithms is relatively easy in the mono-objective case - we compare the quality of the best solution obtained in a fixed time, or the time needed to obtain a solution of a certain quality. The same idea cannot be immediately transposed to the case where the output of the algorithms is a set of solutions having several quality measures, and not a single solution.

Various indicators have been proposed in the literature for evaluating the performance of multi-objective optimization methods but no indicator seems to outperform the others [65]. The DOLPHIN research group has proposed two indicators: the contribution and the entropy [59]. The contribution evaluates the supply in term of Pareto-optimal solutions of a front compared to another one. The entropy gives an idea of the diversity of the solutions found. These two metrics are used to compare the different metaheuristics in the research group, for example in the resolution of the bi-objective flow-shop problem, and also to show the contribution of the various mechanisms introduced in these metaheuristics.

3.1.4. Goals

One of the main issues in the DOLPHIN project is the study of the landscape of multi-objective problems and the performance assessment of multi-objective optimization methods to design efficient and robust resolution methods:

- **Landscape study:** The goal here is to extend the study of landscapes of the mono-objective combinatorial optimization problems to multi-objective problems in order to determine the structure of the Pareto frontier and to integrate this knowledge about the problem structure in the design of resolution methods.
  
  This study has been initiated for the bi-objective flow-shop problem. We have studied the convexity of the frontiers obtained in order to show the interest of our Pareto approach compared to an aggregation approach, which only allows one to obtain the Pareto solutions situated on the convex hull of the Pareto front (supported solutions).

  Our preliminary study of the landscape of the bi-objective flow-shop problem shows that the supported solutions are very closed to each other. This remark leads us to improve an exact method initially proposed for bi-objective problems. Furthermore, a new exact method able to deal with any number of objectives has been designed.

- **Performance assessment:** The goal here is to extend GUIMOO in order to provide efficient visual and metric tools for evaluating the assessment of multi-objective resolution methods.

3.2. Hybrid multi-objective optimization methods

The success of metaheuristics is based on their ability to find efficient solutions in a reasonable time [58]. But with very large problems and/or multi-objective problems, efficiency of metaheuristics may be compromised. Hence, in this context it is necessary to integrate metaheuristics in more general schemes in order to develop even more efficient methods. For instance, this can be done by different strategies such as cooperation and parallelization.

The DOLPHIN project deals with “a posteriori” multi-objective optimization where the set of Pareto solutions (solutions of best compromise) have to be generated in order to give the decision maker the opportunity to choose the solution that interests him/her.

Population-based methods, such as evolutionary algorithms, are well fitted for multi-objective problems, as they work with a set of solutions [54], [57]. To be convinced one may refer to the list of references on Evolutionary Multi-objective Optimization maintained by Carlos A. Coello Coello⁴, which contains more

⁴http://www.lania.mx/~ccoello/EMOO/EMOObib.html
than 5500 references. One of the objectives of the project is to propose advanced search mechanisms for intensification and diversification. These mechanisms have been designed in an adaptive manner, since their effectiveness is related to the landscape of the MOP and to the instance solved.

In order to assess the performances of the proposed mechanisms, we always proceed in two steps: first, we carry out experiments on academic problems, for which some best known results exist; second, we use real industrial problems to cope with large and complex MOPs. The lack of references in terms of optimal or best known Pareto set is a major problem. Therefore, the obtained results in this project and the test data sets will be available at the URL http://dolphin.lille.inria.fr/ at "benchmark".

3.2.1. Cooperation of metaheuristics

In order to benefit from the various advantages of the different metaheuristics, an interesting idea is to combine them. Indeed, the hybridization of metaheuristics allows the cooperation of methods having complementary behaviors. The efficiency and the robustness of such methods depend on the balance between the exploration of the whole search space and the exploitation of interesting areas.

Hybrid metaheuristics have received considerable interest these last years in the field of combinatorial optimization. A wide variety of hybrid approaches have been proposed in the literature and give very good results on numerous single objective optimization problems, which are either academic (traveling salesman problem, quadratic assignment problem, scheduling problem, etc) or real-world problems. This efficiency is generally due to the combinations of single-solution based methods (iterative local search, simulated annealing, tabu search, etc) with population-based methods (genetic algorithms, ants search, scatter search, etc). A taxonomy of hybridization mechanisms may be found in [61]. It proposes to decompose these mechanisms into four classes:

- **LRH class - Low-level Relay Hybrid**: This class contains algorithms in which a given metaheuristic is embedded into a single-solution metaheuristic. Few examples from the literature belong to this class.
- **LTH class - Low-level Teamwork Hybrid**: In this class, a metaheuristic is embedded into a population-based metaheuristic in order to exploit strengths of single-solution and population-based metaheuristics.
- **HRH class - High-level Relay Hybrid**: Here, self contained metaheuristics are executed in a sequence. For instance, a population-based metaheuristic is executed to locate interesting regions and then a local search is performed to exploit these regions.
- **HTH class - High-level Teamwork Hybrid**: This scheme involves several self-contained algorithms performing a search in parallel and cooperating. An example will be the island model, based on GAs, where the population is partitioned into small subpopulations and a GA is executed per subpopulation. Some individuals can migrate between subpopulations.

Let us notice, that if hybrid methods have been studied in the mono-criterion case, their application in the multi-objective context is not yet widely spread. The objective of the DOLPHIN project is to integrate specificities of multi-objective optimization into the definition of hybrid models.

3.2.2. Cooperation between metaheuristics and exact methods

Until now only few exact methods have been proposed to solve multi-objective problems. They are based either on a Branch-and-bound approach, on the algorithm $A^*$, or on dynamic programming. However, these methods are limited to two objectives and, most of the time, cannot be used on a complete large scale problem. Therefore, sub search spaces have to be defined in order to use exact methods. Hence, in the same manner as hybridization of metaheuristics, the cooperation of metaheuristics and exact methods is also a main issue in this project. Indeed, it allows us to use the exploration capacity of metaheuristics, as well as the intensification ability of exact methods, which are able to find optimal solutions in a restricted search space. Sub search spaces have to be defined along the search. Such strategies can be found in the literature, but they are only applied to mono-objective academic problems.
We have extended the previous taxonomy for hybrid metaheuristics to the cooperation between exact methods and metaheuristics. Using this taxonomy, we are investigating cooperative multi-objective methods. In this context, several types of cooperations may be considered, according to the way the metaheuristic and the exact method cooperate. For instance, a metaheuristic can use an exact method for intensification or an exact method can use a metaheuristic to reduce the search space.

Moreover, a part of the DOLPHIN project deals with studying exact methods in the multi-objective context in order: i) to be able to solve small size problems and to validate proposed heuristic approaches; ii) to have more efficient/dedicated exact methods that can be hybridized with metaheuristics. In this context, the use of parallelism will push back limits of exact methods, which will be able to explore larger size search spaces.

3.2.3. Goals

Based on the previous works on multi-objective optimization, it appears that to improve metaheuristics, it becomes essential to integrate knowledge about the problem structure. This knowledge can be gained during the search. This would allow us to adapt operators which may be specific for multi-objective optimization or not. The goal here is to design auto-adaptive methods that are able to react to the problem structure. Moreover, regarding the hybridization and the cooperation aspects, the objectives of the DOLPHIN project are to deepen these studies as follows:

- **Design of metaheuristics for the multi-objective optimization**: To improve metaheuristics, it becomes essential to integrate knowledge about the problem structure, which we may get during the execution. This would allow us to adapt operators that may be specific for multi-objective optimization or not. The goal here is to design auto-adaptive methods that are able to react to the problem structure.

- **Design of cooperative metaheuristics**: Previous studies show the interest of hybridization for a global optimization and the importance of problem structure study for the design of efficient methods. It is now necessary to generalize hybridization of metaheuristics and to propose adaptive hybrid models that may evolve during the search while selecting the appropriate metaheuristic. Multi-objective aspects have to be introduced in order to cope with the specificities of multi-objective optimization.

- **Design of cooperative schemes between exact methods and metaheuristics**: Once the study on possible cooperation schemes is achieved, we will have to test and compare them in the multi-objective context.

- **Design and conception of parallel metaheuristics**: Our previous works on parallel metaheuristics allow us to speed up the resolution of large scale problems. It could be also interesting to study the robustness of the different parallel models (in particular in the multi-objective case) and to propose rules that determine, given a specific problem, which kind of parallelism to use. Of course these goals are not disjoined and it will be interesting to simultaneously use hybrid metaheuristics and exact methods. Moreover, those advanced mechanisms may require the use of parallel and distributed computing in order to easily make cooperating methods evolve simultaneously and to speed up the resolution of large scale problems.

- **Validation**: In order to validate the obtained results we always proceed in two phases: validation on academic problems, for which some best known results exist and use on real problems (industrial) to cope with problem size constraints.

Moreover, those advanced mechanisms are to be used in order to integrate the distributed multi-objective aspects in the ParadisEO platform (see the paragraph on software platform).

3.3. Parallel multi-objective optimization: models and software frameworks

Parallel and distributed computing may be considered as a tool to speedup the search to solve large MOPs and to improve the robustness of a given method. Moreover, the joint use of parallelism and cooperation allows improvements on the quality of the obtained Pareto sets. Following this objective, we will design and implement parallel models for metaheuristics (evolutionary algorithms, tabu search approach) and exact methods (branch-and-bound algorithm, branch-and-cut algorithm) to solve different large MOPs.
One of the goals of the DOLPHIN project is to integrate the developed parallel models into software frameworks. Several frameworks for parallel distributed metaheuristics have been proposed in the literature. Most of them focus only either on evolutionary algorithms or on local search methods. Only few frameworks are dedicated to the design of both families of methods. On the other hand, existing optimization frameworks either do not provide parallelism at all or just supply at most one parallel model. In this project, a new framework for parallel hybrid metaheuristics is proposed, named Parallel and Distributed Evolving Objects (ParadisEO) based on EO. The framework provides in a transparent way the hybridization mechanisms presented in the previous section, and the parallel models described in the next section. Concerning the developed parallel exact methods for MOPs, we will integrate them into well-known frameworks such as COIN.

3.3.1. Parallel models

According to the family of addressed metaheuristics, we may distinguish two categories of parallel models: parallel models that manage a single solution, and parallel models that handle a population of solutions. The major single solution-based parallel models are the following: the parallel neighborhood exploration model and the multi-start model.

- The parallel neighborhood exploration model is basically a "low level" model that splits the neighborhood into partitions that are explored and evaluated in parallel. This model is particularly interesting when the evaluation of each solution is costly and/or when the size of the neighborhood is large. It has been successfully applied to the mobile network design problem (see Application section).

- The multi-start model consists in executing in parallel several local searches (that may be heterogeneous), without any information exchange. This model raises particularly the following question: is it equivalent to execute \( k \) local searches during a time \( t \) than executing a single local search during \( k \times t \) ? To answer this question we tested a multi-start Tabu search on the quadratic assignment problem. The experiments have shown that the answer is often landscape-dependent. For example, the multi-start model may be well-suited for landscapes with multiple basins.

Parallel models that handle a population of solutions are mainly: the island model, the central model and the distributed evaluation of a single solution. Let us notice that the last model may also be used with single-solution metaheuristics.

- In the island model, the population is split into several sub-populations distributed among different processors. Each processor is responsible of the evolution of one sub-population. It executes all the steps of the metaheuristic from the selection to the replacement. After a given number of generations (synchronous communication), or when a convergence threshold is reached (asynchronous communication), the migration process is activated. Then, exchanges of solutions between sub-populations are realized, and received solutions are integrated into the local sub-population.

- The central (Master/Worker) model allows us to keep the sequentiality of the original algorithm. The master centralizes the population and manages the selection and the replacement steps. It sends sub-populations to the workers that execute the recombination and evaluation steps. The latter returns newly evaluated solutions to the master. This approach is efficient when the generation and evaluation of new solutions is costly.

- The distributed evaluation model consists in a parallel evaluation of each solution. This model has to be used when, for example, the evaluation of a solution requires access to very large databases (data mining applications) that may be distributed over several processors. It may also be useful in a multi-objective context, where several objectives have to be computed simultaneously for a single solution.

As these models have now been identified, our objective is to study them in the multi-objective context in order to use them advisedly. Moreover, these models may be merged to combine different levels of parallelism and to obtain more efficient methods [56], [60].
3.3.2. Goals

Our objectives focus on these issues are the following:

- **Design of parallel models for metaheuristics and exact methods for MOPs**: We will develop parallel cooperative metaheuristics (evolutionary algorithms and local search algorithms such as the Tabu search) for solving different large MOPs. Moreover, we are designing a new exact method, named PPM (Parallel Partition Method), based on branch and bound and branch and cut algorithms. Finally, some parallel cooperation schemes between metaheuristics and exact algorithms have to be used to solve MOPs in an efficient manner.

- **Integration of the parallel models into software frameworks**: The parallel models for metaheuristics will be integrated in the ParadisEO software framework. The proposed multi-objective exact methods must be first integrated into standard frameworks for exact methods such as COIN and BOB++. A coupling with ParadisEO is then needed to provide hybridization between metaheuristics and exact methods.

- **Efficient deployment of the parallel models on different parallel and distributed architecture including GRIDs**: The designed algorithms and frameworks will be efficiently deployed on non-dedicated networks of workstations, dedicated cluster of workstations and SMP (Symmetric Multi-processors) machines. For GRID computing platforms, peer to peer (P2P) middlewares (XtremWeb-Condor) will be used to implement our frameworks. For this purpose, the different optimization algorithms may be re-visited for their efficient deployment.
MODAL Team

3. Scientific Foundations

3.1. Generative model design

The first objective of MODAL consists in designing, analyzing, estimating and evaluating new generative parametric models for multivariate and/or heterogeneous data. It corresponds typically to continuous and categorical data but it includes also other widespread ones like ordinal, functional, ranks,... Designed models have to take into account potential correlations between variables while being (1) justifiable and realistic, (2) meaningful and parsimoniously parameterized, (3) of low computational complexity. The main purpose is to identify a few theoretical and general principles for model generation, loosely dependent on the variable nature. In this context, we propose two concurrent approaches which could be general enough for dealing with correlation between many types of homogeneous or heterogeneous variables:

- Designs general models by combining two extreme models (full dependent and full independent) which are well-defined for most of variables;
- Uses kernels as a general way for dealing with multivariate and heterogeneous variables.

3.2. Data visualization

The second objective of MODAL is to propose meaningful and quite accurate low dimensional visualizations of data typically in two-dimensional (2D) space, less frequently in one-dimensional (1D) or three-dimensional (3D) spaces, by using the generative models designed in the first objective. We propose also to visualize simultaneously the data and the model. All visualizations will depend on the aim at hand (typically clustering, classification or density estimation). The main originality of this objective lies in the use of models for visualization, strategy from which we expect to have a better control on the subjectivity necessarily induced by any graphical display. In addition, the proposed approach has to be general enough to be independent on the variable nature. Note that the visualization objective is consistent with the dissemination of our methodologies through specific softwares. Indeed, displaying data is an important step in the data analysis process.
3. Scientific Foundations

3.1. Fast parametric estimation and its applications

Parametric estimation may often be formalized as follows:

\[ y = F(x, \Theta) + n, \]  

(1)

where:

- the measured signal \( y \) is a functional \( F \) of the "true" signal \( x \), which depends on a set \( \Theta \) of parameters,
- \( n \) is a noise corrupting the observation.

