RESEARCH CENTER
Lille - Nord Europe

FIELD

Activity Report 2015

Section Scientific Foundations

Edition: 2016-03-21

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 approach 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 consisted of extending a fixed base language (either C or PL/1 variant) 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 [5] 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, interpreters 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 [15].

BONSAI Project-Team

3. Research Program

3.1. Sequence processing for Next Generation Sequencing

As said in the introduction of this document, biological sequence analysis is a foundation subject for the team. In the last years, sequencing techniques have experienced remarkable advances with Next Generation Sequencing (NGS), that allow for fast and low-cost acquisition of huge amounts of sequence data, and outperforms conventional sequencing methods. These technologies can apply to genomics, with DNA sequencing, as well as to transcriptomics, with RNA sequencing. They promise to address a broad range of applications including: Comparative genomics, individual genomics, high-throughput SNP detection, identifying small RNAs, identifying mutant genes in disease pathways, profiling transcriptomes for organisms where little information is available, researching lowly expressed genes, studying the biodiversity in metagenomics. From a computational point of view, NGS gives rise to new problems and gives new insight on old problems by revisiting them: Accurate and efficient remapping, pre-assembling, fast and accurate search of non exact but quality labelled reads, functional annotation of reads, ...

3.2. Noncoding RNA

Our expertise in sequence analysis also applies to noncoding RNA. Noncoding RNA plays a key role in many cellular processes. First examples were given by microRNAs (miRNAs) that were initially found to regulate development in *C. elegans*, or small nucleolar RNAs (snoRNAs) that guide chemical modifications of other RNAs in mammals. Hundreds of miRNAs are estimated to be present in the human genome, and computational analysis suggests that more than 20% of human genes are regulated by miRNAs. To go further in this direction, the 2007 ENCODE Pilot Project provides convincing evidence that the Human genome is pervasively transcribed, and that a large part of this transcriptional output does not appear to encode proteins. All those observations open a universe of “RNA dark matter” that must be explored. From a combinatorial point of view, noncoding RNAs are complex objects. They are single stranded nucleic acid sequences that can fold forming long-range base pairings. This implies that RNA structures are usually modelled by complex combinatorial objects, such as ordered labeled trees, graphs or arc-annotated sequences.

3.3. Genome structures

Our third application domain is concerned with the structural organization of genomes. Genome rearrangements are able to change genome architecture by modifying the order of genes or genomic fragments. The first studies were based on linkage maps and fifteen year old mathematical models. But the usage of computational tools was still limited due to the lack of data. The increasing availability of complete and partial genomes now offers an unprecedented opportunity to analyse genome rearrangements in a systematic way and gives rise to a wide spectrum of problems: Taking into account several kinds of evolutionary events, looking for evolutionary paths conserving common structure of genomes, dealing with duplicated content, being able to analyse large sets of genomes even at the intraspecific level, computing ancestral genomes and paths transforming these genomes into several descendant genomes.

3.4. Nonribosomal peptides

Lastly, the team has been developing for several years a tight collaboration with Probiogem lab on nonribosomal peptides, and has became a leader on that topic. Nonribosomal peptide synthesis produces small peptides not going through the central dogma. As the name suggests, this synthesis uses neither messenger RNA nor ribosome but huge enzymatic complexes called nonribosomal peptide synthetases (NRPSs). This alternative pathway is found typically in bacteria and fungi. It has been described for the first time in the 70’s. For the last decade, the interest in nonribosomal peptides and their synthetases has considerably increased, as witnessed by the growing number of publications in this field. These peptides are or can be used in many biotechnological and pharmaceutical applications (e.g. anti-tumors, antibiotics, immuno-modulators).

DEFROST Team

3. Research Program

3.1. Introduction

Our research crosses different disciplines: numerical mechanics, control design, robotics, optimisation methods, clinical applications. Our organisation aims at facilitating the team work and cross- fertilisation of research results in the group. We have three objectives (1, 2 and 3) that correspond to the main scientific challenges. In addition, we have two transversal objectives that are also highly challenging: the development of a high performance software support for the project (objective 4) and the validation tools and protocols for the models and methods (objective 5).

3.2. Objective 1: Accurate model of soft-robot deformation computed in finite time

The objective is to find concrete numerical solutions to the challenge of modelling soft robots with strong real-time constraints. To solve continuum mechanics equations, we will start our research with real-time FEM or equivalent methods that were developed for soft-tissue simulation. We will extend the functionalities to account for the needs of a soft-robotic system:

- Coupling with other physical phenomenons that govern the activity of sensors and actuators (hydraulic, pneumatic, electro-active polymers, shape-memory alloys...).
- Fulfil the new computational time constraints (harder than surgical simulation for training) and find better tradeoff between cost and precision of numerical solvers using reduced-order modelling techniques with error control.
- Exploring interactive and semi-automatic optimisation methods for design based on obtained solution for fast computation on soft-robot models.

3.3. Objective 2: Model based control of soft-robot behaviour

The focus of this objective is on obtaining a generic methodology for soft-robot feedback control. Several steps are needed to design a model based control from FEM approach:

- The fundamental question of the kinematic link between actuators, sensors, effectors and contacts us- ing the most reduced mathematical space must be carefully addressed. We need to find efficient algo- rithms for real-time projection of non-linear FEM models in order to pose the control problem using the only relevant parameters of the motion control.
- Intuitive remote control is obtained when the user directly controls the effector motion. To add this functionality, we need to obtain real-time inverse models of the soft-robots by optimisation. Several criteria will be combined in this optimisation: effector motion control, structural stiffness of the robot, reduce intensity of the contact with the environment...
- Investigating closed-loop approaches using sensor feedback: as sensors cannot monitor all points of the deformable structure, the information provided will only be partial. We will need additional algorithms based on the FEM model to obtain the best possible treatment of the information. The final ob- jective of these models and algorithms is to have robust and efficient feedback control strategies for soft-robots. One of the main challenge here is to ensure / prove stability in closed-loop.

3.4. Objective 3: Modeling the interaction with a complex environment

Even if the inherent mechanical compliance of soft-robots makes them more safe, robust and particularly adapted to interaction with frag- ile environments, the contact forces need to be controlled by:

- Setting up real-time modelling and the control methods needed to pilot the forces that the robot imposes on its environment and to control the robot deformations imposed by its environment. Note that if an operative task requires to apply forces on the surrounding structures, the robot must be anchored to other structures or structurally rigidified.
- Providing mechanics models of the environment that include the uncertainties on the geometry and on the mechanical properties, and are capable of being readjusted in real-time.
- Using the visual feedback of the robot behaviour to adapt dynamically the models. The observation provided in the image coupled with an inverse accurate model of the robot could transform the soft robot into sensor: as the robot deforms with the contact of the surroundings, we could retrieve some missing parameters of the environment by a smart monitoring of the robot deformations.

