Activity Report 2013

Section Scientific Foundations

<div align="center">

ADAM Project-Team

</div>

3. Research Program

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 [60], [68] 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 [65], software components [69] 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 [59] and CCM [58]), 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 domain [1], CAC is commonly referred to as the new paradigm that employs this idea of context in order to enmesh computing in our daily lives [72]. Many efforts that exist today focus on human-computer interaction based on context. On the one hand, computational models, middleware, and programming languages are being developed to take

---

[1]These terms are more or less equivalent.

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 [83]. 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 Model Driven Engineering (MDE). The goal of MDE [80] 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.

ATEAMS Project-Team

3. Research Program

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, transform it and generate it. Each analysis or transformation begins with fact extraction. After that we may analyze specific software systems or large bodies of software systems. Our goal is to improve software systems by understanding and resolving the causes of software complexity. The aoproach is captured in the EASY acronym: Extract Analyze SYnthesize. The first step is to extract facts from source code. These facts are then enriched and refined in an analysis phase. Finally the result is synthesized in the form of transformed or generated source code, a metrics report, a visualization or some other output artifact.

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 are exploring 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 closure) and the performance problems encountered, this approach has not seen wider use.

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 existing (legacy) programming languages, which is highly relevant for experimental research and validation.

Extracted facts are often refined, enriched and queried in the analysis phase. We propose to use a relational formalization of the facts. That is, facts are represented as sets of tuples, which can then be queried using relational algebra operators (e.g., domain, transitive closure, projection, composition etc.). This relational representation facilitates dealing with graphs, which are commonly needed during program analysis, for instance when processing control-flow or data-flow graphs. The Rascal language integrates a relational sub-language by providing comprehensions over different kinds of data types, in combination with powerful pattern matching and built-in primitives for computing (transitive/reflexive) closures and fixpoint computations (equation solving).

3.2.1. Goals

The main goal is to replace labour-intensive manual programming of fact extractors by automatic generation based on concise and formal specification. 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, yet flexible enough to be customized to the analysis case at hand. 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. Refactoring and Transformation

The second 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.

3.3.1. 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.

3.4. 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.

3.4.1. 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. As an end-result we envision Rascal to be a one-stop shop for source code analysis, transformation, generation and visualization.

3.5. Domain-specific Languages

Our final goal is centered around Domain-specific languages (DSLs), which are software languages tailored to a specific problem domain. DSLs can provide orders of magnitude improvement in terms of software quality and productivity. However, the implementation of DSLs is challenging and requires not only thorough knowledge of the problem domain (e.g., finance, digital forensics, insurance, auditing etc.), but also knowledge of language implementation (e.g., parsing, compilation, type checking etc.). Tools for language implementation have been around since the archetypical parser generator YACC. However, many of such tools are characterized by high learning curves, lack of integration of language implementation facets, and lead to implementations that are hard to maintain. This line of research focuses on two topics: improve the practice and experience of DSL implementation, and evaluate the success of DSLs in industrial practice.

Language workbenches [24] are integrated environments to facilitate the development of all aspects of DSLs. This includes IDE support (e.g., syntax coloring, outlining, reference resolving etc.) for the defined languages. Rascal can be seen as a language workbench that focuses on flexibility, programmability and modularity. DSL implementation is, in essence, an instance of source code analysis and transformation. As a result, Rascal's features for fact extraction, analysis, tree traversal and synthesis are an excellent fit for this area. An important aspect in this line of research is bringing the IDE closer to the source code. This will involve investigation of heterogeneous representations of source code, by integrating graphical, tabular or forms-based user interface elements. As a result, we propose Rascal as a feature-rich workbench for model-driven software development.

The second component of this research is concerned with evaluating DSLs in industrial contexts. This means that DSLs constructed using Rascal will be applied in real-life environments so that expected improvements in quality, performance, or productivity can be observed. We already have experience with this in the domain of digital forensics, computational auditing and games.

3.5.1. *Goals*

The goal of this research topic is to improve the practice of DSL-based software development through language design and tool support. A primary focus is to extend the IDE support provided by Rascal, and to facilitate incremental, and iterative design of DSLs. The latter is supported by new (meta-)language constructs for extending existing language implementations. This will require research into extensible programming and composition of compilers, interpeters and type checkers. Finally, a DSL is never an island: it will have to integrate with (third-party) source code, such as host language, libraries, runtime systems etc. This leads to the vision of multi-lingual programming environments [22]. .

BONSAI Project-Team

3. Research Program

3.1. Combinatorial discrete models and algorithms

Our research is driven by biological questions. At the same time, we have in mind to develop well-founded models and efficient algorithms. Biological macromolecules are naturally modelled by various types of discrete structures: String, trees, and graphs. String algorithms is an established research subject of the team. We have been working on spaced seed techniques for several years [13], [20], [22], [16], [15]. Members of the team have also a strong expertise in text indexing and compressed index data structures [21], [24], [23]. Such methods are widely-used for the analysis of biological sequences because they allow a data set to be stored and queried efficiently. Ordered trees and graphs naturally arise when dealing with structures of molecules, such as RNAs [25], [19], [18], [17], [11] or non-ribosomal peptides [12]. The underlying questions are: how to compare molecules at structural level, how to search for structural patterns ? 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.

High-performance computing is another tool that we use to achieve our goals. It covers several paradigms: grids, single-instruction, multiple-data (SIMD) instructions or manycore processors such as graphics cards (GPU). For example, libraries like CUDA and OpenCL also facilitate the use of these manycore processors. These hardware architectures bring promising opportunities for time-consuming bottlenecks arising in bioinformatics.

3.2. Discrete statistics and probability

At a lower level, our work relies on a basic background on discrete statistics and probability. 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. Probabilistic models are also used to describe genome evolution. In this context, Bayesian models and their MCMC sampling allow to approximate probability distributions over parameters and to describe more biologically relevant models.

DOLPHIN Project-Team

3. Research Program

3.1. Hybrid multi-objective optimization methods

The success of metaheuristics is based on their ability to find efficient solutions in a reasonable time [89]. 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 [84], [88]. To be convinced one may refer to the list of references on Evolutionary Multi-objective Optimization maintained by Carlos A. Coello [4], which contains more 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'.

3.1.1. 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 [93]. 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.

---

[4]http://www.lania.mx/~ccoello/EMOO/EMOObib.html

- *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, 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.

3.1.2. *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 $A^{\star}$, 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 [86].

3.1.3. *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).

3.2. 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.

3.2.1. 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 back 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 [87], [92].

3.2.2. 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.

<p align="center" style="color:red">DREAMPAL Team</p>

3. Research Program

3.1. New Models for New Technologies

Over the past 25 years there have been several hardware-architecture generations dedicated to massively parallel computing. We have contributed to them in the past, and shall continue doing so in the Dreampal project. The three generations, chronologically ordered, are:

- Supercomputers from the 80s and 90s, based on massively parallel architectures that are more or less distributed (from the Cray T3D or Connection Machine CM2 to GRID 5000). Computer scientists have proposed methods and tools for mapping sequential algorithms to those parallel architectures in order to extract maximum power from them. We have contributed in this area in the past: http://www.lifl.fr/west/team.html.

- Parallelism pervades the chips! A new challenge appears: hardware/software co-design, in order to obtain performance gains by designing algorithms together with the parallel architectures of chips adapted to the algorithms. During the previous decade many studies, including ours in the Inria DaRT team, were dedicated to this type of co-design. DaRT has contributed to the development of the OMG MARTE standard (http://www.omgmarte.org) and to its implementation on several parallel platforms. Gaspard2, our implementation of this concept, was identified as one of the key software tools developed at Inria: http://www.inria.fr/en/centre/lille/research/platforms-and-flagship-software/flagship-software.

- The new challenge of the 2010s is, in our opinion, the integration of dynamic reconfiguration and massive parallelism. New circuits with high-density integration and supporting dynamic hardware reconfiguration have been proposed. In such architectures one can dynamically change the architecture while an algorithm is running on it. The Dynamic Partial Reconfiguration (DPR) feature offered by recent FPGA boards even allows, in theory, to generate optimized hardware at runtime, by adding, removing, and replacing components on a by-need basis. This integration of dynamic reconfiguration and massive parallelism induces a new degree of complexity, which we, as computer scientists, need to understand and deal with order to make possible the design of applications running on such architectures. This is the main challenge that we address in the Dreampal project. We note that we adress these problems as computer scientists; we do, however, collaborate with electronics specialists in order to benefit from their expertise in 3-D FPGAs.

Excerpt from the HiPEAC vision 2011/12

> *"The advent of 3D stacking enables higher levels of integration and reduced costs for off-chip communications. The overall complexity is managed due to the separation in different dies, independently designed."*

FPGAs (Field Programmable Gate Arrays) are configurable circuits that have emerged as a privileged target platform for intensive signal processing applications. FPGAs take advantage of the latest technological developments in circuits. For example, the Virtex7 from Xilinx offers a 28-nanometer integration, which is only one or two generations behind the latest general-purpose processors. 3D-Stacked Integrated Circuits (3D SICs) consist of two or more conventional 2D circuits stacked on the top of each other and built into the same IC. Recently, 3D SICs have been released by Xilinx for the Virtex 7 FPGA family. 3D integration will vastly increase the integration capabilities of FPGA circuits. The convergence of massive parallelism and dynamic reconfiguration in inevitable: we believe it is one of the main challenges in computing for the current decade.