Finding a "good" approximation of the components of \( \Theta \) has been the subject of a huge literature in various fields of applied mathematics. Most of those researches have been done in a probabilistic setting, which necessitates a good knowledge of the statistical properties of \( n \). Our project is devoted to a new standpoint which does not require this knowledge and which is based on the following tools, which are of algebraic flavor:

- differential algebra\(^2\), which plays with respect to differential equations a similar role to commutative algebra with respect to algebraic equations;
- module theory, i.e., linear algebra over rings which are not necessarily commutative;
- operational calculus which was the most classical tool among control and mechanical engineers\(^3\).

3.1.1. Linear identifiability

In most problems appearing in linear control as well as in signal processing, the unknown parameters are linearly identifiable: standard elimination procedures are yielding the following matrix equation

\[ P \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_r \end{pmatrix} = Q, \]  

(2)

where:

- \( \theta_i, 1 \leq i \leq r \), represents unknown parameter,
- \( P \) is a \( r \times r \) square matrix and \( Q \) is a \( r \times 1 \) column matrix,
- the entries of \( P \) and \( Q \) are finite linear combinations of terms of the form \( t^\mu \frac{d^\nu \xi}{dt^\nu}, \mu, \nu \geq 0 \), where \( \xi \) is an input or output signal,
- the matrix \( P \) is generically invertible, i.e., \( \det(P) \neq 0 \).

\(^2\)Differential algebra was introduced in nonlinear control theory by one of us almost twenty years ago for understanding some specific questions like input-output inversion. It allowed to recast the whole of nonlinear control into a more realistic light. The best example is of course the discovery of flat systems which are now quite popular in industry.

\(^3\)Operational calculus is often formalized via the Laplace transform whereas the Fourier transform is today the cornerstone in estimation. Note that the one-sided Laplace transform is causal, but the Fourier transform over \( R \) is not.
3.1.2. How to deal with perturbations and noises?

With noisy measurements equation (2) becomes:

\[ P \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_r \end{pmatrix} = Q + R, \quad (3) \]

where \( R \) is a \( r \times 1 \) column matrix, whose entries are finite linear combination of terms of the form \( \mu \frac{d^\nu}{dt^\nu}, \mu, \nu \geq 0 \), where \( \eta \) is a perturbation or a noise.

3.1.2.1. Structured perturbations

A perturbation \( \pi \) is said to be structured if, and only if, it is annihilated by a linear differential operator of the form \( \sum_{\text{finite}} a_k(t) \frac{d^k}{dt^k} \), where \( a_k(t) \) is a rational function of \( t \), i.e., \( \left( \sum_{\text{finite}} a_k(t) \frac{d^k}{dt^k} \right) \pi = 0 \). Note that many classical perturbations like a constant bias are annihilated by such an operator. An unstructured noise cannot be annihilated by a non-zero differential operator.

By well known properties of the non-commutative ring of differential operators, we can multiply both sides of equation (3) by a suitable differential operator \( \Delta \) such that equation (3) becomes:

\[ \Delta P \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_r \end{pmatrix} = \Delta Q + R', \quad (4) \]

where the entries of the \( r \times 1 \) column matrix \( R' \) are unstructured noises.

3.1.2.2. Attenuating unstructured noises

Unstructured noises are usually dealt with stochastic processes like white Gaussian noises. They are considered here as highly fluctuating phenomena, which may therefore be attenuated via low pass filters. Note that no precise knowledge of the statistical properties of the noises is required.

3.1.2.3. Comments

Although the previous noise attenuation\(^4\) may be fully explained via formula (4), its theoretical comparison\(^5\) with today’s literature\(^6\) has yet to be done. It will require a complete resetting of the notions of noises and perturbations. Besides some connections with physics, it might lead to quite new “epistemological” issues \(^80\).

3.1.3. Some hints on the calculations

The time derivatives of the input and output signals appearing in equations (2), (3), (4) can be suppressed in the two following ways which might be combined:

- integrate both sides of the equation a sufficient number of times,
- take the convolution product of both sides by a suitable low pass filter.

The numerical values of the unknown parameters \( \Theta = (\theta_1, \cdots, \theta_r) \) can be obtained by integrating both sides of the modified equation (4) during a very short time interval.

\(^4\)It is reminiscent to what most practitioners in electronics are doing.
\(^5\)Let us stress again that many computer simulations and several laboratory experiments have been already successfully achieved and can be quite favorably compared with the existing techniques.
\(^6\)Especially in signal processing.
3.1.4. A first, very simple example

Let us illustrate on a very basic example, the grounding ideas of the algebraic approach, based on algebra. For this, consider the first order, linear system:

$$\dot{y}(t) = ay(t) + u(t) + \gamma_0,$$

where \(a\) is an unknown parameter to be identified and \(\gamma_0\) is an unknown, constant perturbation. With the notations of operational calculus and \(y_0 = y(0)\), equation (5) reads:

$$s\hat{y}(s) = a\hat{y}(s) + \hat{u}(s) + y_0 + \frac{\gamma_0}{s}$$

where \(\hat{y}(s)\) represents Laplace transform.

In order to eliminate the term \(\gamma_0\), multiply first the two hand-sides of this equation by \(s\) and, then, take their derivatives with respect to \(s\):

$$\frac{d}{ds} \left[ s \left\{ s\hat{y}(s) = a\hat{y}(s) + \hat{u}(s) + y_0 + \frac{\gamma_0}{s} \right\} \right]$$

$$\Rightarrow 2s\hat{y}(s) + s^2\hat{y}'(s) = a (s\hat{y}'(s) + \hat{y}(s)) + s\hat{u}'(s) + \hat{u}(s) + y_0.$$  

Recall that \(\hat{y}'(s) \triangleq \frac{d\hat{y}(s)}{ds}\) corresponds to \(-ty(t)\). Assume \(y_0 = 0\) for simplicity’s sake. Then, for any \(\nu > 0\),

$$s^{-\nu} \left[ 2s\hat{y}(s) + s^2\hat{y}'(s) \right] = s^{-\nu} [a(s\hat{y}'(s) + \hat{y}(s)) + s\hat{u}'(s) + \hat{u}(s)].$$

For \(\nu = 3\), we obtained the estimated value \(a\):

$$a = \frac{2 \int_0^T d\lambda \int_0^\lambda y(t)dt - \int_0^T \int_0^\lambda t u(t)dt}{\int_0^T \int_0^\lambda y(t)dt - \int_0^T \int_0^\lambda t u(t)dt}$$

Since \(T > 0\) can be very small, estimation via (10) is very fast.

Note that equation (10) represents an on-line algorithm that only involves two kinds of operations on \(u\) and \(y\): (1) multiplications by \(t\), and (2) integrations over a pre-selected time interval.

If we now consider an additional noise, of zero mean, in (5), say:

$$\dot{y}(t) = ay(t) + u(t) + \gamma_0 + n(t),$$

it will be considered as fast fluctuating signal. The order \(\nu\) in (9) determines the order of iterations in the integrals (3 integrals in (10)). Those iterated integrals are low-pass filters which are attenuating the fluctuations.

\[\text{If } y_0 \neq 0 \text{ one has to take above derivatives of order 2 with respect to } s, \text{ in order to eliminate the initial condition.}\]
This example, even simple, clearly demonstrates how algebraic’s techniques proceed:

- they are algebraic: operations on \( s \)-functions;
- they are non-asymptotic: parameter \( a \) is obtained from (10) in finite time;
- they are deterministic: no knowledge of the statistical properties of the noise \( n \) is required.

### 3.1.5. A second simple example, with delay

Consider the first order, linear system with constant input delay:

\[
\dot{y}(t) + ay(t) = y(0)\delta + \gamma_0 H + bu(t - \tau).
\]

Here we use a distributional-like notation where \( \delta \) denotes the Dirac impulse and \( H \) is its integral, i.e., the Heaviside function (unit step). Still for simplicity, we suppose that the parameter \( a \) is known. The parameter to be identified is now the delay \( \tau \). As previously, \( \gamma_0 \) is a constant perturbation, \( a, b, \) and \( \tau \) are constant parameters. Consider also a step input \( u = u_0 H \). A first order derivation yields:

\[
\ddot{y} + a\dot{y} = \varphi_0 + \gamma_0 \delta + bu_0 \delta_\tau,
\]

where \( \delta_\tau \) denotes the delayed Dirac impulse and \( \varphi_0 = (\dot{y}(0) + ay(0))\delta + y(0)\delta^{(1)} \), of order 1 and support \( \{0\} \), contains the contributions of the initial conditions. According to Schwartz theorem, multiplication by a function \( \alpha \) such that \( \alpha(0) = \alpha'(0) = 0 \), \( \alpha(\tau) = 0 \) yields interesting simplifications. For instance, choosing \( \alpha(t) = t^3 - \tau t^2 \) leads to the following equalities (to be understood in the distributional framework):

\[
\begin{align*}
t^3 [\ddot{y} + a\dot{y}] &= \tau t^2 [\ddot{y} + a\dot{y}], \\
bu_0 t^3 \delta_\tau &= bu_0 \tau t^2 \delta_\tau.
\end{align*}
\]

The delay \( \tau \) becomes available from \( k \geq 1 \) successive integrations (represented by the operator \( H \)), as follows:

\[
\tau = \frac{H^k(w_0 + aw_2)}{H^k(w_1 + aw_3)} \quad t > \tau,
\]

where the \( w_i \) are defined, using the notation \( z_i = t^i y \), by:

\[
\begin{align*}
w_0 &= t^3 y^{(2)} = -6z_1 + 6z_2^{(1)} - z_3^{(2)}, \\
w_1 &= t^2 y^{(2)} = -2z_0 + 4z_1^{(1)} - z_2^{(2)}, \\
w_2 &= t^2 y^{(1)} = 2z_1 - z_2^{(1)}, \\
w_3 &= t^3 y^{(1)} = 3z_2 - z_3^{(1)}.
\end{align*}
\]

These coefficients show that \( k \geq 2 \) integrations are avoiding any derivation in the delay identification.

---

8This example is taken from [69]. For further details, we suggest the reader to refer to it.

9In this document, for the sake of simplicity, we make an abuse of the language since we merge in a single notation the Heaviside function \( H \) and the integration operator. To be rigorous, the iterated integration (\( k \) times) corresponds, in the operational domain, to a division by \( s^k \), whereas the convolution with \( H \) (\( k \) times) corresponds to a division by \( s^k/(k-1)! \). For \( k = 0 \), there is no difference and \( H \ast y \) realizes the integration of \( y \). More generally, since we will always apply these operations to complete equations (left- and right-hand sides), the factor \((k-1)!\) makes no difference.
Figure 1 gives a numerical simulation with $k = 2$ integrations and $a = 2, b = 1, \tau = 0.6$. $y(0) = 0.3, \gamma_0 = 2, u_0 = 1$. Due to the non identifiability over $(0, \tau)$, the delay $\tau$ is set to zero until the numerator or the denominator in the right hand side of (15) reaches a significant nonzero value.

Again, note the realization algorithm (15) involves two kinds of operators: (1) integrations and (2) multiplications by $t$.

It relies on the measurement of $y$ and on the knowledge of $a$. If $a$ is also unknown, the same approach can be utilized for a simultaneous identification of $a$ and $\tau$. The following relation is derived from (14):

$$\tau (H^k w_1) + a \tau (H^k w_2) - a (H^k w_3) = H^k w_0,$$

and a linear system with unknown parameters $(\tau, a \tau, a)$ is obtained by using different integration orders:

$$
\begin{pmatrix}
H^2 w_1 & H^2 w_2 & H^2 w_3 \\
H^3 w_1 & H^3 w_2 & H^3 w_3 \\
H^4 w_1 & H^4 w_2 & H^4 w_3
\end{pmatrix}
\begin{pmatrix}
\hat{\tau} \\
\hat{a} \\
\hat{\gamma}
\end{pmatrix}
= 
\begin{pmatrix}
H^2 w_0 \\
H^3 w_0 \\
H^4 w_0
\end{pmatrix}.
$$

The resulting numerical simulations are shown in Figure 2. For identifiability reasons, the obtained linear system may be not consistent for $t < \tau$.

3.2. Finite time estimation of derivatives

Numerical differentiation, i.e., determining the time derivatives of various orders of a noisy time signal, is an important but difficult ill-posed theoretical problem. This fundamental issue has attracted a lot of attention in many fields of engineering and applied mathematics (see, e.g. in the recent control literature [70], [71], [91], [90], [97], [98], and the references therein).

3.2.1. Model-free techniques for numerical differentiation

A common way of estimating the derivatives of a signal is to resort to a least squares fitting and then take the derivatives of the resulting function. In [101], [99], this problem was revised through our algebraic approach. The approach can be briefly explained as follows:
The coefficients of a polynomial time function are linearly identifiable. Their estimation can therefore be achieved as above. Indeed, consider the real-valued polynomial function

\[ x_N(t) = \sum_{\nu=0}^{N} x^{(\nu)}(0) t^\nu \] 

where \( t \geq 0 \), of degree \( N \). Rewrite it in the well-known notations of operational calculus:

\[ X_N(s) = \sum_{\nu=0}^{N} x^{(\nu)}(0) s^{\nu+1} \]

Here, we use \( \frac{d}{dt} \), which corresponds in the time domain to the multiplication by \( -t \). Multiply both sides by \( s^{N+1} \), \( \alpha = 0, 1, \cdots, N \). The quantities \( x^{(\nu)}(0) \), \( \nu = 0, 1, \cdots, N \) are given by the triangular system of linear equations:

\[ \frac{d^\alpha}{ds^\alpha} s^{N+1} X_N = \frac{d^\alpha}{ds^\alpha} \left( \sum_{\nu=0}^{N} x^{(\nu)}(0) s^{N-\nu} \right) \]  

(17)

The time derivatives, i.e., \( s^{\mu} \frac{d^{N+1} x_N}{ds^{\mu}} \), \( \mu = 1, \cdots, N \), \( 0 \leq \tau \leq N \), are removed by multiplying both sides of Equation (17) by \( s^{-N} \), \( N > N \).

For an arbitrary analytic time function, apply the preceding calculations to a suitable truncated Taylor expansion. Consider a real-valued analytic time function defined by the convergent power series

\[ x(t) = \sum_{\nu=0}^{\infty} x^{(\nu)}(0) t^\nu \] 

where \( 0 \leq t < \rho \). Approximate \( x(t) \) in the interval \( (0, \varepsilon) \), \( 0 < \varepsilon \leq \rho \), by its truncated Taylor expansion \( x_N(t) = \sum_{\nu=0}^{N} x^{(\nu)}(0) t^\nu \) of order \( N \). Introduce the operational analogue of \( x(t) \), i.e., \( X(s) = \sum_{\nu \geq 0} x^{(\nu)}(0) s^{N-\nu} \). Denote by \( [x^{(\nu)}(0)]_{\nu \leq N} \), \( 0 \leq \nu \leq N \), the numerical estimate of \( x^{(\nu)}(0) \), which is obtained by replacing \( X_N(s) \) by \( X(s) \) in Eq. (17). It can be shown that a good estimate is obtained in this way.

Thus, using elementary differential algebraic operations, we derive explicit formulae yielding point-wise derivative estimation for each given order. Interesting enough, it turns out that the Jacobi orthogonal polynomials [112] are inherently connected with the developed algebraic numerical differentiators. A least-squares interpretation then naturally follows [100], [101] and this leads to a key result: the algebraic numerical differentiation is as efficient as an appropriately chosen time delay. Though, such a delay may not be tolerable in some real-time applications. Moreover, instability generally occurs when introducing delayed signals in a
control loop. Note however that since the delay is known \textit{a priori}, it is always possible to derive a control law which compensates for its effects (see [110]). A second key feature of the algebraic numerical differentiators is its very low complexity which allows for a real-time implementation. Indeed, the $n^{th}$ order derivative estimate (that can be directly managed for $n \geq 2$, without using $n$ cascaded estimators) is expressed as the output of the linear time-invariant filter, with finite support impulse response $h_{n,\mu,n,r}(\cdot)$. Implementing such a stable and causal filter is easy and simple. This is achieved either in continuous-time or in discrete-time when only discrete-time samples of the observation are available. In the latter case, we obtain a tapped delay line digital filter by considering any numerical integration method with equally-spaced abscissas.