3.5. Objective 4: Soft Robotic Software

Expected research results of this project are numerical methods and algorithms that require high-performance computing and suitability with robotic applications. There is no existing software support for such development. We propose to develop our own software, in a suite split into three applications:

- The first one will facilitate the design of deformable robots by an easy passage from CAD software (for the design of the robot) to the FEM based simulation
- The second one is an anticipative clinical simulator. The aim is to co-design the robotic assistance with the physicians, thanks to a realistic simulation of the procedure or the robotic assistance. This will facilitate the work of reflection on new clinical approaches prior any manufacturing
- The third one is the control design software. It will provide the real-time solutions for soft-robot control developed in the project.

3.6. Objective 5: Validation and application demonstrations

The implementation of experimental valida- tion is a key challenge for the project. On one side, we need to validate the model and control algorithms using concrete test case example in order to improve the modelling and to demonstrate the concrete feasibility of our methods. On the other side, concrete applications will also feed the reflexions on the objec- tives of the scientific program.

We will build our own experimental soft-robots for the validation of objective 2 and 3 when there is no existing « turn-key » solution. Designing and making our own soft-robots, even if only for validation, will help the setting-up of adequate models.

For the validation of objective 4, we will develop « anatomical soft robot »: soft-robot with the shape of organs, equipped with sensors (to measure the contact forces) and actuators (to be able to stiffen the walls and recreate natural motion of soft-tissues). We will progressively increase the level of realism of this novel validation set-up to come closer to the anatomical properties.

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 [69]. 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 [65], [68]. To be convinced one may refer to the list of references on Evolutionary Multi-objective Optimization maintained by Carlos A. Coello [0], 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 [71]. 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.

[0] http://delta.cs.cinvestav.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 [66].

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 [67], [70].

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 architectures 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.

DREAMPAL Project-Team

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 (∼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 [0]), 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.

We are developing a sofcore processor called HoMade (http://www.lifl.fr/~dekeyser/Homade) following these ideas.

In the multi-reflective softcores that we 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."*

[0] 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.

FUN Project-Team

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.

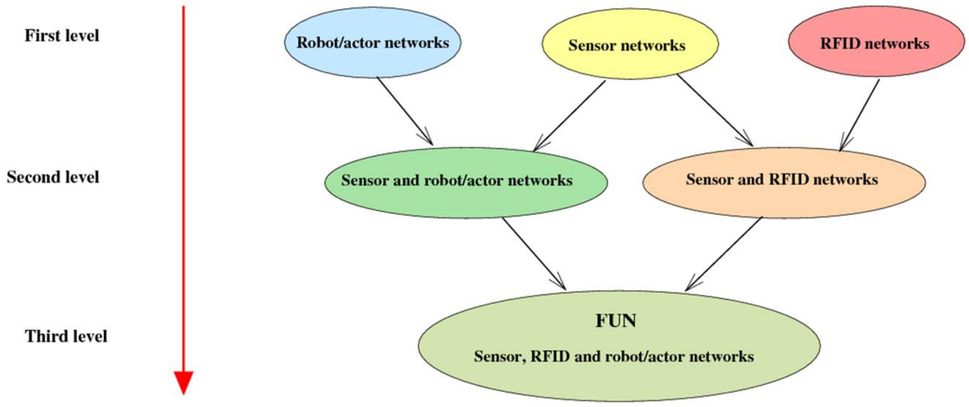

Figure 1. Panorama of FUN.

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 [40]) or self-stabilization [42] 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.

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 [47], FIT [46] and Aspire [32]. Next steps will be to experiment not only on testbeds but also on real use cases. These latter will be given through different collaborations.

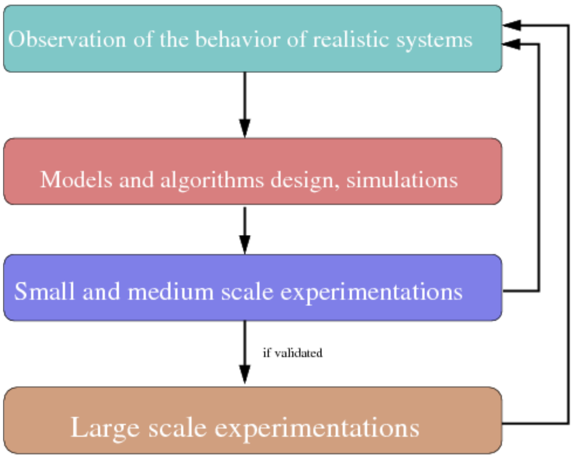
Observation of the behavior of realistic systems
Models and algorithms design, simulations
Small and medium scale experimentations
if validated
Large scale experimentations

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) [39]. 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.

INOCS Team

3. Research Program

3.1. Introduction

An optimization problem consists in finding a best solution from a set of feasible solutions. Such a problem can be typically modeled as a mathematical program in which decision variables must

1. satisfy a set of constraints that translate the feasibility of the solution and
2. optimize some (or several) objective function(s). Optimization problems are usually classified according to types of decision to be taken into strategic, tactical and operational problems.

We consider that an optimization problem presents a complex structure when it involves decisions of different types/nature (i.e. strategic, tactical or operational), and/or presenting some hierarchical leader-follower structure. The set of constraints may usually be partitioned into global constraints linking variables associated with the different types/nature of decision and constraints involving each type of variables separately. Optimization problems with a complex structure lead to extremely challenging problems since a global optimum with respect to the whole sets of decision variables and of constraints must be determined.

Significant progresses have been made in optimization to solve academic problems. Nowadays large-scale instances of some NP-Hard problems are routinely solved to optimality. *Our vision within INOCS is to make the same advances while addressing CS optimization problems*. To achieve this goal we aim to develop global solution approaches at the opposite of the current trend. INOCS team members have already proposed some successful methods following this research lines to model and solve CS problems (e.g. ANR project RESPET, Brotcorne *et al.* 2011, 2012, Gendron *et al.* 2009, Strack *et al.* 2009). However, these are preliminary attempts and a number of challenges regarding modeling and methodological issues have still to be met.

3.2. Modeling problems with complex structures

A classical optimization problem can be formulated as follows:

$$\begin{array}{ll} \min & f(x) \\ s.t. & x \in X, \end{array} \tag{1}$$

In this problem, X is the set of feasible solutions. Typically, in mathematical programming, X is defined by a set of constraints. x may be also limited to non-negative integer values.

INOCS team plan to address optimization problem where two types of decision are addressed jointly and are interrelated. More precisely, let us assume that variables x and y are associated with these decisions. A generic model for CS problems is the following:

$$\begin{array}{ll} \min & g(x,y) \\ s.t. & x \in X, \\ (x,y) & \in XY, \\ y & \in Y(x). \end{array} \tag{2}$$

In this model, X is the set of feasible values for x. XY is the set of feasible values for x and y jointly. This set is typically modeled through linking constraints. Last, $Y(x)$ is the set of feasible values for y for a given x. In INOCS, we do not assume that $Y(x)$ has any properties.

The INOCS team plans to model optimization CS problems according to three types of optimization paradigms: large scale complex structures optimization, bilevel optimization and robust/stochastic optimization. These paradigms instantiate specific variants of the generic model.