By incorporating the configuration and/or data/program memory on the top of the FPGA fabric, with fast and numerous connections between memory and elementary logic blocks ($\sim$10000 connections between dies), it will be possible to obtain dynamically reconfigurable computing platforms with a very high reconfiguration rate. Such a rate was not possible before, due to the serial nature of the interface between the configuration memory and the FPGA fabric itself. The FPGA technology also enables massively parallel architectures due to the large number of programmable logic fabrics available on the chip. For instance, Xilinx demonstrated 3600 8-bit picoBlaze softcore processors running simultaneously on the Virtex-7 2000T FPGA. For specific applications, picoBlaze can be replaced by specialized hardware accelerators or other IPs (Intellectual Property) components. This opens the possibility of creating massively parallel IP-based machines.

3.2. Multi-softcore on 3D FPGA

From the 2010 Xilinx white paper on FPGAs:

> *"Unlike a processor, in which architecture of the ALU is fixed and designed in a general-purpose manner to execute various operations, the CLBs (configurable logic blocks) can be programmed with just the operations needed by the application... The FPGA architecture provides the flexibility to create a massive array of application-specific ALUs..The new solution enables high-bandwidth connectivity between multiple die by providing a much greater number of connections... enabling the integration of massive quantities of interconnect logic resources within a single package"*

Softcore processors are processors implemented using hardware synthesis. Proprietary solutions include PicoBlaze, MicroBlaze, Nios, and Nios II; open-source solutions include Leon, OpenRisk, and FC16. The choice is wide and many new solutions emerge, including multi-softcore implementations on FPGAs. An alternative to softcores are hardware accelerators on FPGAs, which are dedicated circuits that are an order of magnitude faster than softcores. Between these two approaches, there are other various approaches that connect IPs to softcores, in which, the processor's machine-code language is extended, and IP invocations become new instructions. We envisage a new class of softcores (we call them reflective softcores [1]), where almost everything is implemented in IPs; only the control flow is assigned to the softcore itself. The partial dynamic reconfiguration of next-generation FPGAs makes such dynamic IP management possible in practice. We believe that efficient reflective softcores on the new 3D-FPGAs should be as small as possible: low-performance generic hardware components (ALU, registers, memory, I/O...) should be replaced by dedicated high-performance IPs.

The reflective softcore HoMade that we have started developing in 2012 (http://www.lifl.fr/~dekeyser/Homade) follows these ideas. The current version is statically configurable; dynamically reconfigurable multi-processors are the next steps. Users have to add to it the functionality they need in their applications via IPs. We have also developped a library of IPs for the most common processor functions (ALU, registers, ...).

In the multi-reflective softcores that we shall develop, some softcores will be slaves and others will be masters. Massively parallel dynamically reconfigurable architectures of softcores can thus be envisaged. This requires, additionally, a parallel management of the partial dynamic reconfiguration system. This can be done, for example, on a given subset of softcores: a massively parallel reconfiguration will replace the current replication of a given IP with the replication of a new IP. Thanks to the new 3D-FPGAs this task can be performed efficiently and in parallel using the large number of 3D communication links (Through-Silicon-Vias). Our roadmap for HoMade is to evolve towards this multi-reflective softcore model.

3.3. When Hardware Meets Software

HIPEAC vision 2011/12: *"The number of cores and instruction set extensions increases with every new generation, requiring changes in the software to effectively exploit the new features."*

---

[1] Hereafter, by reflective system, we mean a system that is able to modify its own structure and behaviour while it is running. A reflective softcore thus dynamically adds, removes, and replaces IPs in the application running on it, and is able to dynamically modify its own program memory, thereby dynamically altering the program it is executing.

When the new massively parallel dynamically reconfigurable architectures become reality users will need languages for programming software applications on them. The languages will be themselves dynamic and parallel, in order to reflect and to fully exploit the dynamicity and parallelism of the architectures. Thus, developers will be able to invoke reconfiguration and call parallel instructions in their programs. This expressiveness comes with a cost, however, because new classes of bugs can be induced by the interaction between dynamic reconfiguration and parallelism; for example, deadlocks due to waiting for output from an IP that does not exist any more due to a reconfiguration. The detection and elimination of such bugs before deployment is paramount for cost-effectiveness and safety reasons.

Thus, we shall build an environment for developing software on parallel, dynamically reconfigurable architectures that will include languages and adequate formal analyses and verification tools for them, in addition to more traditional tools (emulators, compilers, etc). To this end we shall be using formal-semantics frameworks associated with easy-to-use formal verification tools in order to formally define our languages of interest and allow users to formally verify their programs. The K semantic framework (http://k-framework.org), developed jointly by Univs. Urbana Champaign, USA, and Iasi, Romania) is one such framework, which is mature enough (it has allowed defining a formal semantics of the largest subset of the C language to date, as well as many other languages from essentially all programming paradigms) and is familiar to us from previous work. In K, one can rapidly prototype a language definition and try several versions of the syntax and semantics of instructions. This is important in our project, where the proposed programming languages (in particular, the HoMade assembly language) will go through several versions before being stabilized. Moreover, once a language is defined in K one gets an interpreter of the language and one gains access to formal verification tools for free. We are also developing new analysis verification tools for K (in collaboration with the K team), which will be adapted and used in the Dreampal project.

We shall focus on embedded systems performing intensive computations, in particular smart-camera systems and set-up-boxes. Some of the targeted classes of applications are safety-critical, and formal verification is essential for them. For the other ones, formal verification provides an added value in terms of quality of service.

<p style="text-align:center; color:red">FUN Project-Team</p>

3. Research Program

3.1. Introduction

The research area of FUN research group is represented in Figure 1 . FUN research group will address every item of Figure 1 starting from the highest level of the figure, *i.e.* in area of homogeneous FUNs to the lowest one. Going down brings more applications and more issues to solve. Results achieved in the upper levels can be re-used in the lower ones. Current networks encountered nowadays are the ones at the higher level, without any interaction between them. In addition, solutions provided for such networks are rarely directly applicable in realistic networks because of the impact of the wireless medium.

FUN research group intends to fill the scientific gap and extend research performed in the area of wireless sensor and actor networks and RFID systems in two directions that are complementary and should be performed in parallel:

- **From theory to experimentation and reciprocally** On one hand, FUN research group intends to investigate new self-organization techniques for these future networks that take into account realistic parameters, emphasizing experimentation and considering mobility.

- **Towards heterogeneous FUNs** On the other hand, FUN research group intends to investigate techniques to allow heterogeneous FUNs to work together in a transparent way for the user. Indeed, new applications integrating several of these components are very much in demand (i.e. smart building) and thus these different technologies need to cooperate.

3.2. From theory to experimentation and reciprocally

Nowadays, even if some powerful and efficient propositions arise in the literature for each of these networks, very few are validated by experimentations. And even when this is the case, no lesson is learnt from it to improve the algorithms. FUN research group needs to study the limits of current assumptions in realistic and mobile environments.

Solutions provided by the FUN research group will mainly be algorithmic. These solutions will first be studied theoretically, principally by using stochastic geometry (like in [57]) or self-stabilization [59] tools in order to derive algorithm behavior in ideal environment. Theory is not an end in itself but only a tool to help in the characterization of the solution in the ideal world. For instance, stochastic geometry will allow quantifying changes in neighborhood or number of hops in a routing path. Self-stabilization will allow measuring stabilization times.

Those same solutions will then be confronted to realistic environments and their 'real' behavior will be analyzed and compared to the expected ones. Comparing theory, simulation and experimentation will allow will allow the influence of a realistic environment be better measured. From this and from the analysis of the information really available for nodes, FUN research group will investigate some means either to counterbalance these effects or to take advantage of them. New solutions provided by the FUN research group will take into consideration the vagaries of a realistic wireless environment and the node mobility. New protocols will take as inputs environmental data (as signal strength or node velocity/position, etc) and node characteristics (the node may have the ability to move in a controlled way) when available. FUN research group will thus adopt a **cross-layered** approach between hardware, physical environment, application requirements, self-organizing and routing techniques. For instance, FUN research group will study how the controlled node mobility can be exploited to enhance the network performance at lowest cost.
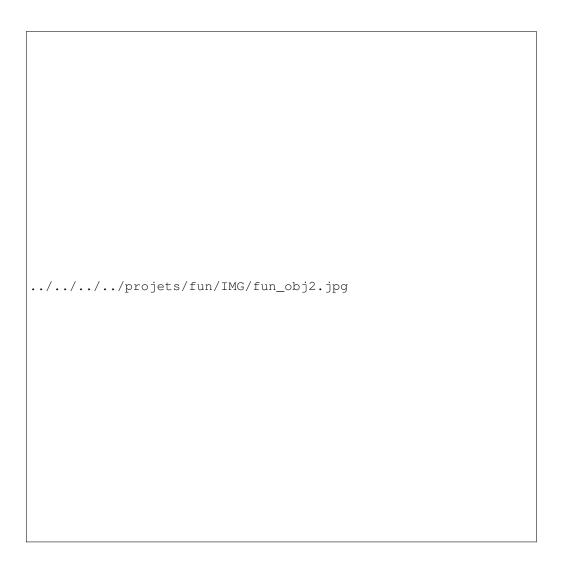
../../../../projets/fun/IMG/fun_obj2.jpg

Figure 1. Panorama of FUN.