### 3.2.2. Model-based estimation of derivatives

If we consider that the derivatives to be estimated are unmeasured states of the process that generates the signal, differentiation techniques can be regarded as left invertibility algorithms. In this sense, the previous algebraic estimation achieves a “model-free” left inversion. Now, when such a model is available, the \textit{finite-time observers}, relying on higher order sliding modes [105] and homogeneity properties [106], [102], also represent possible non-asymptotic algorithms for differentiation\textsuperscript{10}. Using such model-based techniques appears to be complementary\textsuperscript{11} and we already obtained left-inversion results for several classes of models: linear systems [87], nonlinear systems [68], delay systems [2] and hybrid systems [96].

\textsuperscript{10} Usually, observer design yields asymptotic convergence of the estimation error dynamics. The main advantages of such a technique in the case of linear systems are simplicity of design, estimation with a filtering action and global stability property. Nevertheless, the filtering property is not ensured for nonlinear systems and the stability property is generally obtained only locally. For these reasons, in the case of nonlinear systems, finite-time observers and estimators have been proposed in the literature [98], [106], [107], [86]...

\textsuperscript{11} The choice between the two approaches will be done after comparison with respect to the indicators 1, 2, 3, and taking into account the application (for instance, the system bandwidth, system dimension), the kind of discontinuity, the observer in the control loop or not...
3. Scientific Foundations

3.1. Introduction

SEQUEL is primarily grounded on two domains:

- the problem of decision under uncertainty,
- statistical analysis and statistical learning, which provide the general concepts and tools to solve this problem.

To help the reader who is unfamiliar with these questions, we briefly present key ideas below.

3.2. Decision under uncertainty

The phrase “Decision under uncertainty” refers to the problem of taking decisions when we do not have a full knowledge neither of the situation, nor of the consequences of the decisions, as well as when the consequences of decision are non deterministic.

We introduce two specific sub-domains, namely the Markov decision processes which models sequential decision problems, and bandit problems.

3.2.1. Markov decision processes

Sequential decision processes occupy the heart of the SEQUEL project; a detailed presentation of this problem may be found in Puterman’s book [78].

A Markov Decision Process (MDP) is defined as the tuple \((X, A, P, r)\) where \(X\) is the state space, \(A\) is the action space, \(P\) is the probabilistic transition kernel, and \(r : X \times A \times X \rightarrow \mathbb{R}\) is the reward function. For the sake of simplicity, we assume in this introduction that the state and action spaces are finite. If the current state (at time \(t\)) is \(x \in X\) and the chosen action is \(a \in A\), then the Markov assumption means that the transition probability to a new state \(x' \in X\) (at time \(t + 1\)) only depends on \((x, a)\). We write \(p(x'|x, a)\) the corresponding transition probability. During a transition \((x, a) \rightarrow x'\), a reward \(r(x, a, x')\) is incurred.

In the MDP \((X, A, P, r)\), each initial state \(x_0\) and action sequence \(a_0, a_1, \ldots\) gives rise to a sequence of states \(x_1, x_2, \ldots\), satisfying \(P(x_{t+1} = x'|x_t = x, a_t = a) = p(x'|x, a)\), and rewards \(r_1, r_2, \ldots\) defined by \(r_t = r(x_t, a_t, x_{t+1})\).

The history of the process up to time \(t\) is defined to be \(H_t = (x_0, a_0, \ldots, x_{t-1}, a_{t-1}, x_t)\). A policy \(\pi\) is a sequence of functions \(\pi_0, \pi_1, \ldots\), where \(\pi_t\) maps the space of possible histories at time \(t\) to the space of probability distributions over the space of actions \(A\). To follow a policy means that, in each time step, we assume that the process history up to time \(t\) is \(x_0, a_0, \ldots, x_t\) and the probability of selecting an action \(a\) is equal to \(\pi_t(x_0, a_0, \ldots, x_t)(a)\). A policy is called stationary (or Markovian) if \(\pi_t\) depends only on the last visited state. In other words, a policy \(\pi = (\pi_0, \pi_1, \ldots)\) is called stationary if \(\pi_t(x_0, a_0, \ldots, x_t) = \pi_0(x_t)\) holds for all \(t \geq 0\). A policy is called deterministic if the probability distribution prescribed by the policy for any history is concentrated on a single action. Otherwise it is called a stochastic policy.

We move from an MD process to an MD problem by formulating the goal of the agent, that is what the sought policy \(\pi\) has to optimize? It is very often formulated as maximizing (or minimizing), in expectation, some functional of the sequence of future rewards. For example, an usual functional is the infinite-time horizon sum of discounted rewards. For a given (stationary) policy \(\pi\), we define the value function \(V^\pi(x)\) of that policy \(\pi\) at a state \(x \in X\) as the expected sum of discounted future rewards given that we state from the initial state \(x\) and follow the policy \(\pi\):

\[1\text{Note that for simplicity, we considered the case of a deterministic reward function, but in many applications, the reward } r_t \text{ itself is a random variable.} \]
\[ V^\pi(x) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r_t | x_0 = x, \pi \right], \]  

where \( \mathbb{E} \) is the expectation operator and \( \gamma \in (0, 1) \) is the discount factor. This value function \( V^\pi \) gives an evaluation of the performance of a given policy \( \pi \). Other functionals of the sequence of future rewards may be considered, such as the undiscounted reward (see the stochastic shortest path problems [69]) and average reward settings. Note also that, here, we considered the problem of maximizing a reward functional, but a formulation in terms of minimizing some cost or risk functional would be equivalent.

In order to maximize a given functional in a sequential framework, one usually applies Dynamic Programming (DP) [67], which introduces the optimal value function \( V^*(x) \), defined as the optimal expected sum of rewards when the agent starts from a state \( x \). We have \( V^*(x) = \sup_{\pi} V^\pi(x) \). Now, let us give two definitions about policies:

- We say that a policy \( \pi \) is optimal, if it attains the optimal values \( V^*(x) \) for any state \( x \in \mathcal{X} \), i.e., if \( V^\pi(x) = V^*(x) \) for all \( x \in \mathcal{X} \). Under mild conditions, deterministic stationary optimal policies exist [68]. Such an optimal policy is written \( \pi^* \).
- We say that a (deterministic stationary) policy \( \pi \) is greedy with respect to (w.r.t.) some function \( V \) (defined on \( \mathcal{X} \)) if, for all \( x \in \mathcal{X} \),

\[
\pi(x) \in \arg\max_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} p(x'|x,a) \left[ r(x,a,x') + \gamma V(x') \right].
\]

where \( \arg\max_{a \in \mathcal{A}} f(a) \) is the set of \( a \in \mathcal{A} \) that maximizes \( f(a) \). For any function \( V \), such a greedy policy always exists because \( \mathcal{A} \) is finite.

The goal of Reinforcement Learning (RL), as well as that of dynamic programming, is to design an optimal policy (or a good approximation of it).

The well-known Dynamic Programming equation (also called the Bellman equation) provides a relation between the optimal value function at a state \( x \) and the optimal value function at the successors states \( x' \) when choosing an optimal action: for all \( x \in \mathcal{X} \),

\[ V^*(x) = \max_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} p(x'|x,a) \left[ r(x,a,x') + \gamma V^*(x') \right]. \]  

(19)

The benefit of introducing this concept of optimal value function relies on the property that, from the optimal value function \( V^* \), it is easy to derive an optimal behavior by choosing the actions according to a policy greedy w.r.t. \( V^* \). Indeed, we have the property that a policy greedy w.r.t. the optimal value function is an optimal policy:

\[ \pi^*(x) \in \arg\max_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} p(x'|x,a) \left[ r(x,a,x') + \gamma V^*(x') \right]. \]  

(20)

In short, we would like to mention that most of the reinforcement learning methods developed so far are built on one (or both) of the two following approaches ([86]):

- Bellman’s dynamic programming approach, based on the introduction of the value function. It consists in learning a “good” approximation of the optimal value function, and then using it to derive a greedy policy w.r.t. this approximation. The hope (well justified in several cases) is that the performance \( V^\pi \) of the policy \( \pi \) greedy w.r.t. an approximation \( V \) of \( V^* \) will be close to optimality. This approximation issue of the optimal value function is one of the major challenge
inherent to the reinforcement learning problem. **Approximate dynamic programming** addresses the problem of estimating performance bounds (e.g. the loss in performance \( ||V^* - V^\pi|| \)) resulting from using a policy \( \pi \)-greedy w.r.t. some approximation \( V^- \) instead of an optimal policy) in terms of the approximation error \( ||V^* - V|| \) of the optimal value function \( V^* \) by \( V \). Approximation theory and Statistical Learning theory provide us with bounds in terms of the number of sample data used to represent the functions, and the capacity and approximation power of the considered function spaces.

- Pontryagin’s maximum principle approach, based on sensitivity analysis of the performance measure w.r.t. some control parameters. This approach, also called **direct policy search** in the Reinforcement Learning community aims at directly finding a good feedback control law in a parameterized policy space without trying to approximate the value function. The method consists in estimating the so-called **policy gradient**, *i.e.* the sensitivity of the performance measure (the value function) w.r.t. some parameters of the current policy. The idea being that an optimal control problem is replaced by a parametric optimization problem in the space of parameterized policies. As such, deriving a policy gradient estimate would lead to performing a stochastic gradient method in order to search for a local optimal parametric policy.

Finally, many extensions of the Markov decision processes exist, among which the Partially Observable MDPs (POMDPs) is the case where the current state does not contain all the necessary information required to decide for sure of the best action.

### 3.2.2. Bandits

Bandit problems illustrate the fundamental difficulty of decision making in the face of uncertainty: A decision maker must choose between what seems to be the best choice ("exploit"), or to test ("explore") some alternative, hoping to discover a choice that beats the current best choice.

The classical example of a bandit problem is deciding what treatment to give each patient in a clinical trial when the effectiveness of the treatments are initially unknown and the patients arrive sequentially. These bandit problems became popular with the seminal paper [79], after which they have found applications in diverse fields, such as control, economics, statistics, or learning theory.

Formally, a K-armed bandit problem \( (K \geq 2) \) is specified by K real-valued distributions. In each time step a decision maker can select one of the distributions to obtain a sample from it. The samples obtained are considered as rewards. The distributions are initially unknown to the decision maker, whose goal is to maximize the sum of the rewards received, or equivalently, to minimize the regret which is defined as the loss compared to the total payoff that can be achieved given full knowledge of the problem, *i.e.*, when the arm giving the highest expected reward is pulled all the time.

The name “bandit” comes from imagining a gambler playing with K slot machines. The gambler can pull the arm of any of the machines, which produces a random payoff as a result: When arm k is pulled, the random payoff is drawn from the distribution associated to k. Since the payoff distributions are initially unknown, the gambler must use exploratory actions to learn the utility of the individual arms. However, exploration has to be carefully controlled since excessive exploration may lead to unnecessary losses. Hence, to play well, the gambler must carefully balance exploration and exploitation. Auer et al. [66] introduced the algorithm UCB (Upper Confidence Bounds) that follows what is now called the “optimism in the face of uncertainty principle”. Their algorithm works by computing upper confidence bounds for all the arms and then choosing the arm with the highest such bound. They proved that the expected regret of their algorithm increases at most at a logarithmic rate with the number of trials, and that the algorithm achieves the smallest possible regret up to some sub-logarithmic factor (for the considered family of distributions).

### 3.3. Statistical analysis of time series

Many of the problems of machine learning can be seen as extensions of classical problems of mathematical statistics to their (extremely) non-parametric and model-free cases. Other machine learning problems are
founded on such statistical problems. Statistical problems of sequential learning are mainly those that are concerned with the analysis of time series. These problems are as follows.

3.3.1. Sequence prediction

Given a series of observations \( x_1, \ldots, x_n \) it is required to predict the probability distribution of the next outcome \( x_{n+1} \), before it is revealed and the process continues. Different goals can be formulated in this setting. One can either make some assumptions on the probability measure that generates the sequence \( x_1, \ldots, x_n, \ldots \), such as that the outcomes are independent and identically distributed (i.i.d.), or that the sequence is a Markov chain, that it is a stationary process, etc. More generally, one can assume that the data is generated by a probability measure that belongs to a certain set \( \mathcal{C} \). In these cases the goal is to have the discrepancy between the predicted and the “true” probabilities to go to zero, if possible, with guarantees on the speed of convergence.

Alternatively, rather than making some assumptions on the data, one can change the goal: the predicted probabilities should be asymptotically as good as those given by the best reference predictor from a certain pre-defined set.

3.3.2. Hypothesis testing

Given a series of observations of \( x_1, \ldots, x_n, \ldots \) generated by some unknown probability measure \( \mu \), the problem is to test a certain given hypothesis \( H_0 \) about \( \mu \) versus a given alternative hypothesis \( H_1 \). There are many different examples of this problem. Perhaps the simplest one is testing a simple hypothesis “\( \mu \) is Bernoulli i.i.d. measure with probability of 0 equals 1/2” versus “\( \mu \) is Bernoulli i.i.d. with the parameter different from 1/2”. More interesting cases include the problems of model verification: for example, testing that \( \mu \) is a Markov chain, versus that it is a stationary ergodic process but not a Markov chain. In the case when we have not one but several series of observations, we may wish to test the hypothesis that they are independent, or that they are generated by the same distribution. Applications of these problems to a more general class of machine learning tasks include the problem of feature selection, the problem of testing that a certain behaviour (such pulling a certain arm of a bandit, or using a certain policy) is better (in terms of achieving some goal, or collecting some rewards) than another behaviour, or than a class of other behaviours.

The problem of hypothesis testing can also be studied in its general formulations: given two (abstract) hypothesis \( H_0 \) and \( H_1 \) about the unknown measure that generates the data, find out whether it is possible to test \( H_0 \) against \( H_1 \) (with confidence), and if yes then how can one do it.

3.3.3. Clustering

The problem of clustering, while being a classical problem of mathematical statistics, belongs to the realm of unsupervised learning. For time series, this problem can be formulated as follows: given several samples \( x^1 = (x^1_1, \ldots, x^1_n), \ldots, x^N = (x^N_1, \ldots, x^N_N) \), we wish group similar objects together. While this is of course not a precise formulation, it can be made precise if we assume that the samples were generated by \( k \) different distributions. Alternatively, one may assume some specific model on the data, leading to different formalizations of the problem.

3.4. Statistical learning

Before detailing some issues of statistical learning, let us remind the definition of a few terms.

**Glossary**

**Machine learning** refers to a system capable of the autonomous acquisition and integration of knowledge. This capacity to learn from experience, analytical observation, and other means, results in a system that can continuously self-improve and thereby offer increased efficiency and effectiveness. (source: http://www.aaai.org/AITopics/html/machine.html AAAI website)

**Statistical learning** is an approach to machine intelligence which is based on statistical modeling of data. With a statistical model in hand, one applies probability theory and decision theory to get an algorithm. This is opposed to using training data merely to select among different algorithms or using heuristics/"common sense” to design an algorithm.
Kernel method

Generally speaking, a kernel function is a function that maps a couple of points to a real value. Typically, this value is a measure of dissimilarity between the two points. Assuming a few properties on it, the kernel function implicitly defines a dot product in some function space. This very nice formal property as well as a bunch of others have ensured a strong appeal for these methods in the last 10 years in the field of function approximation. Many classical algorithms have been “kernelized”, that is, restated in a much more general way than their original formulation. Kernels also implicitly induce the representation of data in a certain “suitable” space where the problem to solve (classification, regression, ...) is expected to be simpler (non-linearity turns to linearity).