Large scale complex structures optimization problems can be formulated through the simplest variant of the generic model given above. In this case, it is assumed that $Y(x)$ does not depend on x. In such models, X and Y are associated with constraints on x and on y, XY are the linking constraints. x and y can take continuous or integer values. Note that all the problem data are deterministically known.

Bilevel programs allow the modeling of situations in which a decision-maker, hereafter the leader, optimizes his objective by taking explicitly into account the response of another decision maker or set of decision makers (the follower) to his/her decisions. Bilevel programs are closely related to Stackelberg (leader-follower) games as well as to the principal-agent paradigm in economics. In other words, bilevel programs can be considered as demand-offer equilibrium models where the demand is the result of another mathematical problem. Bilevel problems can be formulated through the generic CS model when $Y(x)$ corresponds to the optimal solutions of a mathematical program defined for a given x, i.e. $Y(x) = \mathrm{argmin}\,\{h(x,y)|y \in Y_2, (x,y) \in XY_2\}$ where Y_2 is defined by a set of constraints on y, and XY_2 is associated with the linking constraints.

In robust/stochastic optimization, it is assumed that the data related to a problem are subject to uncertainty. In stochastic optimization, probability distributions governing the data are known, and the objective function involves mathematical expectation(s). In robust optimization, uncertain data take value within specified sets, and the function to optimize is formulated in terms of a min-max objective typically (the solution must be optimal for the worst-case scenario). . A standard modeling of uncertainty on data is obtained by defining a set of possible scenarios that can be described explicitly or implicitly. In stochastic optimization, in addition, a probability of occurrence is associated with each scenario and the expected objective value is optimized.

3.3. Solving problems with complex structures

Standard solution methods developed for CS problems solve independent sub-problems associated with each type of variables without explicitly integrating their interactions or integrating them iteratively in a heuristic way. However these subproblems are intrinsically linked and should be addressed jointly. In *mathematicaloptimization* a classical approach is to approximate the convex hull of the integer solutions of the model by its linear relaxation. The main solution methods are i) polyhedral solution methods which strengthen this linear relaxation by adding valid inequalities, ii) decomposition solution methods (Dantzig Wolfe, Lagrangian Relaxation, Benders decomposition) which aim to obtain a better approximation and solve it by generating extreme points/rays. Main challenges are i) the analysis of the strength of the cuts and their separations for polyhedral solution methods, ii) the decomposition schemes and iii) the extreme points/rays generations for the decomposition solution methods.

The main difficulty in solving *bilevel problems* is due to their non convexity and non differentiability. Even linear bilevel programs, where all functions involved are affine, are computationally challenging despite their apparent simplicity . Up to now, much research has been devoted to bilevel problems with linear or convex follower problems. In this case, the problem can be reformulated as a single-level program involving complementarity constraints, exemplifying the dual nature, continuous and combinatorial, of bilevel programs.

LINKS Team

3. Research Program

3.1. Background

The main objective of LINKS is to develop methods for querying and managing linked data collections. Even though open linked data is the most prominent example, we will focus on hybrid linked data collections, which are collections of semi-structured datasets in hybrid formats: graph-based, RDF, relational, and NoSQL. The elements of these datasets may be linked, either by pointers or by additional relations between the elements of the different datasets, for instance the “same-as” or “member-of” relations as in RDF.

The advantage of traditional data models is that there exist powerful querying methods and technologies that one might want to preserve. In particular, they come with powerful schemas that constraint the possible manners in which knowledge is represented to a finite number of patterns. The exhaustiveness of these patterns is essential for writing of queries that cover all possible cases. Pattern violations are excluded by schema validation. In contrast, RDF schema languages such as RDFS can only enrich the relations of a dataset by new relations, which also helps for query writing, but which cannot constraint the number of possible patterns, so that they do not come with any reasonable notion of schema validation.

The main weakness of traditional formats, however, is that they do not scale to large data collections as stored on the Web, while the RDF data models scales well to very big collections such as linked open data. Therefore, our objective is to study mixed data collections, some of which may be in RDF format, in which we can lift the advantages of smaller datasets in traditional formats to much larger linked data collections. Such data collections are typically distributed over the internet, that some data sources have rigid query facilities that cannot be easily adapted or extended.

The main assumption that we impose in order to enable the logical approach, is that the given linked data collection must be correct in most dimensions. This means that all datasets are well-formed with respect to their available constraints and schemas, and clean with respect to the data values in most of the components of the relations in the datasets. One of the challenges is to integrate good quality RDF datasets into this setting, another is to clean the incorrect data in those dimensions that are less proper. It remains to be investigated in how far these assumptions can be maintained in realistic applications, and how much they can be weakened otherwise.

For querying linked data collections, the main problems are to resolve the heterogeneity of data formats and schemas, to understand the efficiency and expressiveness of recursive queries, that can follow links repeatedly, to answer queries under constraints, and to optimize query answering algorithms based on static analysis. When linked data is dynamically created, exchanged, or updated, the problems are how to process linked data incrementally, and how to manage linked data collections that change dynamically. In any case (static and dynamic) one needs to find appropriate schema mappings for linking semi-structured datasets. We will study how to automatize parts of this search process by developing symbolic machine learning techniques for linked data collections.

3.2. Querying Heterogeneous Linked Data

Our main objective is to query collections of linked datasets. In the static setting, we consider two kinds of links: explicit links between elements of the datasets, such as equalities or pointers, and logical links between relations of different datasets such as schema mappings. In the dynamic setting, we permit a third kind of links that point to “intentional” relations computable from a description, such as the application of a Web service or the application of a schema mapping.

We believe that collections of linked datasets are usually too big to ensure a global knowledge of all datasets. Therefore, schema mappings and constraints should remain between pairs of datasets. Our main goal is to be able to pose a query on a collection of datasets, while accounting for the possible recursive effects of schema mappings. For illustration, consider a ring of datasets D_1, D_2, D_3 linked by schema mappings M_1, M_2, M_3 that tell us how to complete a database D_i by new elements from the next database in the cycle.

The mappings M_i induce three intentional datasets I_1, I_2, and I_3, such that I_i contains all elements from D_i and all elements implied by M_i from the next intentional dataset in the ring:

$$I_1 = D_1 \cup M_1(I_2), \quad I_2 = D_2 \cup M_2(I_3), \quad I_3 = D_3 \cup M_3(I_1)$$

Clearly, the global information collected by the intentional datasets depends recursively on all three original datasets D_i. Queries to the global information can now be specified as standard queries to the intentional databases I_i. However, we will never materialize the intentional databases I_i. Instead, we can rewrite queries on one of the intentional datasets I_i to recursive queries on the union of the original datasets D_1, D_2, and D_3 with their links and relations. Therefore, a query answering algorithm is needed for recursive queries, that chases the “links” between the D_i in order to compute the part of I_i needed for the purpose of query answering.

This illustrates that we must account for the graph data models when dealing with linked data collections whose elements are linked, and that query languages for such graphs must provide recursion in order to chase links. Therefore, we will have to study graph databases with recursive queries, such as RDF graphs with SPARQL queries, but also other classes of graph databases and queries.