Solutions will follow the building process presented by Figure 2 . Propositions will be analyzed not only theoretically and by simulation but also by experimentation to observe the impact of the realistic medium on the behavior of the algorithms. These observations should lead to the derivation of cross-layered models. Experimentation feedbacks will be re-injected in solution design in order to propose algorithms that best fit the environment, and so on till getting satisfactory behavior in both small and large scale environments. All this should be done in such a way that the resulting propositions fit the hardware characteristics (low memory, CPU and energy capacity) and easy to deploy to allow their use by non experts. Since solutions should take into account application requirements as well as hardware characteristics and environment, solutions should be generic enough and then able to self-configure to adapt their environment settings.

In order to achieve this experimental environments, the FUN research group will maintain its strong activity on platform deployment such as SensLAB [63], FIT and Aspire [53]. Next steps will be to experiment not only on testbeds but also on real use cases. These latter will be given through different collaborations.

../../../../projets/fun/IMG/methodology.png

Figure 2. Methodology applied in the FUN research group.

FUN research group will investigate self-organizing techniques for FUNs by providing cross-layered solutions that integrate in their design the adaptability to the realistic environment features. Every solution will be validated with regards to specific application requirements and in realistic environments.

Facing the medium instability. The behavior of wireless propagation is very depending of the surrounding environment (in-door vs outdoor, night vs day, etc) and is very instable. Many experiments in different environment settings should be conducted. Experiment platforms such as SensLAB, FIT, our wifiBot as robots and actuators and our RFID devices will be used offering ways to experiment easily and quickly in different environments but might not be sufficient to experiment every environment.

Adaptability and flexibility. Since from one application to another one, requirements and environments are different, solutions provided by FUN research group should be **generic** enough and **self-adapt** to their environment. Algorithm design and validation should also take into account the targeted applications brought for instance by our industrial partners like Etineo. All solution designs should keep in mind the devices constrained capacities. Solutions should consume low resources in terms of memory, processor and energy to provide better performances and scale. All should be self-adaptive.

FUN research group will try to take advantage of some observed features that could first be seen as drawbacks. For instance, the broadcast nature of wireless networks is first an inconvenient since the use of a link between two nodes inhibits every other communication in the same transmission area. But algorithms should exploit that feature to derive new behaviors and a node blocked by another transmission should overhear it to get more information and maybe to limit the overall information to store in the network or overhead communication.

3.3. Towards unified heterogeneous FUNs

The second main direction to be followed by the FUN research group is to merge networks from the upper layer in Fig. 1 into networks from the lowest level. Indeed, nowadays, these networks are still considered as separated issues. But considering mixed networks bring new opportunities. Indeed, robots can deploy, replace, compensate sensor nodes. They also can collect periodically their data, which avoids some long and multi-hop communications between sensor nodes and thus preserving their resources. Robots can also perform many additional tasks to enhance network performance like positioning themselves on strategic points to ensure area coverage or reduce routing path lengths. Similarly, coupling sensors and RFID tags also bring new opportunities that are more and more in-demand from the industrial side. Indeed, an RFID reader may be a sensor in a wireless sensor network and data hold by RFID tags and collected by readers might need to be reported to a sink. This will allow new applications and possibilities such as the localization of a tagged object in an environment be covered by sensors.

When at last all components are gathered, this leads us to a new era in which every object is autonomous. Let's consider for instance a smart home equipped with sensors and RFID reader. An event triggered by a sensor (*i.e.* an increase of the temperature) or a RFID reader (*i.e.* detection of a tag hold by a person) will trigger actions from actuators (*i.e.* lowering of stores, door opening). Possibilities are huge. But with all these new opportunities come new technological issues with other constraints. Every entity is considered as an object possibly mobile which should be dynamically identified and controlled. To support this dynamics, protocols should be localized and distributed. Model derived from experiment observations should be unified to fit all these classes of devices.

FUN research group will investigate new protocols and communication paradigms that allow the technologies to be transparently merged. Objects and events might interconnect while respecting on-going standards and building an autonomic and smart network while being compliant with hardware resources and environment.

Technologies such as wireless sensors, wireless robots/actuators and RFID tags/ readers, although presenting many common points are still part of different disciplines that have evolved in parallel ways. Every branch is at different maturity levels and has developed its own standards. Nevertheless, making all these devices part of a single unified network leverages technological issues (partly addressed in the former objective) but also regarding to on-going standards and data formatting. FUN research group will have to study current standards of every area in order to propose compliant solutions. Such works have been initiated in the POPS research group in the framework of the FP7 ASPIRE project. Members of FUN research group intend to continue and enlarge these works.

Todays' EPCGlobal compliant RFID readers must comply to some rules and be configurable through an ALE (Application Level Event) [51]. While a fixed and connected RFID reader is easily configurable, configuring remotely a mobile RFID reader might be very difficult since it implies to first locate it and then send configuration data through a wireless dynamic network. FUN research group will investigate some tools that make the configuration easy and transparent for the user. This remote configuration of mobile readers through the network should consider application requirements and network and reader characteristics to choose the best trade-off relative to the software part embedded in the reader. The biggest part embedded, the lowest bandwidth overhead (data can be filtered and aggregated in the reader) and the greater mobility (readers are still fully operational even when disconnected) but the more difficult to set up and the more powerful readers. All these aspects will be studied within the FUN research group.

LINKS Team (section vide)

MAGNET Team

3. Research Program

3.1. Introduction

The main objective of MAGNET is to develop original machine learning methods for networked data. We consider information networks in which the data are vectorial data and texts. We model such information networks as (multiple) (hyper)graphs wherein nodes correspond to entities (documents, spans of text, users, ...) and edges correspond to relations between entities (similarity, answer, co-authoring, friendship, ...). Our main research goal is to propose new learning algorithms to build applications like browsing, monitoring and recommender systems, and more broadly information extraction in information networks. Hence, we will investigate new learning algorithms for node clustering and node classification, link classification and link prediction. Also, we will search for the best hidden graph structure to be generated for solving a given learning task. We will base our research on generative models for graphs, on machine learning for graphs and on machine learning for texts. The challenges are the dimensionality of the input space, possibly the dimensionality of the output space, the high level of dependencies between the data, the inherent ambiguity of textual data and the limited amount of human labeling. An additional challenge will be to design scalable methods for large information networks. Hence, we will explore how sampling and randomization can be used in new machine learning algorithms. Also, active machine learning algorithms for graphs will be investigated.

On the first hand we want to design machine learning algorithms on graphs to solve problems in networks of texts and documents in natural language. The main originality of this research is to consider and take advantage of the setting of networked data exploiting the relationships between different data entities and, overall, the graph topology. On the second hand, in a concomitant way, we want to develop prediction models for graph-like data. This includes prediction, ranking and classification of links and nodes in an on-line or batch setting. The two objectives are intertwined, enrich each other and raise important scientific questions we want to focus on. Our research proposal is organized according to the following questions:

1. How to go beyond vectorial classification models in natural language oriented tasks?
2. How to adaptively build graphs with respect to the given tasks? How to create network from observations of information diffusion processes?
3. How to design methods able to achieve very good predictive accuracy without giving up on scalability?
4. How to go beyond strict node homophilic/similarity assumptions in graph-based learning methods?

3.2. Beyond vectorial models for NLP

One of our overall research objectives is to derive graph-based machine learning algorithms for natural language and text information extraction tasks. This section discusses the motivations behind the use of graph-based ML approaches for these tasks, the main challenges associated with it, as well as some concrete projects. Some of the challenges go beyond NLP problems and will be further developed in the next sections. An interesting aspect of the project is that we anticipate some important cross-fertilizations between NLP and ML graph-based techniques, with NLP not only benefiting from but also pushing ML graph-based approaches into new directions.

Motivations for resorting to graph-based algorithms for texts are at least threefold. First, online texts are organized in networks. With the advent of the web, and the development of forums, blogs, and micro-blogging, and other forms of social media, text productions have become strongly connected. Thus, documents on the web are linked through hyperlinks, forum posts and emails are organized in threads, tweets can be retweeted, etc. Additional connections can be made through users connections (co-authorship, friendship, follower, etc.). Interestingly, NLP research has been rather slow in coming to terms with this situation, and most work still focus on document-based or sentence-based predictions (wherein inter-document or inter-sentence structure is not exploited). Furthermore, several multi-document tasks exist in NLP (such as multi-document summarization and cross-document coreference resolution), but most existing work typically ignore document boundaries and simply apply a document-based approach, therefore failing to take advantage of the multi-document dimension [28], [30].

A second motivation comes from the fact that most (if not all) NLP problems can be naturally conceived as graph problems. Thus, NL tasks often involve discovering a relational structure over a set of text spans (words, phrases, clauses, sentences, etc.). Furthermore, the *input* of numerous NLP tasks is also a graph; indeed, most end-to-end NLP systems are conceived as pipelines wherein the output of one processor is in the input of the next. For instance, several tasks take POS tagged sequences or dependency trees as input. But this structured input is often converted to a vectorial form, which inevitably involves a loss of information.

Finally, graph-based representations and learning methods in principle appear to address some core problems faced by NLP, such as the fact that textual data are typically not independent and identically distributed, they often live on a manifold, they involve very high dimensionality, and their annotations is costly and scarce. As such, graph-based methods represent an interesting alternative, or at least complement, to structured prediction methods (such as CRFs or structured SVMs) commonly used within NLP. While structured output approaches are able to model local dependencies (e.g., between neighboring words or sentences), they cannot efficiently capture long distance dependencies, like forcing a particular n-gram to receive the same labeling in different sentences or documents for instance. On the other hand, graph-based models provide a natural way to capture global properties of the data through the exploitation of walks and neighborhood in graphs. Graph-based methods, like label propagation, have also been shown to be very effective in semi-supervised settings, and have already given some positive results on a few NLP tasks [10], [32].