The fundamental tools used in SEQUEL come from the field of statistical learning [73]. We briefly present the most important for us to date, namely, kernel-based non parametric function approximation, and non parametric Bayesian models.

3.4.1. Kernel methods for non parametric function approximation

In statistics in general, and applied mathematics, the approximation of a multi-dimensional real function given some samples is a well-known problem (known as either regression, or interpolation, or function approximation, ...). Regressing a function from data is a key ingredient of our research, or to the least, a basic component of most of our algorithms. In the context of sequential learning, we have to regress a function while data samples are being obtained one at a time, while keeping the constraint to be able to predict points at any step along the acquisition process. In sequential decision problems, we typically have to learn a value function, or a policy.

Many methods have been proposed for this purpose. We are looking for suitable ones to cope with the problems we wish to solve. In reinforcement learning, the value function may have areas where the gradient is large; these are areas where the approximation is difficult, while these are also the areas where the accuracy of the approximation should be maximal to obtain a good policy (and where, otherwise, a bad choice of action may imply catastrophic consequences).

We particularly favor non parametric methods since they make quite a few assumptions about the function to learn. In particular, we have strong interests in $l_1$-regularization, and the (kernelized-)LARS algorithm. $l_1$-regularization yields sparse solutions, and the LARS approach produces the whole regularization path very efficiently, which helps solving the regularization parameter tuning problem.

3.4.2. Non–parametric Bayesian models

Numerous problems in signal processing may be solved efficiently by way of a Bayesian approach. The use of Monte-Carlo methods allows us to handle non–linear, as well as non–Gaussian, problems. In their standard form, they require the formulation of probability densities in a parametric form. For instance, it is a common usage to use Gaussian likelihood, because it is handy. However, in some applications such as Bayesian filtering, or blind deconvolution, the choice of a parametric form of the density of the noise is often arbitrary. If this choice is wrong, it may also have dramatic consequences on the estimation quality.

To overcome this shortcoming, one possible approach is to consider that this density must also be estimated from data. A general Bayesian approach then consists in defining a probabilistic space associated with the possible outcomes of the object to be estimated. Applied to density estimation, it means that we need to define a probability measure on the probability density of the noise: such a measure is called a random measure. The classical Bayesian inference procedures can then been used. This approach being by nature non parametric, the associated frame is called Non Parametric Bayesian.

In particular, mixtures of Dirichlet processes [72] provide a very powerful formalism. Dirichlet Processes are a possible random measure and Mixtures of Dirichlet Processes are an extension of well-known finite mixture models. Given a mixture density $f(x|\theta)$, and $G(d\theta) = \sum_{k=1}^{\infty} \omega_k \delta_{\theta_k}(d\theta)$, a Dirichlet process, we define a mixture of Dirichlet processes as:
where $F(x)$ is the density to be estimated. The class of densities that may be written as a mixture of Dirichlet processes is very wide, so that they really fit a very large number of applications.

Given a set of observations, the estimation of the parameters of a mixture of Dirichlet processes is performed by way of a Monte Carlo Markov Chain (MCMC) algorithm. Dirichlet Process Mixtures are also widely used in clustering problems. Once the parameters of a mixture are estimated, they can be interpreted as the parameters of a specific cluster defining a class as well. Dirichlet processes are well known within the machine learning community and its potential in statistical signal processing still need to be developed.

### 3.4.3. Random Finite Sets for multisensor multitarget tracking

In the general multi-sensor multi-target Bayesian framework, an unknown (and possibly varying) number of targets whose states $x_1, \ldots, x_n$ are observed by several sensors which produce a collection of measurements $z_1, \ldots, z_m$ at every time step $k$. Well-known models to this problem are track-based models, such as the joint probability data association (JPDA), or joint multi-target probabilities, such as the joint multi-target probability density. Common difficulties in multi-target tracking arise from the fact that the system state and the collection of measures from sensors are unordered and their size evolve randomly through time. Vector-based algorithms must therefore account for state coordinates exchanges and missing data within an unknown time interval.

Although this approach is very popular and has resulted in many algorithms in the past, it may not the optimal way to tackle the problem, since the state and the data are in fact sets and not vectors.

The random finite set theory provides a powerful framework to deal with these issues. Mahler’s work on finite sets statistics (FISST) provides a mathematical framework to build multi-object densities and derive the Bayesian rules for state prediction and state estimation. Randomness on object number and their states are encapsulated into random finite sets (RFS), namely multi-target (state) sets $X = \{x_1, \ldots, x_n\}$ and multi-sensor (measurement) set $Z_k = \{z_1, \ldots, z_m\}$. The objective is then to propagate the multitarget probability density $f_{k|k}(X|Z(k))$ by using the Bayesian set equations at every time step $k$:

\[
\begin{align*}
    f_{k+1|k}(X|Z^{(k)}) &= \int f_{k+1|k}(X|W)f_{k|k}(W|Z^{(k)})\delta W \\
    f_{k+1|k+1}(X|Z^{(k+1)}) &= \frac{f_{k+1}(Z_{k+1}|X)f_{k+1|k}(X|Z^{(k)})}{\int f_{k+1}(Z_{k+1}|W)f_{k+1|k}(W|Z^{(k)})\delta W}
\end{align*}
\]

where:

- $X = \{x_1, \ldots, x_n\}$ is a multi-target state, i.e. a finite set of elements $x_i$ defined on the single-target space $\mathcal{X}$;\(^2\)
- $Z_{k+1} = \{z_1, \ldots, z_m\}$ is the current multi-sensor observation, i.e. a collection of measures $z_i$ produced at time $k + 1$ by all the sensors;
- $Z^{(k)} = \bigcup_{t\leq k} Z_t$ is the collection of observations up to time $k$;
- $f_{k|k}(W|Z^{(k)})$ is the current multi-target posterior density in state $W$;
- $f_{k+1|k}(X|W)$ is the current multi-target Markov transition density, from state $W$ to state $X$;
- $f_{k+1}(Z|X)$ is the current multi-sensor/multi-target likelihood function.

\(^2\)The state $x_i$ of a target is usually composed of its position, its velocity, etc.
Although equations (5) may seem similar to the classical single-sensor/single-target Bayesian equations, they are generally intractable because of the presence of the set integrals. For, a RFS $\Xi$ is characterized by the family of its Janossy densities $j_{\Xi,1}(x_1), j_{\Xi,2}(x_1,x_2), \ldots$ and not just by one density as it is the case with vectors. Mahler then introduced the PHD, defined on single-target state space. The PHD is the quantity whose integral on any region $S$ is the expected number of targets inside $S$. Mahler proved that the PHD is the first-moment density of the multi-target probability density. Although defined on single-state space $X$, the PHD encapsulates information on both target number and states. The Probability Hypothesis Density is a well-known method for single-sensor multi-target tracking problems in a Bayesian framework, but the extension to the multi-sensor case seems to remain a challenge.
3. Scientific Foundations

3.1. PDEs for particles and fluids

The scientific activity of the project is concerned with PDEs arising from the physical description of particles and fluids. It covers various viewpoints:

- At first, the words “particles and fluids” could simply mean that we are interested independently in models for particles, which can either be considered as individuals (which leads to “N-particle models”, $N$ ranging from 1 to many) or through a statistical description (which leads to kinetic equations) as well as in models for fluids like Euler and Navier-Stokes equations or plasma physics.
- However, many particle systems can also be viewed as a fluid, via a passage from microscopic to macroscopic viewpoint, that is, a hydrodynamic limit.
- Conversely, a fruitful idea to build numerical solvers for hyperbolic conservation laws consists in coming back to a kinetic formulation. This approach has recently motivated the introduction of the so-called kinetic schemes.
- Eventually, one of the main topics of the project is to deal with models of particles interacting with a fluid.

By nature these problems describe multiscale phenomena and one of the major difficulties when studying them lies in the interactions between the various scales: number of particles, size, different time and length scales, coupling...

The originality of the project is to consider a wide spectrum of potential applications. In particular, the word “particles” covers various and very different physical situations, like for instance:

- charged particles: description of semi-conductor devices or plasmas;
- photons, as arising in radiative transfer theory and astrophysics;
- neutrons, as arising in nuclear engineering;
- bacteria, individuals or genes as in models motivated by biology or population dynamics;
- planets or stars as in astrophysics;
- vehicles in traffic flow modelling;
- droplets and bubbles, as in Fluid/Particles Interaction models which arise in the description of sprays and aerosols, smoke and dust, combustion phenomena (aeronautics or engine design), industrial process in metallurgy...

We aim at focusing on all the aspects of the problem:

- Modelling mathematically complex physics requires a deep discussion of the leading phenomena and the role of the physical parameters. With this respect, the asymptotic analysis is a crucial issue, the goal being to derive reduced models which can be solved with a reduced numerical cost but still provide accurate results in the physical situations that are considered.
- The mathematical analysis of the equations provides important qualitative properties of the solutions: well-posedness, stability, smoothness of the solutions, large time behavior... which in turn can motivate the design of numerical methods.
- Eventually, we aim at developing specific numerical methods and performing numerical simulations for these models, in order to validate the theoretical results and shed some light on the physics.
3.2. Interactions of Micro- and Macroscopic Scales, Modelling and Simulations

3.2.1. Reduced Models; Hydrodynamic Limits

In the study of kinetic equations, it is a very usual strategy to compute a hydrodynamic limit, and then to get rid of the velocity variable and replace the kinetic equation by a convection-diffusion model. This kind of derivation is well established, under various forms, and in several fields of applications: neutron transport, semiconductor theory, SHE\(^1\) models... However, several questions of great interest have not yet been solved:

- The computation of the convection-diffusion coefficients of the limit equation, a question which leads to additional difficulties when the small mean free path asymptotics are combined with a homogenization limit. This problem is motivated by applications in nuclear engineering. In this case, the effective coefficients are defined through auxiliary equations and suitable averages of the oscillatory coefficients.

- Some recent works have revealed the formation of singularities in the solutions of some limit convection-diffusion equations, while the original kinetic equation has globally defined solutions. This is due to a coupling in the definition of the convective term with the macroscopic density. This singularity formation is typical of aggregation dynamics. It occurs in models with gravitational forces in astrophysics, and chemotaxis models in biology. Therefore, the natural problems are either to provide a sharp analysis (theoretical and/or based on numerics) of the singularity formation, or to complete the model to avoid such trouble.

- A crucial question for applications is to write models for intermediate regimes, for small but non zero values of the mean free path. Such models are required to remain solvable with a moderate computational cost, and to preserve more features from the kinetic level (as for instance finite speeds of propagation, which is lost with a diffusion equation). An example of such an intermediate model is the moment system obtained by using a closure by Entropy Minimization. We have proved that this model is indeed consistent with the diffusion approximation, and we propose an original scheme to treat these equations numerically. We introduce a relaxation strategy which in turn is naturally amenable to the use of asymptotic preserving splitting methods and anti-diffusive schemes for transport equations that are developed in the team. Therefore, we can compare various limited flux models and discuss on numerics their properties and advantages.

3.2.2. Radiative Transfer Theory

We are interested in the equations of the radiative transfer theory which are motivated by the description of high temperature combustion processes (spacecraft propulsion, reentry problems), space observation, nuclear weapons engineering, or inertial confinement fusion. Such problems can be described by a coupling between kinetic and macroscopic equations that comes from the “collision term”, through energy, or energy-impulsion, exchanges. The hydrodynamic limit yields coupled macroscopic equations, with possibly two distinct temperatures: the temperature of the radiations and the temperature of the material. Taking into account Doppler and relativistic effects adds convective terms, which in turn might give rise to the formation of specific singularities.

The interesting points can be summarized as follows:

- The derivation of the reduced models, based on modeling arguments, is an issue, bearing in mind to describe a complete hierarchy of models;

\(^1\)referring to the standard vocabulary in Physics for Spherical Harmonics Expansion
The coupling induces non-trivial effects on the structure of the hydrodynamics system, which can modify strongly the qualitative properties of the solutions. In particular, the radiative transfer equations might exhibit non-standard shocks profiles, with possible discontinuities. The computation of such discontinuous shock profiles requires a very accurate and non-diffusive numerical scheme for the convective terms. This also leads to the delicate question of the stability of travelling waves solutions.

These topics are the object of a very intense research activity e.g. at the Department of Computational Physics of the Los Alamos National Laboratory as well as at the French Atomic Energy Agency (CEA). We develop alternative numerical methods, based on tricky splitting approaches. When dealing with kinetic models, such methods have to be specifically designed to preserve the asymptotic properties of the model. In this approach, one computes on a time step the evolution of the unknown due to the convective terms, which will be handled by antidiffusive schemes (see the paragraph Conservation Laws below), and on the next time step, we treat source and interaction terms, that can be nonlocal and/or stiff. This leads to a fully explicit scheme which provides accurate results for a cheap numerical cost and which does not require a tedious inversion step as the implicit methods usually do. We are able to treat numerically the full coupling of radiation with hydrodynamics (Euler equations) in the non-equilibrium diffusion regime.

3.2.3. Fluid/Particles Interactions

These models arise in the modelling of disperse suspensions in fluids, say droplet or bubble motion. Their study is motivated by applications to combustion, rocket propulsor engineering, biology, aerosols engineering, or for certain industrial processes. The main effect to take into account is the Stokes drag force, which is proportional to the relative velocity between the particle and the surrounding fluid $F(t, x, v) = \gamma(u(t, x) - v)$. However, modelling remains a major issue in this field; in particular, here are some important questions:

- Complementary effects can be taken into account: the so-called Basset force, or the added mass effect, etc. For instance, when particles flow in a pipe, a phenomenological lift force has been proposed to mimic the tendency of particles to concentrate at the center of the pipe. Even though it is moderate in strength, such a force can have crucial effects on blood flows, or on industrial processes of steel production.

- Up to now, there are only a few contributions on the description of size variations, by coagulation or fragmentation and break-up. However, in practical situations, as for combustion or biology applications, these phenomena cannot be neglected.

- Of course, the coupling with the evolution of the surrounding fluid is a crucial question that leads naturally to problems of asymptotics. Effects of “turbulence”, which roughly means high and fast variations of $u$ on the behavior of the particles, have been analyzed in some simplified situations.

The coupling with the Navier-Stokes or the Euler equations is a privileged subject for SIMPAF. Some asymptotics lead to two-phase flows models, that we are interested in investigating both from a theoretical and numerical point of view. In particular, the effect of an external force (gravitational or centrifugal) can lead to sedimentation profiles that are suspected to be stable; we would like to confirm these heuristics by a thorough numerical and theoretical study. Of course, such investigations require efficient numerical schemes to solve the fluid equations with source terms, which will be detailed in the next sections. To this end, we adapt to this framework the numerical schemes we develop for radiative transfer problems, based on splitting methods and a suitable use of the asymptotic expansion.

3.2.4. Homogenization methods

Homogenization methods aim at replacing a PDE with highly oscillatory coefficients by an effective PDE with smoother coefficients, whose solution captures the averaged behavior of the true oscillatory solution. The effective determination of the homogenized PDE is however not trivial (especially in the nonlinear or/and stochastic cases). Numerical approximations of the solution of the homogenized PDE is the heart of numerical homogenization.
Homogenization methods are used in many application fields. The two applications we are specifically interested in are material sciences (in particular the determination of macroscopic constitutive laws for rubber starting from polymer-chain networks) and nuclear waste storage (in particular the evolution of nuclear wastes in complex storage devices).