We study schemas and mappings between datasets with different kinds of data models and the complexity of evaluating recursive queries over graphs. In order to use schema mapping for efficiently querying the different datasets, we need to optimize the queries by taking into account the mappings. Therefore, we will study static analysis of schema mappings and recursive queries. Finally, we develop concrete applications in which our fundamental techniques can be applied.

3.3. Managing Dynamic Linked Data

With the quick growth of the information technology on the Web, more and more Web data gets created dynamically every day, for instance by smartphones, industrial machines, users of social networks, and all kinds of sensors. Therefore, large amounts of dynamic data need to be exchanged and managed by various data-centric web services, such as online shops, online newspapers, and social networks.

Dynamic data is often created by the application of some kind of service on the Web. This kind of data is intentional in the same spirit as the intentional data specified by the application of a schema mapping, or the application of some query to the hidden Web. Therefore, we will consider a third kind of links in the dynamic setting, that map to intentional data specified by whatever kind of function application. Such a function can be defined in data-centric programming languages, in the style of Active XML, XSLT, and NoSQL languages.

The dynamicity of data adds a further dimension to the challenges for linked data collections that we described before, while all the difficulties remain valid. One of the new aspects is that intentional data may be produced incrementally, as for instance when exchanged over data streams. Therefore, one needs incremental algorithms able to evaluate queries on incomplete linked data collections, that are extended or updated incrementally. Note that incremental data may be produced without end, such as a Twitter stream, so that one cannot wait for its completion. Instead, one needs to query and manage dynamic data with as low latency as possible. Furthermore, all static analysis problems are to be re-investigated in the presence of dynamic data.

Another aspect of dynamic data is distribution over the Web, and thus parallel processing as in the cloud. This raises the typical problems coming with data distribution: huge data sources cannot be moved without very high costs, while data must be replicated for providing efficient parallel access. This makes it difficult, if not impossible, to update replicated data consistently. Therefore, the consistency assumption has been removed by NoSQL databases for instance, while parallel algorithmic is limited to naive parallelization (i.e. map/reduce) where only few data needs to be exchanged.

We will investigate incremental query evaluation for distributed data-centered programming languages for linked data collections, dynamic updates as needed for linked data management, and static analysis for linked data workflows.

3.4. Linking Graphs

When datasets from independent sources are not linked with existing schema mappings, we would like to investigate symbolic machine learning solutions for inferring such mappings in order to define meaningful links between data from separate sources. This problem can be studied for various kinds of linked data collections. Before presenting the precise objectives, we will illustrate our approach on the example of linking data in two independent graphs: an address book of a research institute containing detailed personnel information and a (global) bibliographic database containing information on papers and their authors.

We remind that a schema allows to identify a collection of types each grouping objects from the same semantic class e.g., the collection of all persons in the address book and the collection of all authors in the bibliography database. As a schema is often lacking or underspecified in graph data models, we intend to investigate inference methods based on structural similarity of graph fragments used to describe objects from the same class in a given document e.g., in the bibliographic database every author has a name and a number of affiliations, while a paper has a title and a number of authors. Furthermore, our inference methods will attempt to identify, for every type, a set of possible keys, where by key we understand a collection of attributes of an object that uniquely identifies such an object in its semantic class. For instance, for a person in the address book two examples of a key are the name of the person and the office phone number of that person.

In the next step, we plan to investigate employing existing entity linkage solutions to identify pairs of types from different databases whose instances should be linked using compatible keys. For instance, persons in the address book should be linked with authors in the bibliographical database using the name as the compatible key. Linking the same objects (represented in different ways) in two databases can be viewed as an instance of a mapping between the two databases. Such mapping is, however, discriminatory because it typically maps objects from a specific subset of objects of given types. For instance, the mapping implied by linking persons in the address book with authors in the bibliographic database involves in fact researchers, a subgroup of personnel of the research institute, and authors affiliated with the research institute. Naturally, a subset of objects of a given type, or a subtype, can be viewed as a result of a query on the set of all objects, which on very basic level illustrates how learning data mappings can be reduced to learning queries.

While basic mappings link objects of the same type, more general mappings define how the same type of information is represented in two different databases. For instance, the email address and the postal address of an individual may be represented in one way in the address book and in another way in the bibliographic databases, and naturally, the query asking for the email address and the postal address of a person identified by a given name will differ from one database to the other. While queries used in the context of linking objects of compatible types are essentially unary, queries used in the context of linking information are n-ary and we plan to approach inference of general database mappings by investigating and employing algorithms for inference of n-ary queries.

An important goal in this research is elaborating a formal definition of *learnability* (feasibility of inference) of a given class of concepts (schemas of queries). We plan to following the example of Gold (1967), which requires not only the existence of an efficient algorithm that infers concepts consistent with the given input but the ability to infer every concept from the given class with a sufficiently informative input. Naturally, learnability depends on two parameters. The first parameter is the class of concepts i.e., a class of schema and a class of queries, from which the goal concept is to be inferred. The second parameter is the type of input that an inference algorithm is given. This can be a set of examples of a concept e.g., instances of RDF databases for which we wish to construct a schema or a selection of nodes that a goal query is to select. Alternatively, a more general interactive scenario can be used where the learning algorithm inquires the user about the goal concept e.g., by asking to indicate whether a given node is to be selected or not (as membership queries of Angluin (1987)). In general, the richer the input is, the richer class of concepts can be handled, however, the richer class of queries is to be handled, the higher computational cost is to be expected. The primary task is to find a good compromise and identify classes of concepts that are of high practical value, allow efficient inference with possibly simple type of input.

The main open problem for graph-shaped data studied by Links are how to infer queries, schemas, and schema-mappings for graph-structured data.

MAGNET Team

3. Research Program

3.1. Introduction

The main objective of MAGNET is to develop original machine learning methods for networked data in order to build applications like browsing, monitoring and recommender systems, and more broadly information extraction in information networks. We consider information networks in which the data consist of both feature vectors and texts. We model such 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 on-line and batch learning algorithms for various problems (node classification / clustering, link classification / prediction) which exploit the relationships between data entities and, overall, the graph topology. We are also interested in searching for the best hidden graph structure to be generated for solving a given learning task. Our research will be based 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, randomization and active learning can be leveraged to improve the scalability of the proposed algorithms.

Our research program is organized according to the following questions:

1. How to go beyond vectorial classification models in Natural Language Processing (NLP) tasks?
2. How to adaptively build graphs with respect to the given tasks? How to create networks from observations of information diffusion processes?
3. How to design methods able to achieve a good trade-off between predictive accuracy and computational complexity?
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. Interestingly, NLP research has been rather slow in coming to terms with this situation, and most of the literature 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 [33], [36].

A second motivation comes from the fact that most (if not all) NLP problems can be naturally conceived as graph problems. Thus, NLP 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 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 to, or at least complement, structured prediction methods (such as CRFs or structured SVMs) commonly used within NLP. 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 [12], [38].

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 performance for several NLP 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 [37].