Given the above motivations, our first line of research will be to investigate how one can leverage an underlying network structure (e.g., hyperlinks, user links) between documents, or text spans in general, to enhance prediction performances for several NL tasks. We think that a "network effect", similar to the one that took place in Information Retrieval (with the Page Rank algorithm), could also positively impact NLP research. A few recent papers have already opened the way, for instance in attempting to exploit Twitter follower graph to improve sentiment classification [31].

Part of the challenge in this work will be to investigate how adequately and efficiently one can model these problems as instances of more general graph-based problems, such as node clustering/classification or link prediction discussed in the next sections. In a few cases, like text classification or sentiment analysis, graph modeling appears to be straightforward: nodes correspond to texts (and potentially users), and edges are given by relationships like hyperlinks, co-authorship, frienship, or thread membership. Unfortunately, modeling NL problems as networks is not always that obvious. From the one hand, the right level of representation will probably vary depending on the task at hand: the nodes will be sentences, phrases, words, etc. From the other hand, the underlying graph will typically not be given a priori, which in turn raises the question of how we construct it. Of course, there are various well-known ways to obtain similarity measures between text contents (and its associated vectorial data), and graphs can be easily constructed from those combined with some sparsification method. But we would like our similarity to be tailored to the task objective. An additional problem with many NLP problems is that features typically live in different types of spaces (e.g., binary, discrete, continuous). A preliminary discussion of the issue of optimal graph construction for semi-supervised learning in NLP is given in [10], [26]. We identify the issue of adaptative graph construction as an important scientific challenge for machine learning on graphs in general, and we will discuss it further in Section 3.3 .

As noted above, many NLP tasks have been recast as structure prediction problems, allowing to capture (some of the) output dependencies. Structure prediction can be viewed as (set of) link prediction with global loss or dependencies, which means that graph-based learning methods can handle (at least, approximately) output prediction dependencies, and they can in principle capture additional more global dependencies given the right graph structure. How to best combine structured output and graph-based ML approaches is another challenge that we intend to address. We will initially investigate this question within a semi-supervised context, concentrating on graph based regularization and graph propagation methods. Within such approaches, labels are typically binary or they correspond to small finite set. Our objective is to explore how one propagates an exponential number of *structured labels* (like a sequence of tags or a dependency tree) through graphs. Recent attempts at blending structured output models with graph-based models are investigated in [32], [18]. Another related question that we will address in this context is how does one learn with *partial labels* (like partially specificied tag sequence or tree) and use the graph structure to complete the output structure. This last question is very relevant to NL problems where human annotations are costly; being able to learn from partial annotations could therefore allow for more targeted annotations and in turn reduced costs [19].

The NL tasks we will mostly focus on are coreference resolution and entity linking, temporal structure prediction, and discourse parsing. These tasks will be envisonned in both document and cross-document settings, although we expect to exploit inter-document links either way. Choices for these particular tasks is guided by the fact that are still open problems for the NLP community, they potentially have a high impact for industrial applications (like information retrieval, question answering, etc.), and we already have some expertise on these tasks in the team. As a midterm goal, we also plan to work on tasks more directly relating to micro-blogging, such sentiment analysis and the automatic thread structuring of technical forums; the latter task is in fact an instance of rhetorical structure prediction [34].

We have already initiated some work on the coreference resolution problem in the context of ML graph-based approaches. We cast this problem as a spectral clustering problem. Given than features can be numerical or nominal, the definition of a good similarity measure between entities is not straightforward. As a first solution, we consider only numerical attributes to build a k-nn graph of mentions so that graph clustering methods can be applied. Nominal attributes and relations are introduced by means of soft constraints on this clustering. Constraints can have various forms and have the ability of going beyond homophily assumptions, taking into account for instance dissimilarity relationships. From this setting we derive new graph-based learning methods. We propose to study the modification of graph clustering and spectral embeddings to satisfy certain constraints induced by several types of supervision: (i) nodes belong to the same group or to different groups, and (ii) some groups are fully known while others have to be discovered. This semi-supervised graph clustering problem is studied in a batch and transductive setting. But interesting extensions can be investigated in an online and active setting.

3.3. Adaptive Graph Construction

In most applications, edge weights are computed through a complex data-modeling process and convey crucially important information for classifying nodes, which makes it possible to infer information related to each data sample even exploiting the graph topology solely. In fact, a widespread approach to the solution of several classification problems is representing the data through an undirected weighted graph in which edge weights quantify the similarity between data points. This technique for coding input data has been applied to several domains, including classification of genomic data ([29]), face recognition ([17]), and text categorization ([22]).

In some cases, the full adjacency matrix is generated by employing suitable similarity functions chosen through a deep understanding of the problem structure. For example TF-IDF representation of documents, the affinity between pairs of samples is often estimated through the cosine measure or the χ^2 distance. After the generation of the full adjacency matrix, the second phase for obtaining the final graph consists in an edge sparsification/reweighting operation. Some of the edges of the clique obtained in the first step are pruned and the remaining ones can be reweighted to meet the specific requirements of the given classification problem. Constructing a graph with these methods obviously entails various kinds of loss of information. However, in

problems like node classification, the use of graphs generated from several datasets can lead to an improvement in accuracy performance ([35], [11], [12]). Hence, the transformation of a dataset into a graph may, at least in some cases, partially remove various kinds of irregularities present in the original datasets, while keeping some of the most useful information for classifying the data samples. Moreover, it is often possible to accomplish classification tasks on the obtained graph using a running time remarkably lower than is needed by algorithms exploiting the initial datasets, and a suitable sparse graph representation can be seen as a compressed version of the original data. This holds even when input data are provided in a online/stream fashion, so that the resulting graph evolves over time.

In this project we will address the problem of adaptive graph construction towards several directions. One is the question of choosing the best similarity measure given the objective learning task. This question is related to the question of similarity learning ([13]) which has not been considered in the context of graph based learning. In the context of structured prediction, we will develop approaches where output structures are organized in graphs whose similarity is given by top-k outcomes of greedy algorithms.

A different way we envision adaptative graph construction is in the context of semi-supervised learning. Partial supervision can take various forms and an interesting and original setting is governed by two currently studied applications: detection of brain anomaly from connectome data and polls recommendation in marketing. Indeed, for these two applications, a partial knowledge of the information diffusion process can be observed while the network is unknown or only partially known. An objective is to construct (or complete) the network structure from some local diffusion information. The problem can be formalized as a graph construction problem from partially observed diffusion processes. It has been studied very recently in [24]. In our case, the originality comes either from the existence of different sources of observations or from the large impact of node contents in the network.

We will study how to combine graphs defined by networked data and graphs built from flat data to solve a given task. This is of major importance for information networks because, as said above, we will have to deal with multiple relations between entities (texts, spans of texts, ...) and also use textual data and vectorial data. We have started to work on combining graphs in a semi supervised setting for node classification problems along the PhD thesis of T. Ricatte. Future work include combination geared by semi-supervision on link prediction tasks. This can be studied in an active learning setting. But one important issue is to design scalable approaches, thus to exploit locality given by the network. Doing this we address another objective to build non uniformly parameterized combinations.

3.4. Prediction on Graphs and Scalability

As stated in the previous sections, graphs as complex objects provides a rich representation of data. Often enough the data is only partially available and the graph representation is very helpful in predicting the unobserved elements. We are interested in problems where the complete structure of the graph needs to be recover and only a fraction of the links is observed. The link prediction problem falls into this category. We are also interested in the recommendation and link classification problems which can be seen as graphs where the structure is complete but some labels on the links (weights or signs) are missing. Finally we are also interested in labelling the nodes of the graph, with class or cluster memberships or with a real value, provided that we have (some information about) the labels for some of the nodes.

The semi-supervised framework will be also considered. A midterm research plan is to study how graph-based regularization models help for structured prediction problems. This question will be studied in the context of NLP tasks, as noted in Section 3.2 , but we also plan to develop original machine learning algorithms that have a more general applicability. Inputs are networks whose nodes (texts) have to be labeled by structures. We assume that structures lie in some manifold and we want to study how labels can propagate in the network. One approach is to find smooth labeling function corresponding to an harmonic function on both manifolds in input and output. We also plan to extend our results on spectral clustering with must-link and cannot-link constraints in two directions. We have proposed a batch method with an optimization problem based on an adaptive spectral embedding with respects to constraints. We want to extend this approach to an on-line and

active setting where a flow of graphs (each one is a document) is given as input. In the case of large graphs, we also consider the case where partial supervision consists in the knowledge of few clusters.

Scalability is one of the main issue in the design of new prediction algorithms working on networked data. It has gained more and more importance in recent years, because of the growing size of the most popular networked data that are now used by millions of people. In such contexts, learning algorithms whose computation time scales quadratically, or slower, in the number of considered data objects (usually nodes or vertices, depending on the given task) should be considered impractical.

These observations lead to the idea of using graph sparsification techniques in order to work on a part of the original network for getting results that can be easily extended and used for the whole original input. A sparsified version of the original graph can often be seen as a subset of the initial input, i.e. a suitably selected input subgraph which forms the training set (or, more in general, it is included in the training set). This holds even for the active setting.