The team is interested in qualitative as well as quantitative results, and theoretical as well as numerical results. Challenging questions are mainly related to nonlinear problems (nonlinear elasticity for instance) and stochastic problems (especially regarding quantitative results).

3.3. Charged Particles

3.3.1. Modeling of Plasma Confinement

Plasmas, the fourth state of the matter, play an important role in many branches of physics, such as thermonuclear fusion research, astrophysics and space physics. A plasma is a (partially) ionized gas where charged particles interact via electromagnetic fields. Since the announcement of the creation of the experimental fusion reactor ITER, and with the progress on the ICF\(^2\) program, plasmas and their modelling got a renewed interest.

The nuclear fusion mechanisms result from the strong confinement of charged particles, either by inertial confinement (nuclear fusion reactions are initiated by heating and compressing a target - a pellet that most often contains deuterium and tritium - by the use of intense laser or ion beams) or by the more promising - magnetic fusion confinement. The tokamaks are experimental devices which produce a toroidal magnetic field for confining a plasma.

The description of these phenomena is extremely complex and leads to delicate problems in mathematical analysis and numerical simulation. Actually, plasmas may be described with various levels of detail. The simplest possibility is to treat the plasma as a single fluid governed by the Navier Stokes Equations. A more general description is the two-fluid picture, where the ions and electrons are considered to be distinct. If electric or magnetic fields are present, then the Maxwell equations will be needed to describe them. The coupling of the description of a conductive fluid to electromagnetic fields is known generally as magnetohydrodynamics, or simply MHD.

For some cases the fluid description is not sufficient. Then, the kinetic models may become useful. Kinetic models include information on distortions of the velocity distribution functions with respect to a Maxwell-Boltzmann distribution. This may be important when currents flow, when waves are involved, or when gradients are very steep.

The main mathematical difficulties are therefore linked to the conjunction of the following elements

- These two types of models are strongly nonlinear.
- The unknowns depend on the time and space variables and, in the case of kinetic models, also on the velocity variables. Therefore, we can be led to work with variables of 1 + 3 + 3 dimensions.
- There exist many very different scales (time scale, characteristic length, etc).

The numerical resolution of a complete system of equations, with meshes adapted to the lower scales, leads to prohibitive computational costs both in terms of memory and time. The derivation of new reduced models, corresponding to relevant asymptotic regimes (high magnetic field for example), is therefore a crucial issue. Moreover, very serious efforts must be done on the numerical methods that are used in order to reproduce the typical phenomena. This work depends on the one hand on seriously thinking over the models, the physical parameters, their typical respective scales, and on the other hand over some arguments of asymptotic analysis, which can particularly call on deterministic or random homogeneization.

\(^2\)Inertial Confinement Fusion
3.3.2. Spacecraft Environment

Satellites in geostationary and low Earth orbits naturally evolve in a plasma. This ionized environment induces some perturbations which may lead to many kinds of faults and to the partial or complete loss of a mission. The satellites are covered by dielectric coatings in order to protect them against thermal radiations. Electrons and ions species of the space plasma interact with the external surfaces of the satellite and modify their electrostatic charges. This effect produces potential differences between the satellite surfaces and its electric mass. When the electric field exceeds a certain level, an electrostatic discharge appears. This electric current pulse is able to disrupt the equipments, to damage the external surfaces and even to destroy some electronic components. The plasma may also be created by another source : the electric thrusters. This new propulsion device uses the electric energy supplied by solar arrays to speed up charged species. It is more and more used in satellite industry and has preference over the classical chemical propulsion. Indeed, the latter needs a very large amount of propellant inducing an expensive rocket launch. On the one hand, the electric thrusters allow to significantly reduce the satellite weight. On the other hand, it is necessary to understand their potential impacts on the other systems of the satellite.

This line of research, which continues former works of the team CAIMAN at Sophia Antipolis, was the object of a strong collaboration with the Department Research and Technology of the company Thales Alenia Space. N. Vauchelet proposed several evolutions for the SPARCS code, including parallel procedures. A comparison of different numerical schemes (Finite Volume, Semi-Lagrangian, Back Trajectory) to treat the Vlasov equations of spacecraft charging has been discussed in details. Moreover, the PhD of S. Borghol shows that at LEO or PEO altitudes (instead of the standard GEO framework) the Vlasov equations can be replaced by its hydrodynamic limit. Then the question was: how to define boundary conditions for the macroscopic quantities in order to reproduce phenomena due to the kinetic ones? A partial answer to this question was given for the BGK equation and its hydrodynamic limit: the Euler system.

3.3.3. Effective Energy Dissipation Models for Charged Particles

In models of charge transport, say transport of electrons, a phenomenological friction force is generally introduced, which is proportional to the velocity $v$. Our idea is to go back to a more microscopic framework, with a description of the energy exchanges between the electrons and the surrounding medium. In turn, the dissipation of energy by the medium will lead to an effective friction force. The first contributions only model the transport of a unique particle, and we aim at considering now a plasma, through a statistical description. This yields a Vlasov-Poisson-like model. (More precisely, the kinetic equation is coupled to a finite, or infinite, set of oscillators.) This program requires efforts in modelling and analysis, but the questions are also really challenging for numerics, due, on the one hand, to the large number of degrees of freedom involved in the equation, and on the other hand, to the presence of stiff terms. In this way, we expect to be able to shed light on the range of validity of the Ohm law. Similar considerations also apply for heat transport and the derivation of the Fourier law.

3.4. Simulations of Complex Fluid Flows

3.4.1. Conservation Laws

A major issue in the numerical approximation of systems of conservation laws is the preservation of singularities (shocks, contact discontinuities...). Indeed, the derivatives of the solutions usually blow-up in finite time. The numerical scheme should be able to reproduce this phenomenon with accuracy, i.e. with a minimum number of points, by capturing the profile of the singularity (discontinuity), and by propagating it with the correct velocity. The scheme should also be able to give some insight on the interactions between the possible singularities. Quite recently, new anti-diffusive strategies have been introduced, and successfully used on fluid mechanics problems. We focus on multidimensional situations, as well as on boundary value problems. Since a complete theory is not yet available, the numerical analysis of some prototype systems of

---

3Low Earth Orbit and Polar Earth Orbit respectively
4Geostationary orbit
conservation laws is a good starting point to understand multi-dimensional problems. In particular, a good understanding of the linear case is necessary. This is not achieved yet on the numerical point of view on general meshes. This question is particularly relevant in industrial codes, where one has to solve coupled systems of PDEs involving a complex coupling of different numerical methods, which implies we will have to deal with unstructured meshes. Thus, deriving non-dissipative numerical schemes for transport equations on general meshes is an important issue. Furthermore, transport phenomena are the major reason why a numerical diffusion appears in the simulation of nonlinear hyperbolic conservation laws and contact discontinuities are more subject to this than shocks because of the compressivity of shock waves (this is another reason why we focus at first on linear models).

The next step is to combine non-dissipation with nonlinear stability. An example of such a combination of preservation of sharp shocks and entropy inequalities has been recently proposed for scalar equations and is still at study. It has also been partially done in dimension one for Euler equations.

Of course, there are plenty of applications for the development of such explicit methods for conservation laws. We are particularly interested in simulation of macroscopic models of radiative hydrodynamics, as mentioned above. Another field of application is concerned with polyphasic flows and it is worth specifying that certain numerical methods designed by F. Lagoutière are already used in codes at the CEA for that purpose. We also wish to apply these methods for coagulation-fragmentation problems and for PDEs modelling the growth of tumoral cells; concerning these applications, the capture of the large time state is a particularly important question.

3.4.2. Control in Fluid Mechanics

Flow control techniques are widely used to improve the performances of planes or vehicles, or to drive some internal flows arising for example in combustion chambers. Indeed, they can sensibly reduce energy consumption, noise disturbances, or prevent the flow from undesirable behaviors.

Recently, open and closed active flow control were carried out in order to study the flow behavior over a backward-facing step in a transitional regime. It was done either by a global frequency destabilization at the entry of the domain, or by a local blowing or suction through the lower and upper parts of the step by the use of small jets ([54], E. Creusé, A. Giovannini (IMFT Toulouse) and I. Mortazavi (MC2 INRIA EPI, Bordeaux)). The numerical computations were based on a vortex-in-cell method. Such controls were shown to be efficient in reducing the average recirculation length value, the global flow energy, as well as the global flow enstrophy.

We have now in mind to apply such a strategy on cavity-stent flows, in order to study the effect of passive and/or active control on the average emptying time of the cavity, corresponding to a lot of possible industrial or health applications (combustion, blood circulation in arteries,...).

Passive as well as active control were also performed on the "Ahmed body geometry", which can be considered as a first approximation of a vehicle profile. This work was carried out in collaboration with the EPI INRIA MC2 team in Bordeaux (C.H. Bruneau, I. Mortazavi and D. Depeyras), as well as with Renault car industry (P. Gillieron). We recently combined active and passive control strategies in order to reach efficient results, especially concerning the drag coefficient, for two and three dimensional simulations [46]. We are now interested in the same kind of study, but for a 25° rear-window configuration, for which the 3D-effects are very important and have to be considered in the numerical simulations.

In another field of applications, a work was performed with the TEMPO Laboratory of Valenciennes. The objective of this collaboration was to study the pressure wave generated by high-speed trains entering tunnels in order to improve the shape of the tunnel sections.

3.4.3. Numerical Methods for Viscous Flows

Numerical investigations are very useful to check the behavior of systems of equations which modeling very complicate dynamics. In order to simulate the motion of mixtures of immiscible fluids having different densities, a recent contribution of the team was to develop an hybrid Finite Element / Finite Volume scheme for the resolution of the variable density 2D incompressible Navier-Stokes equations. The main points of this work were to ensure the consistency of the new method [49] as well as its stability for high density ratios [
In order to answer to these questions, we have developed respectively a MATLAB code and a C++ code. In the following of this work, we now have in mind the following objectives:

- To allow the corresponding MATLAB code distribution, in order to promote some further collaborations with other researchers in the domain, and to make the comparison of our results with alternative numerical methods possible. For this objective, a graphics interface was developed as well as some post-processing tools and an accurate documentation. This was the object of the Manuel Bernard internship in the SIMPAF team (Mars 2011 – Sept. 2011). For exemple, the code was already used to study the influence of new strategies for updating LU factors of existing preconditioners. In [48] C. Calgaro et al. address the problem of computing preconditioners for solving linear systems of equations with a sequence of slowly varying matrices. This problem arises in many important applications, for example in computational fluid dynamics, when the equations change only slightly possibly in some parts of the domain.

- To generalize the stability results obtained in [47] for the scalar transport equation to the full 2D Euler system. The target is now to ensure a positivity principle for vertex-based finite volume methods, allowing to simulate some cases involving in particular very low density values density (near vacuum), while maintaining a sufficient accuracy. This work is being developed by Yohan Penel during his post-doc position in the SIMPAF team (Nov. 2010 – Dec. 2011), in collaboration with C. Calgaro, E. Creusé and T. Goudon;

- To modify the already existing C++ code to treat certain more general hydrodynamic models arising in combustion theory, as well as models describing mixing of compressible fluids arising for instance when describing the transport of pollutants. The interesting thing is that this kind of model can be derived by a completely different approach through a kinetic model. Besides, this model presents interesting features, since it is not clear at all whether solutions can be globally defined without smallness assumptions on the data. Then, a numerical investigation is very useful to check what the actual behavior of the system is. Accordingly, our program is two-fold. On the one hand, we develop the density dependent Navier-Stokes code, where the incompressibility condition is replaced by a non standard condition on the velocity field. In particular, if the closure model is the Fick’s law, one obtain the so called Kazhikhov-Smagulov model. The first phenomena we try to reproduce are the powder-snow avalanches. The influence of the characteristic Froude number on the front progression is also investigate. On the other hand, we wish to extend our kinetic asymptotic-based schemes to such problems. This work is being developed in collaboration between C. Calgaro, E. Creusé and T. Goudon (INRIA Sophia Antipolis, team COFFEE).

- In the case of the PhD of M. Ezzoug, co-advised by C. Calgaro and E. Zahrouni (Monastir University, Tunisie), to study numerically and theoretically the influence of a specific stress tensor, introduced for the first time by Korteweg.

- Finally, to prove the convergence of the numerical scheme in order to ensure the theoretically performance of the method. This work started very recently by a collaboration between C. Calgaro, E. Creusé and E. Zahrouni (Monastir University, Tunisie).

### 3.5. A posteriori error estimators for finite element methods

E. Creusé works on a posteriori error estimators for finite element methods, applied to the resolution of several partial differential equations. The objective is to derive useful tools in order to control the global error between the exact solution and the approximated one (reliability of the estimator), and to control the local error leading to adaptive mesh refinement strategies (efficiency of the estimator).

A recent work, in collaboration with S. Nicaise (LAMAV, Valenciennes), was devoted to the derivation of some so-called "reconstruction estimators" based on gradient averaging, in order to provide lower and upper bounds of the error arising from a discontinuous Galerkin approximation of a diffusion problem [55].
At the same time, some equilibrated-type estimators were developed for the Reissner-Mindlin system arising in solid mechanics applications, for conforming and locking-free approximations, in the context of the PhD of E. Verhille.

At last, a collaboration with the "Laboratoire d’électrotechnique et d’électronique de puissance de Lille (L2EP)" began two years ago, to derive a residual-based a posteriori error estimator for the Maxwell system in its vectorial and scalar potential formulation $\mathbf{A}/\Phi$ (PhD of Z. Tang). The objective was to obtain a mathematical rigorous error indicator, in order to couple it with the automatic mesh generator used by EDF for very practical issues.

### 3.6. Numerical analysis of Schrödinger equations

In collaboration with M. Taki (PhLAM laboratory, Lille), Christophe Besse and Guillaume Dujardin are considering dispersive equations modelling the propagation of a laser beam in an optic fiber. They are trying to explain the possible ways of creating rogue waves in the propagation of laser beams. More generally, they are trying to explain which terms in the dispersive Schrödinger-like equations obtained by the physicists allow which physical behaviour of the solutions (e.g. the creation of rogue waves).

In collaboration with G. Reinish (Nice Observatory), Christophe Besse and Guillaume Dujardin are working on the numerical computation of the ground state and the first bound states of the non linear Schrödinger-Poisson system with confining quadratic potential in 2 space dimension modelling quantum dot helium. The goal is to perform after that numerical time stepping methods to simulate the dynamics of the NLSP system and compute accurately some quantities of physical interest as functions of time.
3. Scientific Foundations

3.1. Combinatorial models and algorithms

Our research is driven by biological questions. At the same time, we have in mind to develop well-founded models and algorithms. This is essential to guarantee the universality of our results. Our main background comes from combinatorial discrete models and algorithms. Biological macromolecules are naturally modelled by various types of discrete structures: String, trees, and graphs, etc.

String algorithms is an established research subject of the team. We have been working on spaced seed techniques for several years [22], [23], [24], [33], [35], [27], [26]. The whole technique is implemented and made available in the YASS software for DNA sequence alignment together with the tools implemented to design seeds [28] (see Section 4).

Members of the team have also a strong expertise in text indexing data structures that are widely-used for the analysis of biological sequences because they allow a data set to be stored and queried efficiently. We proposed an optimal neighborhood indexing for protein similarity search [34] and compressed index structures for DNA sequences [37], [36].