Part of the challenge here 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, friendship, or thread membership. Unfortunately, modeling NLP 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. A preliminary discussion of the issue of optimal graph construction for semi-supervised learning in NLP is given in [12], [41]. We identify the issue of adaptive 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. 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 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 [38], [23]. Another related question that we will address in this context is how does one learn with *partial labels* (like partially specified tag sequence or tree) and use the graph structure to complete the output structure. This last question is very relevant to NLP 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 [25].

The NLP tasks we will mostly focus on are coreference resolution and entity linking, temporal structure prediction, and discourse parsing. These tasks will be envisioned 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 (see for instance [24], [19], [22]). 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 [40].

We have already initiated some work on the coreference resolution with graph-based learning, by casting the problem as an instance of spectral clustering [22].

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, making it possible to infer information related to each data sample even exploiting the graph topology solely. In fact, a widespread approach to several classification problems is to represent 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 [35], face recognition [21], and text categorization [28].

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 ([42], [13], [14]). 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. The first one is about how to choose the best similarity measure given the objective learning task. This question is related to the question of metric and similarity learning ([15], [16]) 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 adaptive 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 [30]. 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.

3.4. Prediction on Graphs and Scalability

As stated in the previous sections, graphs as complex objects provide 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 recovered 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 labeling 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 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 a smooth labeling function corresponding to an harmonic function on both manifolds in input and output.

Scalability is one of the main issues 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 computational complexity scales quadratically, or slower, in the number of considered data objects (usually nodes or edges, depending on the 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 [39].

At the level of 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 [29], 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 [17]. 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 [27] 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 [26] 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, for which we think nonparametric 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 also interested in studying generative models for graph labeling, exploiting the results obtained in p-stochastic model for link classification (see [20]) and statistical model for node label assignment which can be related to tree-structured Markov random fields [31].

3.5. Beyond Homophilic Relationships

In many cases, algorithms 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 ([18], [32]) 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 common interests is a key reason 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. 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, their spectral properties, 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 [34]. 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 allows to weigh the participation of each node to the relationship modeled by bipartite hyperedges accordingly to an equilibrium condition. This condition provides a competition between nodes in hyperedges and allows interesting modeling properties that go beyond homophily and similarity over nodes (the 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 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.

MEPHYSTO Team

3. Research Program

3.1. From statistical physics to continuum mechanics

Whereas numerical methods in nonlinear elasticity are well-developed and reliable, constitutive laws used for rubber in practice are phenomenological and generally not very precise. On the contrary, at the scale of the polymer-chain network, the physics of rubber is very precisely described by statistical physics. The main challenge in this field is to understand how to derive macroscopic constitutive laws for rubber-like materials from statistical physics.

At the continuum level, rubber is modelled by an energy E defined as the integral over a domain D of $\mathbb{R}^d$ of some energy density W depending only locally on the gradient of the deformation u: $E(u) = \int_D W(\nabla u(x))dx$. At the microscopic level (say 100nm), rubber is a network of cross-linked and entangled polymer chains (each chain is made of a sequence of monomers). At this scale the physics of polymer chains is well-understood in terms of statistical mechanics: monomers thermally fluctuate according to the Boltzmann distribution [55]. The associated Hamiltonian of a network is typically given by a contribution of the polymer chains (using self-avoiding random bridges) and a contribution due to steric effects (rubber is packed and monomers are surrounded by an excluded volume). The main challenge is to understand how this statistical physics picture yields rubber elasticity. Treloar assumed in [69] that for a piece of rubber undergoing some macroscopic deformation, the cross-links do not fluctuate and follow the macroscopic deformation, whereas between two cross-links, the chains fluctuate. This is the so-called affine assumption. Treloar's model is in rather good agreement with mechanical experiments in small deformation. In large deformation however, it overestimates the stress. A natural possibility to relax Treloar's model consists in relaxing the affine assumption while keeping the network description, which allows one to distinguish between different rubbers. This can be done by assuming that the deformation of the cross-links minimizes the free energy of the polymer chains, the deformation being fixed at the boundary of the macroscopic domain D. This gives rise to a “variational model". The analysis of the asymptotic behavior of this model as the typical length of a polymer chain vanishes has the same flavor as the homogenization theory of integral functionals in nonlinear elasticity (see [46], [63] in the periodic setting, and [47] in the random setting).

Our aim is to relate qualitatively and quantitatively the (precise but unpractical) statistical physics picture to explicit macroscopic constitutive laws that can be used for practical purposes.

In collaboration with R. Alicandro (Univ. Cassino, Italy) and M. Cicalese (Univ. Munich, Germany), A. Gloria analyzed in [1] the (asymptotic) Γ-convergence of the variational model for rubber, in the case when the polymer chain network is represented by some ergodic random graph. The easiest such graph is the Delaunay tessellation of a point set generated as follows: random hard spheres of some given radius ρ are picked randomly until the domain is jammed (the so-called random parking measure of intensity ρ). With M. Penrose (Univ. Bath, UK), A. Gloria studied this random graph in this framework [6]. With P. Le Tallec (Mechanics department, Ecole polytechnique, France), M. Vidrascu (project-team REO, Inria Paris-Rocquencourt), and A. Gloria introduced and tested in [57] a numerical algorithm to approximate the homogenized energy density, and observed that this model compares well to rubber elasticity qualitatively.

These preliminary results show that the variational model has the potential to explain qualitatively and quantitatively how rubber elasticity emerges from polymer physics. In order to go further and obtain more quantitative results and rigorously justify the model, we have to address several questions of analysis, modelling, scientific computing, inverse problems, and physics.

3.2. Quantitative stochastic homogenization

Whereas the approximation of homogenized coefficients is an easy task in periodic homogenization, this is a highly nontrivial task for stochastic coefficients. This is in order to analyze numerical approximation methods of the homogenized coefficients that F. Otto (MPI for mathematics in the sciences, Leipzig, Germany) and A. Gloria obtained the first quantitative results in stochastic homogenization [4]. The development of a complete stochastic homogenization theory seems to be ripe for the analysis and constitutes the second major objective of this section.

In order to develop a quantitative theory of stochastic homogenization, one needs to quantitatively understand the corrector equation (3). Provided A is stationary and ergodic, it is known that there exists a unique random field ϕ_ξ which is a distributional solution of (3) almost surely, such that $\nabla\phi_\xi$ is a stationary random field with bounded second moment $\left\langle|\nabla\phi_\xi|^2\right\rangle < \infty$, and with $\phi(0) = 0$. Soft arguments do not allow to prove that ϕ_ξ may be chosen stationary (this is wrong in dimension $d = 1$). In [4], [5] F. Otto and A. Gloria proved that, in the case of discrete elliptic equations with iid conductances, there exists a unique stationary corrector ϕ_ξ with vanishing expectation in dimension $d > 2$. Although it cannot be bounded, it has bounded finite moments of any order:

$$\langle|\phi_\xi|^q\rangle < \infty \text{ for all } q \geq 1. \tag{3}$$

They also proved that the variance of spatial averages of the energy density $(\xi + \nabla\phi_\xi) \cdot A(\xi + \nabla\phi_\xi)$ on balls of radius R decays at the rate R^{-d} of the central limit theorem. These are the *first optimal quantitative results* in stochastic homogenization.