A simple example could be to find a spanning tree of the input graph, possibly using randomization techniques, with properties such that we are allowed to obtain interesting results for the initial graph dataset. We have started to explore this research direction for instance in [33]. This approach leaves us with the problem of choosing a good spanning tree, taking into account that the setting could be adversarial (e.g, in the online case the presentation and the assignment of the labels are both arbitrary). A suitable use of the randomization power becomes therefore remarkably significant. Moreover, it is interesting to observe that running a prediction algorithm on a sparsified version of the input dataset allows the parallelization of prediction tasks. In fact, given a prediction task for a networked dataset, in a preliminary phase one could run a randomized graph sparsification method in parallel on different machines. For example, in the case of the spanning tree use, one could then draw several spanning trees at the same time, each on a different computer. This way it is possible to simultaneously run different prediction experiments on the same task and aggregating the obtained results at the end, with several methods (e.g. simply by majority vote) in order to increase the robustness and accuracy predictions.

At the level of the mathematical foundations, the key issue to be addressed in the study of (large-scale) random networks also concerns the segmentation of network data into sets of independent and identically distributed observations. If we identify the data sample with the whole network, as it has been done in previous approaches [23], we typically end up with a set of observations (such as nodes or edges) which are highly interdependent and hence overly violate the classic i.i.d. assumption. In this case, the data scale can be so large and the range of correlations can be so wide, that the cost of taking into account the whole data and their dependencies is typically prohibitive. On the contrary, if we focus instead on a set of subgraphs independently drawn from a (virtually infinite) target network, we come up with a set of independent and identically distributed observations—namely the subgraphs themselves, where subgraph sampling is the underlying ergodic process [14]. Such an approach is one principled direction for giving novel statistical foundations to random network modeling. At the same time, because one shifts the focus from the whole network to a set of subgraphs, complexity issues can be restricted to the number of subgraphs and their size. The latter quantities can be controlled much more easily than the overall network size and dependence relationships, thus allowing to tackle scalability challenges through a radically redesigned approach.

We intend to develop new learning models for link prediction problems. We have already proposed a conditional model in [21] with statistics based on Fiedler values computed on small subgraphs. We will investigate the use of such a conditional model for link prediction. We will also extend the conditional probabilistic models to the case of graphs with textual and vectorial data by defining joint conditional models. Indeed, an important challenge for information networks is to introduce node contents in link ranking and link prediction methods that usually rely solely on the graph structure. A first step in this direction was already proposed in [20] where we learn a mapping of node content to a new representation constrained by the existing link structure and applied it for link recommendation. This approach opens a different view on recommendation by means of link ranking problems for which we think that non parametric approaches should be fruitful.

Regarding link classification problems, we plan to devise a whole family of active learning strategies, which could be based on spanning trees or sparse input subgraphs, that exploit randomization and the structure of the graph in order to offset the adversarial label assignment. We expect these active strategies to exhibit good accuracies with a remarkably small number of queried edges, where passive learning methods typically break down. The theoretical findings can be supported by experiments run on both synthetic and real-world (Slashdot, Epinions, Wikipedia, and others) datasets.

We are interested in studying generative models for graph labeling, exploiting the results obtained in p-stochastic model for link classification (investigated in [16]) and statistical model for node label assignment which can be related to tree-structured Markov random fields [25].

In developing our algorithms, we focus on providing theoretical guarantees on prediction accuracy and, at the same time, on computational efficiency. The development of methods that simultaneously guarantee optimal accuracy and computational efficiency is a very challenging goal. In fact, the accuracy of most methods in the literature is not rigorously analyzed from a theoretical point of view. Likewise, tight time and space complexity bounds are not generally provided. This contrasts with the need to manage extremely large relational datasets like, e.g., snapshots of the World Wide Web.

3.5. Beyond Homophilic Relationships

In many cases, the algorithms devised for solving node classification problems are driven by the following assumption: linked entities tend to be assigned to the same class. This assumption, in the context of social networks, is known as homophily ([15], [27]) and involves ties of every type, including friendship, work, marriage, age, gender, and so on. In social networks, homophily naturally implies that a set of individuals can be parted into subpopulations that are more cohesive. In fact, the presence of homogeneous groups sharing interests is one of the most significant reasons for affinity among interconnected individuals, which suggests that, in spite of its simplicity, this principle turns out to be very powerful for node classification problems in general networks.

Recently, however, researchers have started to consider networked data where connections may also carry a negative meaning. For instance, disapproval or distrust in social networks, negative endorsements on the Web. Concrete examples are provided by certain types of online social networks. Users of Slashdot can tag other users as friends or foes. Similarly, users of Epinions can give positive or negative ratings not only to products but also to other users. Even in the social network of Wikipedia administrators, votes cast by an admin in favor or against the promotion of another admin can be viewed as positive or negative links. More examples of signed links are found in other domains, such as the excitatory or inhibitory interactions between genes or gene products in biological networks.

Although the introduction of signs on graph edges appears like a small change from standard weighted graphs, the resulting mathematical object, called signed graph, has an unexpectedly rich additional complexity. For example, the spectral properties of signed graphs, which essentially all sophisticated node classification algorithms rely on, are different and less known than those of their unsigned counterparts. Signed graphs naturally lead to a specific inference problem that we have discussed in previous sections: link classification. This is the problem of predicting the sign of links in a given graph. In online social networks, this may be viewed as a form of sentiment analysis, since we would like to semantically categorize the relationship between individuals.

Another way to go beyond homophily between entities will be studied using our recent model of hypergraphs with bipartite hyperedges [8]. A bipartite hyperedge connects two ends which are disjoint subsets of nodes. Bipartite hyperedges is a way to relate two collections of (possibly heterogeneous) entities represented by nodes. In the NLP setting, while hyperedges can be used to model bags of words, bipartite hyperedges are associated with relationships between bags of words. But each end of bipartite hyperedges is also a way to represent complex entities, gathering several attribute values (nodes) into hyperedges viewed as records. Our hypergraph notion naturally extends directed and undirected weighted graph. We have defined a spectral theory for this new class of hypergraphs and opened a way to smooth labeling on sets of nodes. The weighting

scheme permits to weight the participation of each node to the relationship modeled by bipartite hyperedges accordingly to an equilibrium condition. This is exactly that equilibrium condition that provides a competition between nodes in hyperedges and allows interesting modeling properties that go beyond homophily and similarity over nodes. (Theoretical analysis of our hypergraphs exhibits tight relationships with signed graphs). Following this competition idea, bipartite hyperedges are like matches between two teams and examples of applications are team creation. The basic tasks in which we are interested in are hyperedge classification, hyperedge prediction, node weight prediction. Finally, hypergraphs also represent a way to summarize or compress large graphs in which there exists highly connected couples of (large) subsets of nodes.

To conclude, we plan to go beyond the homophilic bias from the algorithmic as well as from the modeling point of view. We will consider new kind of modeling and learning biases provided by graphs with negative weights (signed graphs) and hypergraphs. We will study their spectral properties, smoothness measures of (node or edge) labeling. Sampling and walking also need to be reconsidered. From the machine learning perspective, we will study edge and node labeling in batch and online settings. In connection with our main targeted applications, we will mainly consider unsupervised and semi-supervised situations. We think that allowing negative weights and advanced relationships on nodes will also lead to space efficient representations of graphs.

MINT Project-Team

3. Research Program

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 [32]. 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 [37]. Rapid and iterative prototyping techniques allow to decide between alternative solutions [33]. Controlled studies based on experimental or quasi-experimental designs can then be used to evaluate the chosen solutions [39]. 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 [36]. 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 [40], Dynamic Time Warping [42] 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 [38]; 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 [41].
- *mapping to application*, that studies how to link gesture inputs to application. This ranges from transfer function that is classically involved in pointing tasks [34], 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.

<h2 style="text-align:center; color:red">MODAL Project-Team</h2>

3. Research Program

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) spaces, 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, a 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.

<p style="text-align:center;color:red;font-weight:bold">NON-A Project-Team</p>

3. Research Program

3.1. General annihilators

Estimation is quite easy in the absence of perturbations. It becomes challenging in more realistic situations, faced to measurement noises or other unknown inputs. In our works, as well as in the founding text of *Non-A*, we have shown how our estimation techniques can successfully get rid of perturbations of the so-called *structured* type, which means the ones that can be annihilated by some linear differential operator (called the annihilator). *ALIEN* already defined such operators by integral operators, but using more general convolution operators is an alternative to be analyzed, as well as defining the "best way to kill" perturbations. Open questions are:

OQ1) Does a normal form exist for such annihilators?

OQ2) Or, at least, does there exist an adequate basis representation of the annihilator in some adequate algebra?

OQ3) And lastly, can the annihilator parameters be derived from efficient tuning rules?

The two first questions will directly impact Indicators 1 (time) and 2 (complexity), whereas the last one will impact indicator 3 (robustness).

3.2. Numerical differentiation

Estimating the derivative of a (noisy) signal with a sufficient accuracy can be seen as a key problem in domains of control and diagnosis, as well as signal and image processing. At the present stage of our research, the estimation of the n-th order time derivatives of noisy signals (including noise filtering for $n = 0$) appears as a common area for the whole project, either as a research field, or as a tool that is used both for model-based and model-free techniques. *One of the open questions is about the robustness issues (Indicator 3) with respect to the annihilator, the parameters and the numerical implementation choices.*

Two classes of techniques are considered here (**Model-based** and **Model-free**), both of them aiming at non-asymptotic estimation.