Ordered trees and graphs naturally arise when dealing with structural RNAs. Our knowledge in this field allowed us to make several significant contributions to RNA bioinformatics on the past few years. First, we proposed a new method for RNA structure inference, implemented in a program called CARNA, Second, we worked on theoretical models for RNA comparison, which led to substantial advances on tree edit distance algorithms [20], [38], [31], tree models [30], [29] and comparison of arc-annotated sequences [18], [17].

String, trees and graphs are also useful to study genomic rearrangements: Neighborhoods of genes can be modelled by oriented graphs, genomes as permutations, strings or trees.

Nonribosomal peptides representation also uses graphs: Nonribosomal peptides are small molecules that have a branching or cyclic structure. We developed several efficient algorithms to compare NRP molecules represented as non-oriented labeled graphs [19].

3.2. High-performance computing

High-performance computing is another tool that we will use to achieve our goals. It covers several paradigms: grids, single-instruction, multiple-data (SIMD) instructions, graphics cards (GPU). In a near future, processors may offer tens or hundreds of cores with large vector units, combining again several levels of parallelism. Libraries like CUDA and OpenCL also facilitate the use of these manycore processors. This new hardware architecture brings promising opportunities for time-consuming bottlenecks arising in bioinformatics.

3.3. Discrete statistics and probability

At a lower level, our work relies on a basic background on discrete statistics and probability. Probabilistic models indeed naturally appear in many of our research projects. When dealing with large input data sets, it is essential to be able to discriminate between noisy features observed by chance from those that are biologically relevant. The aim here is to introduce a probabilistic model and to use sound statistical methods to assess the significance of some observations about these data. Examples of such observations are the length of a repeated region, the number of occurrences of a motif (DNA or RNA), the free energy of a conserved RNA secondary structure, etc. Moreover, probabilistic models described according to the Bayesian framework allow to bypass, by using MCMC sampling methods, some limitations resulting from complex mathematical integrations over parameter space. Bayesian models and their MCMC sampling allow to approximate probability distributions over parameters and to describe more biologically relevant models. These methods are applied to the genome rearrangement application domain.
3. Scientific Foundations

3.1. Biomechanical Modeling

3.1.1. Biomechanical modeling of solid structures

Soft tissue modeling holds a very important place in medical simulation. A large part of the realism of a simulation, in particular for surgery or laparoscopy simulation, relies upon the ability to describe soft tissue response during the simulated intervention. Several approaches have been proposed over the past ten years to model soft-tissue deformation in real-time (mainly for solid organs), usually based on elasticity theory and a finite element approach to solve the equations. We were among the first to propose such an approach [29], [32] using different computational strategies. Although significant improvements were obtained later on (for instance with the use of co-rotational methods to handle geometrical non-linearities) these works remain of limited clinical use as they rely on linearized constitutive laws.

![Figure 1. Biomechanical models of organs, based on the Finite Element Method and elasticity theory. Left: a model of the liver based on tetrahedral elements and small strain elasticity. Right: several organ models from a patient dataset combined to create a realistic abdominal anatomy.]

An important part of our research is dedicated to the development of new, more accurate models that remain compatible with real-time computation. Such advanced models will not only permit to increase the realism of future training systems, but they will act as a bridge toward the development of patient-specific preoperative planning as well as augmented reality tools for the operating room. Yet, patient-specific planning or per-operative guidance also requires the models to be parametrized with patient-specific biomechanical data. Very little work has been done in this area, in particular when tissue properties need to be measured in vivo non-invasively. New imaging techniques, such as Ultrasound Elastography or Magnetic Resonance Elastography, could be used to this end [28]. We are currently studying the impact of parametrized patient-specific models of the liver in the context of the PASSPORT European project. This will be used to provide information about the deformation, tissue stiffness and tumor location, for various liver pathologies.

3.1.2. Biomechanical modeling of hollow structures

A large number of anatomical structures in the human body are vascularized (brain, liver, heart, kidneys, ...) and recent interventions (such as interventional radiology) rely on the vascular network as a therapeutical pathway. It is therefore essential to model the shape and deformable behavior of blood vessels. This will be
done at two levels. Global deformation of a vascular network: we have demonstrated previously [9] that we could recover the shape of thousands of vessels from medical images by extracting the centerline of each vessel (see Figure 2). The resulting vascular skeleton can be modeled as a deformable (tree) structure which can capture the global aspects of the deformation. More local deformations can then be described by considering now the actual local shape of the vessel. Other structures such as aneurysms, the colon or stomach can also benefit from being modeled as deformable structures. For this we will rely on shell or thin plate theory. We have recently obtained very encouraging results in the context of the Ph.D. thesis of Olivier Comas [31]. Such local and global models of hollow structures will be particularly relevant for planning coil deployment or stent placement, but also in the context of a new laparoscopic technique called NOTES which uses a combination of a flexible endoscope and flexible instruments. Obtaining patient-specific models of vascular structures and associated pathologies remains a challenge from an image processing stand point, and this challenge is even greater once we require these models to be adapted to complex computational strategies. To this extend we will pursue our collaboration with the MAGRIT team at INRIA (through a PhD thesis starting in January 2010) and the Massachusetts General Hospital in Boston.

3.1.3. Blood Flow Simulation

Beyond biomechanical modeling of soft tissues, an essential component of a simulation is the modeling of the functional interactions occurring between the different elements of the anatomy. This involves for instance modeling physiological flows (blood flow, air flow within the lungs, ...). We particularly plan to study the problem of fluid flow in the context of vascular interventions, such as the simulation of three-dimensional turbulent flow around aneurysms to better model coil embolization procedures. Blood flow dynamics is starting to play an increasingly important role in the assessment of vascular pathologies, as well as in the evaluation of pre- and post-operative status. While angiography has been an integral part of interventional radiology procedures for years, it is only recently that detailed analysis of blood flow patterns has been studied as a mean to assess complex procedures, such as coil deployment. A few studies have focused on aneurysm-related hemodynamics before and after endovascular coil embolization. Groden et al. [36] constructed a simple geometrical model to approximate an actual aneurysm, and evaluated the impact of different levels of coil packing on the flow and wall pressure by solving Navier-Stokes equations, while Kakalis et al. [38] relied on patient-specific data to get more realistic flow patterns, and modeled the coiled aneurysm as a porous medium. As these studies aimed at accurate Computational Fluid Dynamics simulation, they rely on commercial software, and the computation times (dozens of hours in general) are incompatible with interactive simulation or even clinical practice. Generally speaking, accuracy and efficiency are two significant pursuits in numerical calculation, but unfortunately very often contradictory.

With the Ph.D. thesis of Yiyi Wei, we have recently started the development of a new technique for accurately computing, in near real-time, the flow of blood within an aneurysm, as well as the interaction between blood and coils. In this approach we rely on the Discrete Exterior Calculus method to obtain an ideal trade-off between accuracy and computational efficiency. Although still at an early stage, these results show that our approach can accurately capture the main characteristics of the complex blood flow patterns in and around an aneurysm. The model also takes into account the influence of the coil on the blood flow within the aneurysm. The main difference between our approach and many other work done by internationally renowned teams (such as REO team at INRIA or the Computer Vision Laboratory at ETH) comes from the importance we place in the computational efficiency of the method. To some extent our approach is similar to what has been done to obtain real-time finite element methods. We are essentially trying to capture the key characteristics of the behavior for a particular application. This is well illustrated by the work we started on flow modeling, which received an award in September 2009 at the selective conference on Medical Image Computing and Computer Assisted Interventions [10]. We will pursue this direction to accurately model the local flow in a closed domain (blood vessel, aneurysm ventricle, ...) and combine it with some of our previous work describing laminar flow across a large number of vessels [43] in order to define boundary conditions for the three-dimensional model.

3.2. Biomechanical Systems
3.2.1. Constraint models and boundary conditions

To accurately model soft tissue deformations, the approach must account for the intrinsic behavior of the target organ, but also for its biomechanical interactions with surrounding tissues or with medical devices. While the biomechanical behavior of important organs (such as the brain or liver) has been well studied, few work exists regarding the mechanical interactions between the anatomical structures. For tissue-tool interactions, most approaches rely on a simple contact models, and rarely account for friction. While this simplification can produce plausible results in the case of an interaction between the end effector of a laparoscopic instrument and the surface of an organ, it is generally an incorrect approximation. As we move towards simulations for planning or rehearsal, accurately modeling contacts will take an increasingly important place. We have recently shown in [33] and [34] that we could compute, in real-time, complex interactions between a coil and an aneurysm, or between a flexible needle and soft-tissues. In laparoscopic surgery, the main challenge lies in the modeling of interactions between anatomical structures rather than between the instruments and the surface of an organ. During the different steps of a procedure organs slides against each other, while respiratory, cardiac and patient motion also generate contacts. Modeling these multiple interactions becomes even more complex when different biomechanical models are used to characterize the various soft tissues of the anatomy. Consequently, our objective is to accurately model resting contacts with friction, in a heterogeneous environment (spring-mass models, finite element models, particle systems, rigid objects, etc.). When different time integration strategies are used, a challenge lies in the computation of contact forces in a way that integrity and stability of the overall simulation are maintained. Our objective is to work on the definition of these various boundary conditions and on new resolution methods for such heterogeneous simulations. In particular we will investigate a simulation process in which each model continues to benefit from its own optimizations while taking into account the mechanical couplings due to interactions between objects.

3.2.2. Vascularized anatomy

From a clinical standpoint, several procedures involve vascularized anatomical structures such as the liver, the kidneys, or the brain. When a therapy needs to be applied on such structures, it is currently possible to perform
a procedure surgically or to use an endovascular approach. This requires to characterize and model the behavior of vessels (arteries and veins) as well as the behavior of soft tissue (in particular the parenchyma). Another challenge of this research will be to model the interactions between the vascular network and the parenchyma where it is embedded. These interactions are key for both laparoscopic surgery and interventional radiology as they allow to describe the motion of the vessels in a vascularized organ during the procedure. This motion is either induced by the surgical manipulation of the parenchymal tissue during surgery or by respiratory, cardiac or patient motion during interventional radiology procedures. From a biomechanical standpoint, capillaries are responsible for the viscoelastic behavior of the vascularized structures, while larger vessels have a direct impact on the overall behavior of the anatomy. In the liver for instance, the apparent stiffness of the organ changes depending on the presence or absence of large vessels. Also, the relatively isotropic nature of the parenchyma is modified around blood vessels. We propose to model the coupling that exists between these two different anatomical structures to account for their respective influence. For this we will initially rely on the work done during the Ph.D. thesis of Christophe Guebert (see [37] for instance) and we will also investigate coupling strategies based on degrees of freedom reduction to reduce the complexity of the problem (and therefore also computation times). Part of this work is already underway in the context of the PASSPORT european project with IRCAD and soft tissue measurements will be performed in collaboration with the biomechanics laboratory at Strasbourg University.

3.2.3. Parallel Computation

Although the past decade has seen a significant increase in complexity and performance of the algorithms used in medical simulation, major improvements are still required to enable patient-specific simulation and planning. Using parallel architectures to push the complexity of simulated environments further is clearly an approach to consider. However, interactive simulations introduce new constraints and evaluation criteria, such as latencies, multiple update frequencies and dynamic adaptation of precision levels, which require further investigation. New parallel architectures, such as multi-cores CPUs, are now ubiquitous as the performances achieved by sequential units (single core CPUs) stopped to regularly improve. At the same time, graphical processors (GPU) offer a massive computing power that is now accessible to non-graphical tasks thanks to new general-purposes API such as CUDA and OpenCL. GPUs are internally parallel processors, exploiting hundreds of computing units. These architectures can be exploited for more ambitious simulations, as we already have demonstrated in a first step by adding support for CUDA within the SOFA framework. Several preliminary results of GPU-based simulations have been obtained, permitting to reach speedup factors (comparing to a single core GPU) ranging from 16x to 55x. Such improvements permit to consider simulations with finer details, or new algorithms modeling biomechanical behaviors more precisely. However, while the fast evolution of parallel architectures is useful to increase the realism of simulations, their varieties (multi-core CPUs, GPUs, clusters, grids) make the design of parallel algorithm challenging. An important effort needs to be made is to minimize the dependency between simulation algorithms and hardware architectures, allowing the reuse of parallelization efforts on all architecture, as well as simultaneously exploiting all available computing resources present in current and future computers. The largest gains could be achieved by combining parallelism and adaptive algorithms. The design and implementation of such a system is a challenging problem, as it is no longer possible to rely on pre-computed repartition of datas and computations. Thus, further research is required in highly adaptive parallel scheduling algorithms, and highly efficient implementation able to handle both large changes in computational loads due to user interactions and multi-level algorithms, and new massively parallel architectures such as GPUs. A direction that we are also investigating is to combine multi-level representations and locally adaptive meshes. Multi-level algorithms are useful not only to speedup computations, but also to describe different characteristics of the deformation at each level. Combined with local change of details of the mesh (possibly using hierarchical structures), the simulation can reach a high level of scalability.
3. Scientific Foundations

3.1. Introduction

In order to cope with our objective, we will consider software paradigms that will help us in our approach at the various levels of our life-cycle of adaptive systems, but also in the tools themselves for their composition. We will also study these paradigms in the middleware and application design in order to extend them and to have a better understanding. These extensions will be formalized as much as possible.

3.1.1. Aspect-Oriented Software Development (AOSD)

In modern software engineering, language constructs are classified according to how they recombine partial solutions for subproblems of a problem decomposition. Some constructs (e.g., methods and classes) recombine partial solutions using classic hierarchical composition. Others recombine the partial solution using what is known as crosscutting (a.k.a. aspectual) composition. With crosscutting composition, two partial solutions (called aspects) are woven into each other in a way that is dictated by so-called pointcut languages. The necessity of crosscutting composition is the main motivation for the AOSD [49], [64] paradigm. The challenge will be first to study new expressive pointcut languages in order to have a better description of composition locations in adaptable software. The second objective will be to extend and to integrate new techniques of weaving at design time, but also at run time in order to compose software safely. The third objective will be to go beyond simple aspects as persistence and logging services. We plan to study complex aspects such as transactions or replication and to control their weaving in order to master the evolution of complex software.

3.1.2. Component-Based Software Engineering (CBSE)

In a post-object world [62], software components [66] are, with other artifacts such as aspects, one of the approaches that aims at overcoming the limitations of objects and providing more flexibility and dynamicity to complex applications. For that, software components present many interesting properties, such as modularity, encapsulation, and composability. Yet, many different component models and frameworks exist. A survey of the literature references more than 20 different models (including the most well-known, such as EJB [48] and CCM [47]), but the exact number is certainly closer to 30. Indeed, each new author proposes a model to address her/his own need related to a particular execution environment (from grid computing to embedded systems) or the technical services (from advanced transactions to real-time properties), which must be provided to the application components. These different component models seldom interoperate and their design and implementation are never founded on a common ground. The research challenge that we identify is to define and implement solutions for adaptive software components. These components will be adaptive in the sense that they will be able to accommodate execution environments of various granularities (from grid computing, to Internet-based applications, to mobile applications, to embedded systems) and incorporate on-demand different technical services. This challenge will be conducted by designing a micro-kernel for software components. This micro-kernel will contain a well-defined set of core concepts, which are at the root of all component models. Several concrete software component models will then be derived from this micro-kernel.