The proof of these results, which is inspired by [64], is based on the insight that coefficients such as the Poisson random inclusions are special in the sense that the associated probability measure satisfies a spectral gap estimate. Combined with elliptic regularity theory, this spectral gap estimate quantifies ergodicity in stochastic homogenization. This systematic use of tools from statistical physics has opened the way to the quantitative study of stochastic homogenization problems, which we plan to fully develop.

3.3. Nonlinear Schrödinger equations

As well known, the (non)linear Schrödinger equation

$$\partial_t\varphi(t,x) = -\Delta\varphi(t,x) + \lambda V(x)\varphi(t,x) + g|\varphi|^2\varphi(t,x), \quad \varphi(0,x) = \varphi_0(x) \tag{4}$$

with coupling constants $g \in \mathbb{R}, \lambda \in \mathbb{R}_+$ and real potential V (possibly depending also on time) models many phenomena of physics.

When in the equation (5) above one sets $\lambda = 0, g \neq 0$, one obtains the nonlinear (focusing of defocusing) Schrödinger equation. It is used to model light propagation in optical fibers. In fact, it then takes the following form:

$$i\partial_z\varphi(t,z) = -\beta(z)\partial_t^2\varphi(t,z) + \gamma(z)|\varphi(t,z)|^2\varphi(z,t), \tag{5}$$

where β and γ are functions that characterize the physical properties of the fiber, t is time and z the position along the fiber. Several issues are of importance here. Two that will be investigated within the MEPHYSTO project are: the influence of a periodic modulation of the fiber parameters β and γ and the generation of so-called “rogue waves” (which are solutions of unusually high amplitude) in such systems.

If $g = 0, \lambda \neq 0$, V is a random potential, and φ_0 is deterministic, this is the standard random Schrödinger equation describing for example the motion of an electron in a random medium. The main issue in this setting is the determination of the regime of Anderson localization, a property characterized by the boundedness in time of the second moment $\int x^2 |\varphi(t,x)|^2 dx$ of the solution. If this second moment remains bounded in time, the solution is said to be localized. Whereas it is known that the solution is localized in one dimension for all (suitable) initial data, both localized and delocalized solutions exist in dimension 3 and it remains a major open problem today to prove this, cf. [52].

If now $g \neq 0, \lambda \neq 0$ and V is still random, but $|g| \ll \lambda$, a natural question is whether, and in which regime, one-dimensional Anderson localization perdures. Indeed, Anderson localization can be affected by the presence of the nonlinearity, which corresponds to an interaction between the electrons or atoms. Much numerical and some analytical work has been done on this issue (see for example [56] for a recent work at PhLAM, Laser physics department, Univ. Lille 1), but many questions remain, notably on the dependence of the result on the initial conditions, which, in a nonlinear system, may be very complex. The cold atoms team of PhLAM (Garreau-Szriftgiser) is currently setting up an experiment to analyze the effect of the interactions in a Bose-Einstein condensate on a closely related localization phenomenon called “dynamical localization”, in the kicked rotor, see below.

3.4. Processes in random environment

In the course of developing a quantitative theory of stochastic homogenization of discrete elliptic equations, we have introduced new tools to quantify ergodicity in partial differential equations. These tools are however not limited to PDEs, and could also have an impact in other fields where an evolution takes place in a (possibly dynamic) random environment and an averaging process occurs. The goal is then to understand the asymptotics of the motion of the particle/process.

For a random walker in a random environment, the Kipnis-Varadhan theorem ensures that the expected squared-position of the random walker after time t is of order t (the prefactor depends on the homogenized coefficients). If instead of a random walk among random conductances we consider a particle with some initial velocity evolving in a random *potential* field according to the Newton law, the averaged squared-position at time t is expected to follow the scaling law t^2, see [37]. This is called stochastic acceleration.

Similar questions arise when the medium is reactive (that is, when the potential is modified by the particle itself). The approach to equilibrium in such systems was observed numerically and explained theoretically, but not completely proven, in [50].

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 [23]. 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 [27]. Rapid and iterative prototyping techniques allow to decide between alternative solutions [24]. Controlled studies based on experimental or quasi-experimental designs can then be used to evaluate the chosen solutions [29]. 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 [26]. 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 (positional 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 foundations 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 [30], Dynamic Time Warping [32] 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 [28]; 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 [31].
- *mapping to application*, that studies how to link gesture inputs to application. This ranges from transfer function that is classically involved in pointing tasks [25], 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 analyse 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 lead 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 signals to accurately output the good forces.

Mjolnir Team

3. Research Program

3.1. Introduction

Our research program is organized around three main themes: leveraging human control skills, leveraging human perceptual skills, and leveraging human learning skills.

3.2. Leveraging human control skills

Our group has developed a unique and recognized expertise in *transfer functions*, i.e. the algorithmic transformations of raw user input for system use. Transfer functions define how user actions are taken into account by the system. They can make a task easier or impossible and thus largely condition user performance, no matter the criteria (speed, accuracy, comfort, fatigue, etc). Ideally, the transfer function should be chosen or tuned to match the interaction context. Yet the question of how to design a function to maximize one or more criteria in a given context remains an open one, and on-demand adaptation is difficult because functions are usually implemented at the lowest possible level to avoid latency problems. Latency management and transfer function design are two problems that require cross examination to improve human performance with interactive systems. Both also contribute to the senses of *initiation* and *control*, two crucial component of the sense of *agency* [33]. Our ultimate goal on these topics is to adapt the transfer function to the user and task in order to support stable and appropriate control. To achieve this, we investigate combinations of low-level (embedded) and high-level (application) ways to take user capabilities and task characteristics into account and reduce or compensate for latency in different contexts, e.g. using a mouse or a touchpad, a touch-screen, an optical finger navigation device or a brain-computer interface.

3.3. Leveraging human perceptual skills

Our work under this theme concerns the physicality of human-computer interaction, with a focus on haptic perception and related technologies, and the perception of animated displays.

Vibrators have long been used to provide basic kinesthetic feedback. Other piezoceramic and electro-active polymer technologies make it possible to support programmable friction or emboss a surface, and thin, organic technologies should soon provide transparent and conformable, flexible or stretchable substrates. We want to study the use of these different technologies for static and dynamic haptic feedback from both an engineering and an HCI perspective. We want to develop the tools and knowledge required to facilitate and inform the design of future haptic interactions taking best advantage of the different technologies.

Animations are increasingly common in graphical interfaces. Beyond their compelling nature, they are powerful tools that can be used to depict dynamic data, to help understand time-varying behaviors, to communicate a particular message or to capture attention. Yet despite their popularity, they are still largely undercomprehended as cognitive aids. While best practices provide useful directions, very little empirical research examine different types of animation, and their actual benefits and limitations remain to be determined. We want to increase current knowledge and develop the tools required to best take advantage of them.