In what we call *model-based techniques*, the derivative estimation is regarded as an observation problem, which means the software-based reconstruction of unmeasured variables and, more generally, a left inversion problem [2]. This involves linear/homogeneous/nonlinear state models, including ordinary equations, systems with delays, hybrid systems with impulses or switches[3], which still has to be exploited in the finite-time and fixed-time context. Power electronics is already one of the possible applications.

Model-free techniques concern the works initiated by *ALIEN*, which rely on the only information contained in the output signal and its derivatives. The corresponding algorithms rely on our algebraic annihilation viewpoint. *One open question is: How to provide an objective comparison analysis between Model-based and Model-free estimation techniques? For this, we will only concentrate on Non-Asymptotic ones. This comparison will have to be based on the three Indicators 1 (time), 2 (complexity) and 3 (robustness).*

---

[2] Left invertibility deals with the question of recovering the full state of a system ("observation") together with some of its inputs ("unknown input observers"), and also refers to algebraic structural conditions.

[3] Note that hybrid dynamical systems (HDS) constitute an important field of investigation since, in this case, the discrete state can be considered as an unknown input.

3.3. Model-free control

Industry is keen on simple and powerful controllers: the tuning simplicity of the classical PID controller explains its omnipresence in industrial control systems, although its performances drop when working conditions change. The last challenge we consider is to define control techniques which, instead of using sophisticated models (the development of which may be expensive), use the information contained in the output signal and its estimated derivatives, which can be regarded as "signal-based" controllers. *Such design should take into account the Indicators 1 (time), 2 (complexity) and 3 (robustness).*

3.4. Applications

Keeping in mind that we will remain focused at developing and applying fundamental methods for non-asymptotic estimation, we intend to deal with 4 main domains of application (see the lower part of Figure 1). The Lille context offers interesting opportunities in WSAN (wireless sensor and actuator networks and, more particularly, networked robots) at Inria, as well as nano/macro machining at ENSAM. A power electronics platform will be developed in ENSEA Cergy. Last, in contact with companies, several grants, patents and collaborations are expected from the applications of $i-$PID. Each of these four application domains was presented in the *Non-A* proposal:

- Networked robots, WSAN [Lille]
- Nano/macro machining [Lille]
- Multicell chopper [Lille and Cergy]
- *i*-PID for industry

In the present period, we choose to give a particular focus to the first item (Networked robots), which already received some development. It can be considered as the objective 4.

These applications are described with more details below.

<p align="center" style="color:red">**RMOD Project-Team**</p>

3. Research Program

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 programming 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 be selected for a given remodularization?

To help us answer these questions, we work on enriching Moose, our reengineering environment, with a new set of analyses [55], [54]. We decompose our approach in three main and potentially overlapping steps:

1. Tools for understanding applications,
2. Remodularization analyses,
3. Software Quality.

3.1.1. Tools for understanding applications

Context and Problems. We are studying the problems raised by the understanding of applications at a larger level of granularity such as packages or modules. We want to develop a set of conceptual tools to support this understanding.

Some approaches based on Formal Concept Analysis (FCA) [83] show that such an analysis can be used to identify modules. However the presented examples are too small and not representative of real code.

Research Agenda.

FCA provides an important approach in software reengineering for software understanding, design anomalies detection and correction, but it suffers from two problems: (i) it produces lattices that must be interpreted by the user according to his/her understanding of the technique and different elements of the graph; and, (ii) the lattice can rapidly become so big that one is overwhelmed by the mass of information and possibilities [42]. We look for solutions to help people putting FCA to real use.

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 [71]. Until now, few works have attempted to identify layers in practice: Mudpie [85] is a first cut at identifying cycles between packages as well as package groups potentially representing layers. DSM (dependency structure matrix) [84], [79] 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 [67]. 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 [86], [58].

As we are designing and evaluating algorithms and analyses to remodularize applications, we also need a way to understand and assess the results we are obtaining.

Research Agenda. We work on the following items:

Layer identification.   We 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.

Cohesion Metric Assessment.   We are building a validation framework for cohesion/coupling metrics to determine whether they actually measure what they promise to. We are also compiling a number of traditional metrics for cohesion and coupling quality metrics to evaluate their relevance in a software quality setting.

3.1.3. Software Quality

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 Grail in the realm of software engineering. The question is still relevant and important. We work on the two following items:

Quality models. We studied existing quality models and the different options to combine indicators — often, software quality models happily combine metrics, but at the price of losing the explicit relationships between the indicator contributions. There is a need to combine the results of one metric over all the software components of a system, and there is also the need to combine different metric results for any software component. Different combination methods are possible that can give very different results. It is therefore important to understand the characteristics of each method.

Bug prevention.   Another aspect of software quality is validating or monitoring the source code to avoid the apparition of well known sources of errors and bugs. We work on how to best identify such common errors, by trying to identify earlier markers of possible errors, or by helping identifying common errors that programmers did in the past.

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 [81], [56] and classboxes [43] but also start to work on new areas such as isolation in dynamic languages. We will work on the following points: (1) Traits and (2) Modularization as a support for isolation.

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 [81], [56]. 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 [53], [75], [44], [57] and several type systems were defined [59], [82], [76], [69].

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 both to use for software developers and to implement for language designers.

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 [46]. 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 [56], stateful [45], and freezable [57]) 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. Traits may be flattened [74]. 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 [51] then from Smalltalk [62].

3.2.2. Reconciling Dynamic Languages and Isolation

Context and Problems. More and more applications require dynamic behavior such as modification of their own execution (often implemented using reflective features [66]). 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 or broken code, which may be problematic for the system if the application is not properly isolated. Bytecode checking and static code analysis are used to enable isolation, but such approaches do not really work in presence of dynamic languages and reflective features. Therefore there is a tension between the need for flexibility and isolation.

Research Agenda: Isolation in dynamic and reflective languages. To solve this tension, we will work on *Sure*, a language where isolation 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 [47], as well as controlling the access to reflective features [48], [49] are important challenges. We plan to:

- Study the isolation abstractions available in erights (http://www.erights.org) [73], [72], and Java's class loader strategies [68], [63].

- Categorize the different reflective features of languages such as CLOS [65], Python and Smalltalk [77] and identify suitable isolation mechanisms and infrastructure [60].

- Assess different isolation models (access rights, capabilities [78]...) 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 [47],

– the use encapsulation policies as a basis to restrict the interfaces of the controlled objects [80],

– the definition of method modifiers to support controlling encapsulation 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...) [77].

<div align="center">

SequeL Project-Team

</div>

3. Research Program

3.1. In Short

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-making 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. Reinforcement Learning

Sequential decision processes occupy the heart of the SEQUEL project; a detailed presentation of this problem may be found in Puterman's book [41].

A Markov Decision Process (MDP) is defined as the tuple $(\mathcal{X}, \mathcal{A}, P, r)$ where $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, P is the probabilistic transition kernel, and $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to I\!R$ 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 $x \in \mathcal{X}$ and the chosen action is $a \in \mathcal{A}$, then the Markov assumption means that the transition probability to a new state $x' \in \mathcal{X}$ (at time $t + 1$) only depends on (x, a). We write $p(x'|x, a)$ the corresponding transition probability. During a transition $(x, a) \to x'$, a reward $r(x, a, x')$ is incurred.

In the MDP $(\mathcal{X}, \mathcal{A}, P, r)$, each initial state x_0 and action sequence $a_0, a_1, ...$ gives rise to a sequence of states $x_1, x_2, ...$, satisfying $\mathbb{P}\left(x_{t+1} = x'|x_t = x, a_t = a\right) = p(x'|x, a)$, and rewards [1]$r_1, r_2, ...$ 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 $H_t = (x_0, a_0, ..., x_{t-1}, a_{t-1}, x_t)$. A policy π is a sequence of functions $\pi_0, \pi_1, ...$, where π_t maps the space of possible histories at time t to the space of probability distributions over the space of actions $\mathcal{A}$. To follow a policy means that, in each time step, we assume that the process history up to time t is $x_0, a_0, ..., x_t$ and the probability of selecting an action a is equal to $\pi_t(x_0, a_0, ..., x_t)(a)$. A policy is called stationary (or Markovian) if π_t depends only on the last visited state. In other words, a policy $\pi = (\pi_0, \pi_1, ...)$ is called stationary if $\pi_t(x_0, a_0, ..., x_t) = \pi_0(x_t)$ 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.

---

[1]Note that for simplicity, we considered the case of a deterministic reward function, but in many applications, the reward r_t itself is a random variable.

We move from an MD process to an MD problem by formulating the goal of the agent, that is what the sought policy π 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 π, we define the value function $V^\pi(x)$ of that policy π at a state $x \in \mathfrak{X}$ as the expected sum of discounted future rewards given that we state from the initial state x and follow the policy π:

$$V^\pi(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | x_0 = x, \pi\right], \tag{1}$$

where $\mathbb{E}$ is the expectation operator and $\gamma \in (0,1)$ is the discount factor. This value function V^π gives an evaluation of the performance of a given policy π. Other functionals of the sequence of future rewards may be considered, such as the undiscounted reward (see the stochastic shortest path problems [37]) 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) [35], 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 π is optimal, if it attains the optimal values $V^*(x)$ for any state $x \in \mathfrak{X}$, *i.e.*, if $V^\pi(x) = V^*(x)$ for all $x \in \mathfrak{X}$. Under mild conditions, deterministic stationary optimal policies exist [36]. Such an optimal policy is written π^*.