3.1.3. Context-Aware Computing (CAC)

In adaptive systems, the notion of “context” becomes increasingly important. For example, mobile devices sense the environment they are in and react accordingly. This is usually enabled by a set of rules that infer how to react given a certain situation. In the Ambient/Ubiquitous/Pervasive domain¹, CAC is commonly referred to as the new paradigm that employs this idea of context in order to enmesh computing in our daily lives [68]. Many efforts that exist today focus on human-computer interaction based on context. On the one hand,

¹These terms are more or less equivalent.
computational models, middleware, and programming languages are being developed to take the inherent characteristics of multi-scale environments into account, such as connection volatility, ambient resources, etc. An important challenge is to bridge the gap between the domain level and the computational level. The former is concerned with the expected behavior of the system from a user’s viewpoint, such as how and when a system responds to changes in the context, when information can be made public, etc. On the other hand, the computational level deals with the inherent and very stringent hardware phenomena of multi-scale environments. Nevertheless, both levels have to coexist: the computational level needs to be steered by the concepts, behavior and rules which exist at the domain level, whereas the domain needs to adapt to the specificities of the ever changing environment that is monitored and managed by the computational level.

In order to address this challenge, we first intend to investigate representations at the domain level of concepts such as user profile, local positioning information and execution context [78]. Furthermore, a mapping has to be devised between these concepts and generic concepts at the computational level, the latter being as independent as possible from concrete platforms or languages. This mapping has to be bidirectional: the computational level needs to be steered by the concepts, behavior and rules that exist at the domain level, whereas the domain needs to adapt to the particulars of the ever-changing environment that is monitored and managed at the computational level. Furthermore, the mapping has to be dynamic since the changes have to be propagated between the levels at run time. An explicit domain level is not only useful for bridging the aforementioned gap, but also for designing and developing open task-specific languages at the domain level, which allow users to dynamically adapt the behavior of the applications in multi-scale environments in well-defined ways.

We will base the design approach of the future implementation prototype on MDE. The goal of MDE [74] consists of developing, maintaining and evolving complex software systems by raising the level of abstraction from source code to models. The latter is in our case the domain level, which will be connected to the computational level by means of MDE techniques. One added benefit of MDE is that it provides means for managing model inconsistencies.

3.2. Two Research Directions

We propose to follow two research directions to foster software reuse and adaptation. The first direction, that could be coined as the spatial dimension of adaptation, will provide middleware platforms to let applications be adapted to changing execution contexts. The second direction, the so-called temporal dimension of adaptation, will provide concepts and artifacts to let designers specify evolvable applications.

3.2.1. Adaptable Component Frameworks for Middleware

As a cornerstone of next generation software, adaptation is a property which must be present throughout the entire life cycle, from design to execution. We develop then a vision where adaptation is not only a property that is desirable for end-user applications, but also for the middleware platform that executes these applications. Until now, middleware is a rather specialized activity where each new environment forces the development of a corresponding platform, which is specific to the given environment. This has led to a large number of platforms (from Web Services, to EJB, to CORBA, to ad hoc middleware for embedded systems). Although at a high level, solutions for communication interoperability often exist between these platforms, they stay loosely coupled and separated. Furthermore, the concepts which are at the core of these platforms and their architectures are too different to allow, for example, sharing technical services.

The research challenge that we propose here is to define and develop middleware and associated services which could be adapted to a broad range of environments from grid computing, to Internet-based applications, to local networks, to mobile applications on PDA’s and smart phones, to embedded systems. The benefits of that are twofold. First, it enables the easier deployment of mobile applications in different environments by taking advantage of the common ground provided by adaptable middleware. Second, middleware is a rapidly changing domain where new technologies appear frequently. Yet, up to now, each new technological shift has imposed a complete re-development of the middleware. Having a common ground on which middleware is built would help in such transitions by fostering reuse. In terms of industrial output, the impact of these
results will also be helpful for software editors and companies to adapt their products more rapidly to new and emerging middleware technologies.

This research challenge has close links with MDE and product line families. We believe that the added value of our proposal is to cover a more integrated solution: we are not only interested in middleware design with MDE technologies, but we also wish to integrate them with software component technologies and advanced programming techniques, such as AOP. We will then cover a broad spectrum of middleware construction, from design (MDE) to implementation (CBSE) to application development (AOP).

3.2.2. Distributed Application Design for Adaptive Platforms

Considering adaptation in the first design steps of an application allows for its preparation and follow-up during the entire life-cycle. As mentioned previously, some software paradigms help already in the design and the development of adaptable applications. AOSD proposes separation of concerns and weaving of models in order to increase the mastering and the evolution of software. MDE consists of evolving complex software systems by raising the level of abstraction from source code to models. Several programming approaches, such as AOP or reflective approaches, have gained in popularity to implement flexibility. Other approaches, such as CBSE, propose compositional way for reuse and compose sub-systems in the application building. Finally, context-aware programming for mobile environment proposes solutions in order to consider context evolution. Overall, the objective of these approaches is to assist the development of applications that are generic and that can be adapted with respect to the properties of the domain or the context.

The research challenge that we propose to address here is similar to static points of variation in product line families. We plan to study dynamic points of variation in order to take into account adaptation in the first design steps and to match this variation. The first research challenge is the introduction of elements in the modeling phase that allow the specification of evolution related properties. These properties must make it possible to build safe and dynamic software architectures. We wish to express and validate properties in the entire software life cycle. These properties are functional, non-functional, static, behavioral, or even qualitative properties. We also want to be able to check that all the properties are present, that the obtained behavior is the expected one, and that the quality of service is not degraded after the addition or the withdrawal of functionalities. We will base our approach on the definition of contracts expressed in various formalisms (e.g., first order logic, temporal logic, state automata) and we will propose a composition of these contracts.

The second challenge will be to implement design processes that maintain coherence between the various stages of modeling in a MDE approach of the applications, as well as maintaining coherence between the phases of modeling and implementation. To do so, we will design and implement tools that will enable traceability and coherence checking between models, as well as between models and the application at execution time.

Finally, we will introduce context information in the development process. At the modeling level, we will represent concepts, behavior and rules of adaptive systems to express adaptation abstraction. These models will be dynamic and connected to implementation levels at the computational level and they will consider context knowledge. The goal is to bridge the gap between the computational level and the domain level in adaptive systems by synchronization of models and implementations, but also by representation of such common knowledge.
3. Scientific Foundations

3.1. Embedded Operating Systems

We focus our activities on “adaptability” and on “connectivity” of embedded platforms dedicated to POPS. From then on, our researches have evolved around the smart card. In fact, in the nineties (birth date of POPS research group) smart card was the only valuable and industrially deployed POPS. Smart card integration in database management systems, smart card integration in Corba (using the Card Object Adapter), open platform for smart card (the first smart card virtual machine), have been milestones of the POPS research. More recently, we have focused our attention (according to our industrial inputs) on embedded operating system techniques, enabling “on-card” type checking and bytecode compression. Today, smart card manufacturers and other emerging POPS manufacturers have to deal with new technological ‘lock-in’ inside and outside the mobile object. Dedicated operating systems are now powerful enough to run dynamically downloaded applications in a safe way. Typically, Java Card loads and runs a Java-like bytecode. Nevertheless, “Java-like” means “non-Java”. Embedded virtual machines do not support standard abstractions. And so, Java applications cannot be deployed in a limited embedded system. On the other hand, embedded applications do not limit their needs to the Java APIs. To overcome these limitations, we will focus on two complementary studies:

1. Firstly we study a new architectural way to embed a Java virtual machine. Conventional virtual machines are not operating systems but they overlap the abstractions proposed by the system. We plan to define a Java virtual machine designed to be the operating system (the virtual machine will manage the hardware itself).

2. Java is one of the possible hardware abstractions. However different applications require different abstractions: file-system, database systems, and so on. Camille OS is a smart card Exo-kernel enabling the download of different hardware abstractions in a safe way. In this way Camille ensures POPS “adaptability” to the applications requirements. Nevertheless some critical system extensions (enhanced IO protocols for example) need additional guaranties: real-time properties and hardware resources control.

3.2. Mobile Networking

POPS also have a non-conventional communication interface. Due to their mobility, they have transient and unpredictable communications with other entities. This fact motivates our focusing on the ad hoc network communication model which is the most flexible model.

Indeed wireless ad hoc networks [51], [46], [47], [44] encompass a wide range of self-organized network types, including sensor, mobile ad hoc, personal area, and rooftop/mesh networks. The design of data communication techniques in multi-hop ad hoc networks comprises challenges at all layers of communication: physical, medium access control (MAC), network, transport and application layers. This research project concentrates on the network layer. The network layer problems can be divided into three groups: data communication, service access, and topology control problems. Data communication problems include routing, quality-of-service routing, geocasting, multicasting, and broadcasting. The protocols need to minimize the communication overhead (since bandwidth in wireless communication is typically limited) and the power consumption of battery operated POPS. In service access problems, such as multi-hop wireless Internet (hybrid network, see Fig. 4), the goal is to provide or receive services from a fixed infrastructure with other hosts serving as relays if necessary. Topology control problems include neighbor discovery problems (detecting neighboring nodes located within transmission radius) and network organization problems (deciding what communication links to establish with neighboring nodes, operating sleeping period and adjusting transmission radii). Secure routing faces the following challenges: node selfishness, threats using modification...
of routing information, misrepresenting identity, fabrication of routing messages by one node, or between two malicious nodes (wormhole attack), and self-organized public-key management and authentication services. The main paradigm shift is to apply localized (or greedy) schemes as opposed to existing protocols requiring global information. Localized algorithms are distributed algorithms where simple local node behavior achieves a desired global objective. Localized protocols provide scalable solutions, that is, solutions for wireless networks with an arbitrary number of nodes, which is one of the main goals of this research project.

![Diagram of network types](image)

**Figure 4. From wireless network to hybrid networks.**
3. Scientific Foundations

3.1. Software Reengineering

Strong coupling among the parts of an application severely hampers its evolution. Therefore, it is crucial to answer the following questions: How to support the substitution of certain parts while limiting the impact on others? How to identify reusable parts? How to modularize an object-oriented application?

Having good classes does not imply a good application layering, absence of cycles between packages and reuse of well-identified parts. Which notion of cohesion makes sense in presence of late-binding and frameworks? Indeed, frameworks define a context that can be extended by subclassing or composition: in this case, packages can have a low cohesion without being a problem for evolution. How to obtain algorithms that can be used on real cases? Which criteria should we select for a given remodularization?

We plan to enrich Moose, our reengineering environment, with a new set of analyses [51], [49]. We decompose our approach in three main and potentially overlapping steps:

1. Tools for understanding applications at large: packages/modules,
2. Remodularization analyses, and
3. Software Quality and Open DashBoard.

3.1.1. Tools for understanding applications at large: packages/modules

**Context and Problems.** As we are going to design and evaluate several algorithms and analyses to remodularize applications, we need a way to understand and assess the results we will obtain. Our experience on real application analyses taught us that analyses tend to produce a huge amount of data that we should understand and correlate to the original source code situation [50]. The problem is that understanding large systems is already difficult [102], [74], [76], [69], but in our case we need to understand an existing system and the results of our analysis. Parallelism between software programs and cities is commonly used to reason about evolution [69], [103]. While interesting, this metaphor does not scale because location of houses does not have any semantics information related to the connection between classes. A notion of radar has also been proposed [46], but this mechanism suffers from the same problem.

Therefore, there is a definitive need to have ways to support the understanding of large applications at the level of their structure.

**Research Agenda.** We are going to study the problems raised by the understanding of applications at the larger level of granularity such as packages/modules. We will develop a set of conceptual tools to support this understanding. These tools will certainly be visual such as the Distribution Map Software visualization [50] or based on the definition of new metrics taking into account the complexity of packages. Such a step is really crucial as a support for the remodularization analyses that we want to perform. The following tasks are currently ongoing:

**MoQam.** The Qualixo model has been originally implemented on top of the Java platform. An implementation of this model, named MoQam (Moose Quality Assessment Model), is under development in the Moose open-source and free reengineering environment. A first experiment has been conducted [70]. Exporters from Moose to the Squale software are under development.

**Cohesion Metric Assessment.** We are assessing the metrics and practices used originally in the Qualixo model. We are also compiling a number of metrics for cohesion and coupling assessment. We want to assess for each of these metrics their relevance in a software quality setting.
DSM. Dependency Structure Matrix (DSM), an approach developed in the context of process optimization, has been successfully applied to identify software dependencies among packages and subsystems. A number of algorithms help organizing the matrix in a form that reflects the architecture and highlights patterns and problematic dependencies between subsystems. However, the existing DSM implementations often miss important information in their visualization to fully support a reengineering effort. We plan to enrich them to improve their usefulness to assess system general structure.

### 3.1.2. Remodularization analyses

**Context and Problems.** It is a well-known practice to layer applications with bottom layers being more stable than top layers [80]. Until now, few works have attempted to identify layers in practice: Mudpie [101] is a first cut at identifying cycles between packages as well as package groups potentially representing layers. DSM (dependency structure matrix) [100], [95] seems to be adapted for such a task but there is no serious empirical experience that validates this claim. From the side of remodularization algorithms, many were defined for procedural languages [68]. However, object-oriented programming languages bring some specific problems linked with late-binding and the fact that a package does not have to be systematically cohesive since it can be an extension of another one [104], [54].

Some approaches based on Formal Concept Analysis [99] show that such an analysis can be used to identify modules. However, the presented example is small and not representative of real code. Other clustering algorithms [64], [65] have been proposed to identify modules [75], [86]. Once again, the specific characteristics of object-oriented programming are not taken into account. This is a challenge since object-oriented programming tends to scatter classes definitions over multiple packages and inheritance hierarchies. In addition, the existing algorithms or analyses often only work on toy applications. In the context of real applications, other constraints exist such as the least perturbation of the code, minimizing the hierarchy changes, paying attention to code ownership, layers, or library minimization. The approach will have to take into account these aspects.

Many different software metrics exist in the literature [73], [56], [61] such as the McCabe complexity metrics [81]. In the more specific case of object-oriented programming, assessing cohesion and coupling have been the focus of several metrics. However, their success is rather questionable. For example, LCOM [43] has been highly criticized [55], [63], [62], [30], [39], [40]. Other approaches have been proposed such as RFC and CBO [43] to assess coupling between classes. However, many other metrics have not been the subject of careful analysis such as Data Abstraction Coupling (DAC) and Message Passing Coupling (MPC) [32], or some of metrics are not clearly specified (MCX, CCO, CCP, CRE) [73]. New cohesion measures were proposed [77], [89] taking into account class usage.

**Research Agenda.** We will work on the following items:

- Characterization of “good” modularization. Any remodularization effort must use a quality function that allows the programmer to compare two possible decompositions of the system and choose which one represents a more desirable modularization. Remodularization consists in trying to maximize such a function. The typical function used by most researchers is some measure of cohesion/coupling. However, manual system modularization may rely on many different considerations: implemented functionalities, historical considerations, clients or markets served, ...We want to evaluate various modularization quality functions against existing modularizations to identify their respective strengths and weaknesses.

- Cohesion and coupling metric evaluation and definition. Chidamber’s well-known cohesion metric named LCOM has been strongly criticized [55], [63], [62], [40]. However, the solutions rarely take into account that a class is an incremental definition and, as such, can exist in a several packages at once. For example, LCOM* flattens inheritance to determine the cohesion of a class. In addition, these metrics are not adapted to packages. We will thus work on the assessment of existing cohesion metrics for classes, defining new ones if necessary for packages, and assess coupling metrics as well [30], [39]. This work is also related to the notion of software quality treated below.
Build an empirical validation of DSM and enhancements. We want to assess Dependency Structure Matrix (DSM) to support remodularization. DSM is good to identify cyclic dependencies. Now we want to know if we can identify misplaced classes among groups of packages working as layers. For this purpose we will perform controlled experiments and in a second period apply DSM on one of the selected case studies. Based on these results, we will propose enhancements to the approach.