3.4. Leveraging human learning skills

By looking at ways to leverage human control and perceptual skills, the research yet proposed mainly aims at improving perception-action coupling to better support transparent use. This third research theme addresses the different and orthogonal topic of skill acquisition and improvement. We want to move away from the usual binary distinction between “novices” and “experts” and explore means to promote and assist digital skill development in a more progressive fashion. We are interested in means to support the analytic use of computing tools. We want to help people become aware of the particular ways they use their tools, the other ways that exist for the things they do, and the other things they might do. We want to help them increase their performance by adjusting their current ways of doing, by providing new and more efficient ways, and by facilitating transitions from one way to another. We are also interested in means to foster reflection among users and facilitate the dissemination of best practices.

MODAL Project-Team

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 mixed data. It corresponds typically to continuous and categorical data but it includes also other widespread ones like ordinal, functional, rank,...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 mixed 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 of 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.

NON-A Project-Team

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 [0]. This involves linear/homogeneous/nonlinear state models, including ordinary equations, systems with delays, hybrid systems with impulses or switches [0], 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).*

[0] 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.

[0] 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.

RAPSODI Team

3. Research Program

3.1. Design and analysis of structure preserving schemes

3.1.1. Numerical analysis of nonlinear numerical methods

Up to now, the numerical methods dedicated to degenerate parabolic problems that the mathematicians are able to analyze almost all rely on the use of mathematical transformations (like e.g. the Kirchhoff's transform). It forbids the extension of the analysis to complex realistic models. The methods used in the industrial codes for solving such complex problems rely on the use of what we call NNM, i.e., on methods that preserve all the nonlinearities of the problem without reducing them thanks to artificial mathematical transforms. Our aim is to take advantage on the recent breakthrough proposed by C. Cancès & C. Guichard [15], [35] to develop efficient new numerical methods with a full numerical analysis (stability, convergence, error estimates, robustness w.r.t. physical parameters, ...).

3.1.2. Design and analysis of asymptotic preserving schemes

There has been an extensive effort in the recent years to develop numerical methods for diffusion equations that are robust with respect to heterogeneities, anisotropy, and the mesh (see for instance [55] for an extensive discussion on such methods). On the other hand, the understanding of the role of nonlinear stability properties in the asymptotic behaviors of dissipative systems increased significantly in the last decades (see for instance [50], [65]).

Recently, C. Chainais-Hillairet and co-authors [3], [7] and [17] developed a strategy based on the control of the numerical counterpart of the physical entropy to develop and analyze AP numerical methods. In particular, these methods show great promises for capturing accurately the behavior of the solutions to dissipative problems when some physical parameter is small with respect to the discretization characteristic parameters, or in the long-time asymptotic. Since it requires the use of nonlinear test functions in the analysis, strong restrictions on the physics (isotropic problems) and on the mesh (Cartesian grids, Voronoï boxes...) are required in [3], [7] and [17]. The schemes proposed in [15], [35] allow to handle nonlinear test functions in the analysis without restrictions on the mesh and on the anisotropy of the problem. Combining the nonlinear schemes *à la* [15] with the methodology of [3], [7], [17] would provide schemes that are robust both with respect to the meshes and to the parameters. Therefore, they would be also robust under adaptive mesh refinement.

3.1.3. Design and stability analysis of numerical methods for mixture problems

We aim at extending the range of the `NS2DDV-M` software by introducing new physical models, like for instance the Kazhikov and Smagulov model [61]. This will require a theoretical study for proving the existence of weak solutions to this model. Then, we will need to design numerical schemes to approximate these models and study their stability. We will also study their convergence following the path proposed in [58], [62].

3.2. Optimizing the computational efficiency

3.2.1. High order nonlinear numerical methods

The numerical experiments carried out in [15] show that in case of very strong anisotropy, the convergence of the proposed NNM becomes too slow (less than first order). Indeed, the method appears to strongly overestimate the dissipation. In order to make the method more competitive, it is necessary to estimate the dissipation in a more accurate way. Preliminary numerical results show that second order accuracy in space can be achieved in this way. One also aims to obtain (at least) second order accuracy in time without jeopardizing the stability. For many problems, this can be done by using so-called two-step backward differentiation formulas (BDF2) [56].

Concerning the inhomogeneous fluid models, we aim to investigate new methods for the mass equation resolution. Indeed, we aim at increasing the accuracy while maintaining some positivity-like properties and the efficiency for a wide range of physical parameters. To this end, we will consider *residual distribution* (RD) schemes, that appear as an alternative to finite volume methods. RD schemes enjoy very compact stencils. Therefore, their extension from 2D to 3D yield reasonable difficulties. These methods appeared twenty years ago, but recent extensions to unsteady problems [66], [59], with high-order accuracy [43], [42], or for parabolic problems [40], [41] make them very competitive. Relying on these breakthroughs, we aim at designing new RD schemes for fluid mixture models with high-order accuracy while preserving the positivity of the solutions.

3.2.2. A posteriori error control

The question of the *a posteriori* error estimators will also have to be addressed in this optimization context. Since the pioneering papers of Babuska and Rheinboldt more than thirty years ago [45], *a posteriori* error estimators have been widely studied. We will take advantage of the huge corresponding bibliography database in order to optimize our numerical results.

For example, we would like to generalize the results we derived for the harmonic magnetodynamic case (e.g. [9] and [51]) to the temporal magnetodynamic one, for which space/time *a posteriori* error estimators have to be developed. A space/time refinement algorithm should consequently be proposed and tested on academic as well as industrial benchmarks.

We also want to develop *a posteriori* estimators for the variable density Navier-Stokes model or some of its variants. To do so, several difficulties have to be tackled: the problem is nonlinear, unsteady, and the numerical method [5], [12] we developed combines features from finite elements and finite volumes. Fortunately, we do not start from scratch. Some recent references are devoted to the unsteady Navier-Stokes model in the finite element context [48], [68]. In the finite volume context, recent references deal with unsteady convection-diffusion equations [67], [44], [54] and [49]. We want to adapt some of these results to the variable density Navier-Stokes system, and to be able to design an efficient space-time remeshing algorithm.

RMOD Project-Team

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 [58], [57]. 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) [86] 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 [47]. 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 [74]. Until now, few works have attempted to identify layers in practice: Mudpie [88] is a first cut at identifying cycles between packages as well as package groups potentially representing layers. DSM (dependency structure matrix) [87], [82] 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 [70]. 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 [89], [61].

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 emergence 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 [84], [59] and classboxes [48] 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 [84], [59]. 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 [56], [78], [49], [60] and several type systems were defined [62], [85], [79], [72].

Traits are reusable building blocks that can be explicitly composed to share methods across unrelated class hierarchies. In their original form, traits do not contain state and cannot express visibility control for methods. Two extensions, stateful traits and freezable traits, have been proposed to overcome these limitations. However, these extensions are complex 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 [51]. 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 [59], stateful [50], and freezable [60]) 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 [77]. 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 [55] then from Smalltalk [65].