- We say that a (deterministic stationary) policy π is greedy with respect to (w.r.t.) some function V (defined on $\mathfrak{X}$) if, for all $x \in \mathfrak{X}$,

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

  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 $x \in \mathfrak{X}$,

$$V^*(x) = \max_{a \in \mathcal{A}} \sum_{x' \in \mathfrak{X}} p(x'|x,a)\left[r(x,a,x') + \gamma V^*(x')\right]. \tag{2}$$

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:

$$\pi^*(x) \in \arg\max_{a \in \mathcal{A}} \sum_{x' \in \mathfrak{X}} p(x'|x,a)\left[r(x,a,x') + \gamma V^*(x')\right]. \tag{3}$$

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 ([47]):

- 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^π of the policy π 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 challenges 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 π-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. *Multi-arm Bandit Theory*

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 [42], 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.* [34] 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. Prediction of Sequences of Structured and Unstructured Data

Given a series of observations $x_1, \cdots, x_n$ it is required to give forecasts concerning the distribution of the future observations $x_{n+1}, x_{n+2}, \cdots$; in the simplest case, that of the next outcome x_{n+1}. Then x_{n+1} 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 $x_1, \cdots, x_n, \cdots$, 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.

Another dimension of complexity in this problem concerns the nature of observations x_i. In the simplest case, they come from a finite space, but already basic applications often require real-valued observations. Moreover, function or even graph-valued observations often arise in practice, in particular in applications concerning Web data. In these settings estimating even simple characteristics of probability distributions of the future outcomes becomes non-trivial, and new learning algorithms for solving these problems are in order.

3.3.2. Hypothesis testing

Given a series of observations of $x_1, \cdots, x_n, \cdots$ generated by some unknown probability measure μ, the problem is to test a certain given hypothesis H_0 about μ, versus a given alternative hypothesis H_1. There are many different examples of this problem. Perhaps the simplest one is testing a simple hypothesis "μ is Bernoulli i.i.d. measure with probability of 0 equals 1/2" versus "μ 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 μ 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 as 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. Change Point Analysis

A stochastic process is generating the data. At some point, the process distribution changes. In the "offline" situation, the statistician observes the resulting sequence of outcomes and has to estimate the point or the points at which the change(s) occurred. In online setting, the goal is to detect the change as quickly as possible.

These are the classical problems in mathematical statistics, and probably among the last remaining statistical problems not adequately addressed by machine learning methods. The reason for the latter is perhaps in that the problem is rather challenging. Thus, most methods available so far are parametric methods concerning piece-wise constant distributions, and the change in distribution is associated with the change in the mean. However, many applications, including DNA analysis, the analysis of (user) behaviour data, etc., fail to comply with this kind of assumptions. Thus, our goal here is to provide completely non-parametric methods allowing for any kind of changes in the time-series distribution.

3.3.4. *Clustering Time Series, Online and Offline*

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 $x^1 = (x_1^1, \cdots, x_{n_1}^1), \cdots, x^N = (x_1^N, \cdots, x_{n_N}^N)$, we wish to 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.

The online version of the problem allows for the number of observed time series to grow with time, in general, in an arbitrary manner.

3.3.5. *Online Semi-Supervised Learning*

Semi-supervised learning (SSL) is a field of machine learning that studies learning from both labeled and unlabeled examples. This learning paradigm is extremely useful for solving real-world problems, where data is often abundant but the resources to label them are limited.

Furthermore, *online* SSL is suitable for adaptive machine learning systems. In the classification case, learning is viewed as a repeated game against a potentially adversarial nature. At each step t of this game, we observe an example $\mathbf{x_t}$, and then predict its label $\widehat{y}_t$.

The challenge of the game is that we only exceptionally observe the true label y_t. In the extreme case, which we also study, only a handful of labeled examples are provided in advance and set the initial bias of the system while unlabeled examples are gathered online and update the bias continuously. Thus, if we want to adapt to changes in the environment, we have to rely on indirect forms of feedback, such as the structure of data.

3.4. Statistical Learning and Bayesian Analysis

Before detailing some issues in these fields, let us remind the definition of a few terms.

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.

Statistical learning   is an approach to machine intelligence that 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.

Bayesian Analysis   applies to data that could be seen as observations in the more general meaning of the term. These data may not only come from classical sensors but also from any *device* recording information. From an operational point of view, like for statistical learning, uncertainty about the data is modeled by a probability measure thus defining the so-called likelihood functions. This last one depend upon parameters defining the state of the world we focus on for decision purposes. Within the Bayesian framework the uncertainty about these parameters is also modeled by probability measures, the priors that are subjective probabilities. Using probability theory and decision theory, one then defines new algorithms to estimate the parameters of interest and/or associated decisions. According to the International Society for Bayesian Analysis (source: http://bayesian.org), and from a more general point of view, this overall process could be

summarize as follows: one assesses the current state of knowledge regarding the issue of interest, gather new data to address remaining questions, and then update and refine their understanding to incorporate both new and old data. Bayesian inference provides a logical, quantitative framework for this process based on probability theory.

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 [39]. 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. *Non-parametric methods for 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. *Nonparametric Bayesian Estimation*

Numerous problems may be solved efficiently by 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 [38] 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 $G(d\theta) = \sum_{k=1}^{\infty} \omega_k \delta_{U_k}(d\theta)$, a Dirichlet process, we define a mixture of Dirichlet processes as:

$$F(x) = \int_{\Theta} f(x|\theta) G(d\theta) = \sum_{k=1}^{\infty} \omega_k f(x|U_k) \tag{4}$$

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 Mixture 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 their 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 $x_1, ... x_n$ are observed by several sensors which produce a collection of measurements $z_1, ..., z_m$ 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 be the optimal way to tackle the problem, since the sate 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 $X = \{x_1, ..., x_n\}$ and multi-sensor (measurement) set $Zk = \{z_1, ..., z_m\}$. 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:

$$f_{k+1|k}(X|Z^{(k)}) = \int f_{k+1|k}(X|W) f_{k|k}(W|Z^{(k)}) \delta W$$

$$f_{k+1|k+1}(X|Z^{(k+1)}) = \frac{f_{k+1}(Z_{k+1}|X) f_{k+1|k}(X|Z^{(k)})}{\int f_{k+1}(Z_{k+1}|W) f_{k+1|k}(W|Z^{(k)}) \delta W} \tag{5}$$

where:

- $X = \{x_1, ..., x_n\}$ is a multi-target state, *i.e.* a finite set of elements x_i defined on the single-target space $\mathfrak{X}$; [2]

- $Z_{k+1} = \{z_1, ..., z_m\}$ is the current multi-sensor observation, *i.e.* a collection of measures z_i produced at time $k + 1$ by all the sensors;

- $Z^{(k)} = \bigcup_{t \leqslant k} Z_t$ 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 Ξ is characterized by the family of its Janossy densities $j_{\Xi,1}(x_1)$, $j_{\Xi,2}(x_1, x_2)$... 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.

SHACRA Project-Team

3. Research Program

3.1. Real-time biophysical models

The principal objective of this scientific challenge is the modeling of the operative field, *i.e.* the anatomy and physiology of the patient that will be directly or indirectly targeted by a medical intervention. This requires to describe various phenomena such as soft-tissue deformation, fluid dynamics, electrical diffusion, or heat transfer. These models will help simulate the reaction of the patient's anatomy to the procedure, but also represent the behavior of complex organs such as the brain, the liver or the heart. A common requirement across these developments is the need for fast, possibly real-time, computation.

3.1.1. Real-time 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 an approach [3] 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 essentially rely on linearized constitutive laws, and are rarely validated. An important part of our research remains dedicated to the development of new, more accurate models that are 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.

3.1.2. Real-time biomechanical modeling of hollow structures

A large number of anatomical structures in the human body are vascularized (brain, liver, heart, kidneys, etc.) and recent interventions (such as interventional radiology procedures) rely on the vascular network as a therapeutical pathway. It is therefore essential to model the shape and deformable behavior of blood vessels. This can be done at two levels, depending of the objective. The global deformation of a vascular network can be represented using the vascular skeleton as a deformable (tree) structure, while local deformations need to be described using models of deformable surfaces. Other structures such as aneurysms, the colon or stomach can also benefit from being modeled as deformable surface, and we can rely on shell or thin plate theory to reach this objective.

3.1.3. Real-time blood flow

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. We are particularly interested in 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. A few studies have focused on aneurysm-related hemodynamics before and after endovascular coil embolization. As they rely on commercial software, the computation times (dozens of hours in general) are incompatible with interactive simulation or even clinical practice. Our objective is to propose new numerical approaches to reach (near) real-time computation of 3D flows without compromising the accuracy of the solution.

3.1.4. Real-time electrophysiology

Electrophysiology plays an important role in the physiology of the human body, for instance by inducing muscles motion, and obviously through the nervous system. Also, many clinical procedures rely on electrical stimulation, such as defibrillation, neuromuscular or deep brain stimulation for instance. Yet, the modeling and the simulation of this phenomenon is still in its early stages. Our primary objective is to focus on cardiac electrophysiology, which plays a critical role in the understanding of heart mechanisms, and also in the planning of certain cardiac procedures. We propose to develop models and computational strategies aimed at real-time simulation, and to also provide means to define patient-specific parameterizations of the model(s).