Layer identification. We want to propose an approach to identify layers based on a semi-automatic classification of package and class interrelationships that they contain. However, taking into account the wish or knowledge of the designer or maintainer should be supported. We will try to apply different algorithms and adapt them to the specific context of object-oriented programming [68].

3.1.3. Software Quality

Companies often look for the assessment of their software quality. Several models of software quality have been proposed: J.A. McCall [82] with his Factor-Criteria-Metrics has identified more than 50 candidate factors that may be used to assess software quality. Among those factors, only 11 were retained. Each of those has been characterized by 23 criteria that represent the internal project quality view. This approach is not easily used because of the high number of metrics —more than 300, some of which are not automatically computed. In an effort of conformance, the ISO (International Standardization Organization) and the IEC (International Electronical Commission) jointly defined the ISO 9126 norm in 1999. This norm, currently being restructured, will be composed of 4 parts: quality model (ISO 9126-1), external metrology (ISO 9126-2), Internal metrology (ISO 9126-3), Usage quality of metrology (ISO 9126-4). There is also a body of work focusing on design evaluation as quality criteria [83], [91], [87] and new quality models: QMOOD is for example a hierarchical quality model which proposes to link directly quality criteria to software metrics based on object-oriented software metrics [31] while other works focus on linking different high level criteria with software code metrics [78], [79].

Research Agenda. Since software quality is fuzzy by definition and a lot of parameters should be taken into account we consider that defining precisely a unique notion of software quality is definitively a Graal in the realm of software engineering. The question is still relevant and important. We plan to work on the two following items in the context of the Squale project in contact with the Qualixo company:

Quality Model. We want to study the existing quality models and develop in particular models that take into account (1) the possible overlaps in the source of information —it is important to know whether a model measures the same aspects several times, using different metrics (2) the combination of indicators —often, software quality models happily combine metrics, but at the price of losing the explicit relationships between the indicator contributions.

3.2. Language Constructs for Modular Design

While the previous axis focuses on how to help remodularizing existing software, this second research axis aims at providing new language constructs to build more flexible and recomposable software. We will build on our work on traits [97], [52] and classboxes [33] but also start to work on new areas such as security in dynamic languages. We will work on the following points: (1) Traits: behavioral units and (2) Modularization as a support for security.

3.2.1. Traits-based program reuse

Context and Problems. Inheritance is well-known and accepted as a mechanism for reuse in object-oriented languages. Unfortunately, due to the coarse granularity of inheritance, it may be difficult to decompose an application into an optimal class hierarchy that maximizes software reuse. Existing schemes based on single inheritance, multiple inheritance, or mixins, all pose numerous problems for reuse.
To overcome these problems, we designed a new composition mechanism called Traits \cite{97}, \cite{52}. Traits are pure units of behavior that can be composed to form classes or other traits. The trait composition mechanism is an alternative to multiple or mixin inheritance in which the composer has full control over the trait composition. The result enables more reuse than single inheritance without introducing the drawbacks of multiple or mixin inheritance. Several extensions of the model have been proposed \cite{48}, \cite{90}, \cite{34}, \cite{53} and several type systems were defined \cite{57}, \cite{98}, \cite{92}, \cite{72}.

Traits are reusable building blocks that can be explicitly composed to share methods across unrelated class hierarchies. In their original form, traits do not contain state and cannot express visibility control for methods. Two extensions, stateful traits and freezable traits, have been proposed to overcome these limitations. However, these extensions are complex and not simple to implement.

**Research Agenda: Towards a pure trait language.** We plan distinct actions: (1) a large application of traits, (2) assessment of the existing trait models and (3) bootstrapping a pure trait language.

- To evaluate the expressiveness of traits, some hierarchies were refactored, showing code reuse \cite{37}. However, such large refactorings, while valuable, may not exhibit all possible composition problems, since the hierarchies were previously expressed using single inheritance and following certain patterns. We want to redesign from scratch the collection library of Smalltalk (or part of it). Such a redesign should on the one hand demonstrate the added value of traits on a real large and redesigned library and on the other hand foster new ideas for the bootstrapping of a pure trait-based language.

  In particular we want to reconsider the different models proposed (stateless \cite{52}, stateful \cite{35}, and freezable \cite{53}) and their operators. We will compare these models by (1) implementing a trait-based collection hierarchy, (2) analyzing several existing applications that exhibit the need for traits—the Tweak UI library used in Sophie\textsuperscript{2} and OpenCroquet\textsuperscript{3}, the Icalendar implementation of Seaside\textsuperscript{4}. Traits may be flattened \cite{88}. This is a fundamental property that confers to traits their simplicity and expressiveness over Eiffel’s multiple inheritance. Keeping these aspects is one of our priority in forthcoming enhancements of traits.

- Alternative trait models. This work revisits the problem of adding state and visibility control to traits. Rather than extending the original trait model with additional operations, we use a fundamentally different approach by allowing traits to be lexically nested within other modules. This enables traits to express (shared) state and visibility control by hiding variables or methods in their lexical scope. Although the traits’ “flattening property” no longer holds when they can be lexically nested, the combination of traits with lexical nesting results in a simple and more expressive trait model. We formally specify the operational semantics of this combination. Lexically nested traits are fully implemented in AmbientTalk, where they are used among others in the development of a Morphic-like UI framework.

  We want to evaluate how inheritance can be replaced by traits to form a new object model. For this purpose we will design a minimal reflective kernel, inspired first from ObjVlisp but with trait composition \cite{45}, then from Smalltalk \cite{59}.

### 3.2.2. Reconciling Dynamic Languages and Security

**Context and Problems.** More and more applications require dynamic behavior such as modification of their own execution (often implemented using reflective features \cite{67}). For example, F-script allows one to script Cocoa Mac-OS X applications and Lua is used in Adobe Photoshop. Now in addition more and more applications are updated on the fly, potentially loading untrusted code or simply broken code. Bytecode checking and static code verification are used to enable security, however such approaches do not really work in presence of dynamic languages and reflective features. Therefore there is a tension between a need for flexibility and for security —here, by security we mean a mix between confidentiality and integrity.

\textsuperscript{2}http://www.sophieproject.org/
\textsuperscript{3}http://www.opencroquet.org
\textsuperscript{4}http://www.seaside.st
Research Agenda: A secure dynamic and reflective language. To solve this tension, we will work on Sure, a language where security is provided by construction: as an example, if the language does not offer field access and its reflective facilities are controlled, then the possibility to access and modify private data is controlled. In this context, layering and modularizing the meta-level [38], as well as controlling the access to reflective features [41], [42] are important challenges. We plan to:

- Study the security abstractions available in erights\(^5\) [85], [84], Java as well as classLoader strategies [71], [60].
- Categorize the different reflective features of languages such as CLOS [66], Python and Smalltalk [93] and identify suitable security mechanisms and infrastructure [58].
- Assess different security models (access rights, capabilities [94]...) and identify the ones adapted to our context as well as different access and right propagation.
- Define a language based on
  - the decomposition and restructuring of the reflective features [38],
  - the use encapsulation policies as a basis to restrict the interfaces of the controlled objects [96],
  - the definition of method modifiers to support privacy in the context of dynamic languages.

An open question is whether, instead of providing restricted interfaces, we could use traits to grant additional behavior to specific instances: without trait application, the instances would only exhibit default public behavior, but with additional traits applied, the instances would get extra behavior. We will develop Sure, a modular extension of the reflective kernel of Smalltalk (since it is one of the languages offering the largest set of reflective features such as pointer swapping, class changing, class definition...) [93].

\(^5\) http://www.erights.org
3. Scientific Foundations

3.1. Human-Computer Interaction

The scientific approach that we follow considers user interfaces as means, not an end: our focus is not on interfaces, but on interaction considered as a phenomenon between a person and a computing system [27]. We observe this phenomenon in order to understand it, i.e. describe it and possibly explain it, and we look for ways to significantly improve it. HCI borrows its methods from various disciplines, including Computer Science, Psychology, Ethnography and Design. Participatory design methods can help determine users’ problems and needs and generate new ideas, for example [34]. Rapid and iterative prototyping techniques allow to decide between alternative solutions [28]. Controlled studies based on experimental or quasi-experimental designs can then be used to evaluate the chosen solutions [36]. One of the main difficulties of HCI research is the doubly changing nature of the studied phenomenon: people can both adapt to the system and at the same time adapt it for their own specific purposes [33]. As these purposes are usually difficult to anticipate, we regularly create new versions of the systems we develop to take into account new theoretical and empirical knowledge. We also seek to integrate this knowledge in theoretical frameworks and software tools to disseminate it.

3.2. Numerical and algorithmic real-time gesture analysis

Whatever is the interface, user provides some curves, defined over time, to the application. The curves constitute a gesture (positionnal information, yet may also include pressure). Depending on the hardware input, such a gesture may be either continuous (e.g. data-glove), or not (e.g. multi-touch screens). User gesture can be multi-variate (several fingers captured at the same time, combined into a single gesture, possibly involving two hands, maybe more in the context of co-located collaboration), that we would like, at higher-level, to be structured in time from simple elements in order to create specific command combinations.

One of the scientific fundations of the research project is an algorithmic and numerical study of gesture, which we classify into three points:

- **clustering**, that takes into account intrinsic structure of gesture (multi-finger/multi-hand/multi-user aspects), as a lower-level treatment for further use of gesture by application;

- **recognition**, that identifies some semantic from gesture, that can be further used for application control (as command input). We consider in this topic multi-finger gestures, two-handed gestures, gesture for collaboration, on which very few has been done so far to our knowledge. On the contrary, in the case of single gesture case (i.e. one single point moving over time in a continuous manner), numerous studies have been proposed in the current literature, and interestingly, are of interest in several communities: HMM [37], Dynamic Time Warping [39] are well-known methods for computer-vision community, and hand-writing recognition. In the computer graphics community, statistical classification using geometric descriptors has previously been used [35]; in the Human-Computer interaction community, some simple (and easy to implement) methods have been proposed, that provide a very good compromise between technical complexity and practical efficiency [38].

- **mapping to application**, that studies how to link gesture inputs to application. This ranges from transfer function that is classically involved in pointing tasks [31], to the question to know how to link gesture analysis and recognition to the algorithmic of application content, with specific reference examples.

We ground our activity on the topic of numerical algorithm, expertise that has been previously achieved by team members in the physical simulation community (within which we think that aspects such as elastic deformation energies evaluation, simulation of rigid bodies composed of unstructured particles, constraint-based animation... will bring up interesting and novel insights within HCI community).
3.3. Design and control of haptic devices

Our scientific approach in the design and control of haptic devices is focused on the interaction forces between the user and the device. We search of controlling them, as precisely as possible. This leads to different designs compared to other systems which control the deformation instead. The research is carried out in three steps:

- **identification**: we measure the forces which occur during the exploration of a real object, for example a surface for tactile purposes. We then analyze the record to deduce the key components – on user’s point of view – of the interaction forces.

- **design**: we propose new designs of haptic devices, based on our knowledge of the key components of the interaction forces. For example, coupling tactile and kinesthetic feedback is a promising design to achieve a good simulation of actual surfaces. Our goal is to find designs which leads to compact systems, and which can stand close to a computer in a desktop environment.

- **control**: we have to supply the device with the good electrical conditions to accurately output the good forces.
3. Scientific Foundations

3.1. Modeling XML document transformations

Participants: Guillaume Bagan, Adrien Boiret, Iovka Boneva, Angela Bonifati, Anne-Cécile Caron, Benoît Groz, Joachim Niehren, Yves Roos, Sławek Staworko, Sophie Tison, Antoine Ndione, Tom Sebastian.

XML document transformations can be defined in W3C standards languages XQuery or XSLT. Programming XML transformations in these languages is often difficult and error prone even if the schemata of input and output documents are known. Advanced programming experience and considerable programming time may be necessary, that are not available in Web services or similar scenarios.

Alternative programming language for defining XML transformations have been proposed by the programming language community, for instance XDuce [38], Xtatic [36], [41], and CDuce [27], [28], [29]. The type systems of these languages simplify the programming tasks considerably. But of course, they don’t solve the general difficulty in programming XML transformations manually.

Languages for defining node selection queries arise as sub-language of all XML transformation languages. The W3C standards use XPath for defining monadic queries, while XDuce and CDuce rely on regular queries defined by regular pattern equivalent to tree automata. Indeed, it is natural to look at node selection as a simple form of tree transformation. Monadic node selection queries correspond to deterministic transformations that annotate all selected nodes positively and all others negatively. N-ary node selection queries become non-deterministic transformations, yielding trees annotated by Boolean vectors.

After extensive studies of node selection queries in trees (in XPath and many other languages) the XML community has started more recently to formally investigate XML tree transformations. The expressiveness and complexity of XQuery are studied in [40], [49]. Type preservation is another problem, i.e., whether all trees of the input type get transformed into the output type, or vice versa, whether the inverse image of the output type is contained in the input type [44], [42].

The automata community usually approaches tree transformations by tree transducers [34], i.e., tree automata producing output structure. Macro tree transducers, for instance, have been proposed recently for defining XML transformations [42]. From the view point of logic, tree transducers have been studied for MSO definability [35].

3.2. Machine learning for XML document transformations

Participants: Jean Decoster, Jean-Baptiste Faddoul, Rémi Gilleron, Grégoire Laurence, Aurélien Lemay, Joachim Niehren, Sławek Staworko, Marc Tommasi, Fabien Torre, Gemma Garriga, Antonino Freno, Thomas Ricatte, Mikaela Keller.

Automatic or semi-automatic tools for inferring tree transformations are needed for information extraction. Annotated examples may support the learning process. The learning target will be models of XML tree transformations specified in some of the languages discussed above.

Grammatical inference is commonly used to learn languages from examples and can be applied to learn transductions. Previous work on grammatical inference for transducers remains limited to the case of strings [30], [45]. For the tree case, so far only very basic tree transducers have been shown to be learnable, by previous work of the Mostrare project. These are node selecting tree transducer (NSTTs) which preserve the structure of trees while relabeling their nodes deterministically.

Statistical inference is most appropriate for dealing with uncertain or noisy data. It is generally useful for information extraction from textual data given that current text understanding tools are still very much limited. XML transformations with noisy input data typically arise in data integration tasks, as for instance when converting PDF into XML.
Stochastic tree transducers have been studied in the context of natural language processing [37], [39]. A set of pairs of input and output trees defines a relation that can be represented by a 2-tape automaton called a *stochastic finite-state transducer* (SFST). A major problem consists in estimating the parameters of such transducer. SFST training algorithms are lacking so far [33].

Probabilistic context free grammars (pCFGs) [43] are used in the context of PDF to XML conversion [31]. In the first step, a labeling procedure of leaves of the input document by labels of the output DTD is learned. In the second step, given a CFG as a generative model of output documents, probabilities are learned. Such two steps approaches are in competition with one step approaches estimating conditional probabilities directly.

A popular non generative model for information extraction is *conditional random fields* (CRF, see a survey [46]). One main advantage of CRF is to take into account long distance dependencies in the observed data. CRF have been defined for general graphs but have mainly been applied to sequences, thus CRF for XML trees should be investigated.

So called *structured output* has recently become a research topic in machine learning [48], [47]. It aims at extending the classical categorization task, which consists to associate one or some labels to each input example, in order to handle structured output labels such as trees. Applicability of structured output learning algorithms remains to be asserted for real tasks such as XML transformations.