3.2. Interaction models

3.2.1. Constraint models and boundary conditions

To simulate soft-tissue deformations accurately, the modeling technique must account for the intrinsic behavior of the modeled organ as well as for its biomechanical interactions with surrounding tissues or medical devices. While the biomechanical behavior of important organs (such as the brain or liver) has been studied extensively in the past, only few works exist dealing with the mechanical interactions between the anatomical structures. For tissue–tool interactions, most techniques rely on simple contact models, whereas advanced phenomena such as friction are rarely taken into account. While simplifications can produce plausible results in the case of interaction between the manipulator of a laparoscopic instrument and the surface of an organ, it is generally an insufficient approximation. As we move towards the simulations for planning or rehearsal, accurate modeling of contacts is playing an increasingly important role. For instance, we have shown in [30] and [31] that complex interactions between a coil and aneurysm, or alternatively between a flexible needle and soft-tissue can be computed in real-time. In laparoscopic surgery, the main challenge is represented by modeling of interactions between anatomical structures rather than only between the instruments and the surface of the organ. Consequently, our objective was to model accurately the contacts with friction and other type on non-smooth interactions in a heterogeneous environment and to allow for stable haptic rendering. When different time integration strategies are used, another challenge is to compute the contact forces in such a way that integrity and stability of the overall simulation are maintained. Our objective was to propose a unified definition of such various boundary conditions and develop new numerical methods for simulations of heterogeneous objects.

3.2.2. Coupled biophysical systems

Research dealing with interactive medical simulations is currently limited to (bio-)mechanical aspects. However, an important step needs to be done to capture more precisely the complex nature of human organs such as liver or heart: the liver can be regarded as a composite structure made of parenchyma, vessels and a capsule, while a complete simulation of heart requires a coupled electro-mechanical model. Moreover, computing the interactions (or coupling) between anatomical structures can be useful for a simulation of larger systems; for instance we are investigating the modeling of connective tissues. Since the solutions to the above-mentioned problems usually lead to very large systems of equations, our strategy is based on approach similar to that used in domain-decomposition: instead of solving the large system at once, we propose techniques where one system per model is solved in order to improve the efficiency of solution procedures.

3.3. Towards pre-operative planning and per-operative guidance

Image-guided therapy is a recent area of research that has the potential to bridge the gap between medical imaging and clinical routine by adapting pre-operative data to the time of the procedure. Several challenges are typically related to image-guided therapy, such as multi-modality image registration, which serves to align pre-operative images onto the patient. As most procedures deal with soft-tissues, elastic registration techniques are necessary to perform this step. Novel registration techniques began to account for soft tissue deformation using physically-based methods. Yet, several limitations still hinder the use image-guided therapy in clinical routine. First, as registration methods become more complex, their computation time increases, thus lacking

responsiveness. Second, as we have seen previously, many factors influence the deformation of soft-tissues, from patient-specific material properties to boundary conditions with surrounding anatomy. Another very similar, and related, problem is augmented reality, i.e. the real-time superposition of a virtual model onto the reality. In a clinical context, this can be very useful to help "see through" the anatomy. In this case, however, real-time registration of the virtual information onto the patient is mandatory. Our objective in this area is to combine our expertise in real-time soft-tissue modeling, complex interactions with image data to provide accurate and real-time registration, deformation, and tracking of virtual anatomical structures onto the patient.

The predictive capabilities of computer simulations may also be used to improve minimally invasive surgical procedures. While simulation results are sensitive to model parameters, initial and boundary conditions, we aim at combining computer-vision algorithms and simulation algorithms in order to produce dynamic data-driven simulation in clinical applications. The main idea is to use computer-vision algorithms from pre-operative diagnoses or per-operative video streams in order to extract meaningful data to feed the simulation engine and thus to increase the accuracy of the simulation. Clinical outcomes are expected in interventional radiology where the guidance is based on fluoroscopic imaging modality inducing high absorbed dose of X-rays for the patient and the clinical staff. In that context, using the prediction capabilities of the simulation may decrease the acquisition frequency of images, leading to a lower exposure of X-rays. Our objective in this area is to combine our expertise in patient-specific modeling and constraint models to achieve the dynamic coupling between images, pre-operative data and computer simulation.

SIMPAF Project-Team

3. Research Program

3.1. General framework

Partial Differential Equations, Kinetic Equations, Conservation Laws, Hyperbolic Systems, Fluid Mechanics, Parabolic Systems, Computational Fluid Dynamics, Plasma Physics, Asymptotic analysis

The scientific activity of the project is concerned with Partial Differential Equations (PDE) 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 motivated the introduction of the so-called kinetic schemes.

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 and it has evolved with the composition of the team. One may think of:

- charged particles: description of semi-conductor devices or plasmas;

- bacteria, individuals or genes as in models motivated by biology or population dynamics;

- 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...

- cross-links in polymer chains to describe rubber elasticity;

- oxyde molecules to model corrosion phenomena at the miroscopic scale and derive effective macroscopic equations;

- cold atoms...

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.

The team has been composed in order to study these various aspects simultaneously. In particular, we wish to keep a balance between modelling, analysis, development of original codes and simulations.

3.2. Interactions of Micro- and Macroscopic Scales and Simulations

Statistical Physics, Homogenization, Asymptotic Preserving Schemes

3.2.1. *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 in 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).

3.2.2. *Statistical physics: dynamical friction, fluctuations and approach to equilibrium*

In models of charge transport, say transport of electrons, a phenomenological friction force is generally introduced, which is proportional to the velocity v. The dissipation induced by such a term is essential for the description of phenomena such as Ohm's law and approach to equilibrium. Our idea is to go back to a microscopic framework, with a description of the energy exchanges between the electrons and the surrounding medium which is the ultimate source of the dissipation of energy by the medium and of an effective friction force. We have shown numerically and argued theoretically that the balance between the fluctuations and the dissipation by the medium drives the particle to thermal equilibrium. The goal is now to provide rigorous proof of this statement. As a first step in this program, results will be obtained in an appropriate weak coupling limit. This program requires efforts in modelling, probability and analysis, but the questions are also really challenging for numerics, due, notably, to the large number of degrees of freedom involved in the equation. The subject is at the heart of the PhD work of É. Soret, now in her third year as a PhD student.

3.2.3. *Cold Atoms*

In the framework of the Labex CEMPI, C. Besse, S. De Bièvre and G. Dujardin are working, in collaboration with J.-C. Garreau and the cold-atom team at PhLAM, on the mathematical analysis and the numerical simulation of kicked rotor systems. Such systems are experimentally realized at PhLAM. A triple goal is being pursued: understand the effect of non-linearities on dynamical localization, understand dynamical localization in systems other than kicked rotors, and exploring the limits of the analogy between kicked systems and the Anderson model.

3.3. Finite element and finite volume methods

Conservation Laws, Anti-Diffusive Schemes, Viscous Flows, Control, Turbulence, Finite element methods, Finite volume methods

3.3.1. *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. Passive as well as active control were 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 combined active and passive control strategies in order to reach efficient results, especially concerning the drag coefficient, for two and three dimensional simulations [36], [37].

3.3.2. *Numerical Methods for Viscous Flows*

Numerical investigations are very useful to check the behavior of systems of equations modelling 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 [41] as well as its stability for high density ratios [38]. Now, C. Calgaro and E. Creusé, in collaboration with T. Goudon (Inria-COFFEE), have in mind the following objectives :

- Develop and freely distribute a new version of the matlab code (http://math.univ-lille1.fr/~simpaf/SITE-NS2DDV/ equipped with a graphic interface and an accurate documentation) to promote new collaborations in the domain, allow some easy comparisons with concurrent codes on the same benchmark cases, and compare alternative numerical solution methods (for instance to compare updating LU factorizations, see [40]);

- To generalize the stability results obtained in [39] for the scalar transport equation to the full 2D Euler system, in particular very low density values (near vacuum);

- Complete the C++ code to treat more general hydrodynamic models (combustion theory, transport of pollutants). We plan to check the behavior of the equations (typically the Kazhikhov-Smagulov model of powder-snow avanlanches) in the regime when the current existence theory does not apply, and extend our kinetic asymptotic-based schemes to such problems.

3.3.3. *A posteriori error estimators for finite element methods*

A posteriori estimates, finite element methods

We are interested in a posteriori error estimators for finite element methods, applied to the resolution of several partial differential equations. The objective is to obtain 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). There is a large bibliography database devoted to this topic, but a lot of problems remain to address. For example how to obtain explicit, sharp and robust bounds of the error ?

3.4. Numerical analysis of Schrödinger equations

Dispersive equations, Schrödinger equations

3.4.1. *Modelling of quantum dot-helium*

In collaboration with G. Reinish (Nice Observatory) and V. Guðmundsson (University of Reykjavik), C. Besse and G. 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 dimensions. This models quatum dot helium (*i.e.* the behavior of a pair of quantum electrons in a strong confining potential). 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, in order to be able to compare the competition between the Coulomb (repulsive) interaction and the binding (attractive) forces due to the confinement in this model as well as in other quantum mechanics models.

3.4.2. *Dispersive Schrödinger-like equations*

In collaboration with M. Taki (PhLAM laboratory, Lille), C. Besse and G. Dujardin are considering dispersive equations modelling the propagation of a laser beam in an optical 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).

3.4.3. *Absorbing boundary conditions for Korteweg de Vries equation*

The solution of the Korteweg de Vries equation is defined on an unbounded domain. If one wants to solve such a whole space evolution problem numerically, one has to restrict the computational domain by introducing artificial boundary conditions. So, the objective is to approximate the exact solution of the whole-space problem, restricted to a finite computational domain. This equation is difficult to study due to its third order space derivative. C. Besse end I. Lacroix-Violet, in collaboration with M. Ehrhardt, work on new developments.