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 [69]). 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 [52], as well as controlling the access to reflective features [53], [54] are important challenges. We plan to:

- Study the isolation abstractions available in erights (http://www.erights.org) [76], [75], and Java's class loader strategies [71], [66].
- Categorize the different reflective features of languages such as CLOS [68], Python and Smalltalk [80] and identify suitable isolation mechanisms and infrastructure [63].
- Assess different isolation models (access rights, capabilities [81]...) 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 [52],

- the use of encapsulation policies as a basis to restrict the interfaces of the controlled objects [83],
- 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...) [80].

SEQUEL Project-Team

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 [46].

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 [0]$r_1, r_2, ...$ 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.

[0] 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 $x \in \mathcal{X}$ as the expected sum of discounted future rewards given that we state from the initial state x and follow the policy π:

$$V^{\pi}(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t | x_0 = x, \pi\right], \tag{6}$$

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 [45]) 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) [43], 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 $x \in \mathcal{X}$, *i.e.*, if $V^{\pi}(x) = V^*(x)$ for all $x \in \mathcal{X}$. Under mild conditions, deterministic stationary optimal policies exist [44]. 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 $\mathcal{X}$) if, for all $x \in \mathcal{X}$,

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

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

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

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

$$V^*(x) = \max_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} p(x'|x,a)\left[r(x,a,x') + \gamma V^*(x')\right]. \tag{7}$$

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

$$\pi^*(x) \in \arg\max_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} p(x'|x,a)\left[r(x,a,x') + \gamma V^*(x')\right]. \tag{8}$$

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 ([49]):

- 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 [47], 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.* [42] 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 piecewise 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 $x^1 = (x_1^1, \cdots, x_{n_1}^1), \cdots, x^N = (x_N^1, \cdots, x_{n_N}^N)$, 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.

SPIRALS Project-Team

3. Research Program

3.1. Introduction

Our research program on self-adaptive software targets two key properties that are detailed in the remainder of this section: *self-healing* and *self-optimization*.

3.2. Objective #1: Self-healing - Mining software artifacts to automatically evolve systems

Software systems are under the pressure of changes all along their lifecycle. Agile development blurs the frontier between design and execution and requires constant adaptation. The size of systems (millions of lines of code) multiplies the number of bugs by the same order of magnitude. More and more systems, such as sensor network devices, live in "surviving" mode, in the sense that they are neither rebootable nor upgradable.

Software bugs are hidden in source code and show up at development-time, testing-time or worse, once deployed in production. Except for very specific application domains where formal proofs are achievable, bugs can not be eradicated. As an order of magnitude, on 16 Dec 2011, the Eclipse bug repository contains 366,922 bug reports. Software engineers and developers work on bug fixing on a daily basis. Not all developers spend the same time on bug fixing. In large companies, this is sometimes a full-time role to manage bugs, often referred to as *Quality Assurance* (QA) software engineers. Also, not all bugs are equal, some bugs are analyzed and fixed within minutes, others may take months to be solved [92].

In terms of research, this means that: (i) one needs means to automatically adapt the design of the software system through automated refactoring and API extraction, (ii) one needs approaches to automate the process of adapting source code in order to fix certain bugs, (iii) one needs to revisit the notion of error-handling so that instead of crashing in presence of errors, software adapts itself to continue with its execution, *e.g.*, in degraded mode.

There is no one-size-fits-all solution for each of these points. However, we think that novel solutions can be found by using **data mining and machine learning techniques tailored for software engineering** [93]. This body of research consists of mining some knowledge about a software system by analyzing the source code, the version control systems, the execution traces, documentation and all kinds of software development and execution artifacts in general. This knowledge is then used within recommendation systems for software development, auditing tools, runtime monitors, frameworks for resilient computing, etc.

The novelty of our approach consists of using and tailoring data mining techniques for analyzing software artifacts (source code, execution traces) in order to achieve the **next level of automated adaptation** (*e.g.*, automated bug fixing). Technically, we plan to mix unsupervised statistical learning techniques (*e.g.* frequent item set mining) and supervised ones (*e.g.* training classifiers such as decision trees). This research is currently not being performed by data mining research teams since it requires a high level of domain expertise in software engineering, while software engineering researchers can use off-the-shelf data mining libraries, such as Weka [70].

We now detail the two directions that we propose to follow to achieve this objective.

3.2.1. Learning from software history how to design software and fix bugs

The first direction is about mining techniques in software repositories (*e.g.*, CVS, SVN, Git). Best practices can be extracted by data mining source code and the version control history of existing software systems. The design and code of expert developers significantly vary from the artifacts of novice developers. We will learn to differentiate those design characteristics by comparing different code bases, and by observing the semantic refactoring actions from version control history. Those design rules can then feed the test-develop-refactor constant adaptation cycle of agile development.

Fault localization of bugs reported in bug repositories. We will build a solid foundation on empirical knowledge about bugs reported in bug repository. We will perform an empirical study on a set of representative bug repositories to identify classes of bugs and patterns of bug data. For this, we will build a tool to browse and annotate bug reports. Browsing will be helped with two kinds of indexing: first, the tool will index all textual artifacts for each bug report; second it will index the semantic information that is not present by default in bug management software—*i.e.*, "contains a stacktrace"). Both indexes will be used to find particular subsets of bug reports, for instance "all bugs mentioning invariants and containing a stacktrace". Note that queries with this kind of complexity and higher are mostly not possible with the state-of-the-art of bug management software. Then, analysts will use annotation features to annotate bug reports. The main outcome of the empirical study will be the identification of classes of bugs that are appropriate for automated localization. Then, we will run machine learning algorithms to identify the latent links between the bug report content and source code features. Those algorithms would use as training data the existing traceability links between bug reports and source code modifications from version control systems. We will start by using decision trees since they produce a model that is explicit and understandable by expert developers. Depending on the results, other machine learning algorithms will be used. The resulting system will be able to locate elements in source code related to a certain bug report with a certain confidence.

Automated bug fix generation with search-based techniques. Once a location in code is identified as being the cause of the bug, we can try to automatically find a potential fix. We envision different techniques: (1) infer fixes from existing contracts and specifications that are violated; (2) infer fixes from the software behavior specified as a test suite; (3) try different fix types one-by-one from a list of identified bug fix patterns; (4) search fixes in a fix space that consists of combinations of atomic bug fixes. Techniques 1 and 2 are explored in [64] and [91]. We will focus on the latter techniques. To identify bug fix patterns and atomic bug fixes, we will perform a large-scale empirical study on software changes (also known as changesets when referring to changes across multiple files). We will develop tools to navigate, query and annotate changesets in a version control system. Then, a grounded theory will be built to master the nature of fixes. Eventually, we will decompose change sets in atomic actions using clustering on changeset actions. We will then use this body of empirical knowledge to feed search-based algorithms (*e.g.* genetic algorithms) that will look for meaningful fixes in a large fix space. To sum up, our research on automated bug fixing will try not only to point to source code locations responsible of a bug, but to search for code patterns and snippets that may constitute the skeleton of a valid patch. Ultimately, a blend of expert heuristics and learned rules will be able to produce valid source code that can be validated by developers and committed to the code base.

3.2.2. Run-time self-healing

The second proposed research direction is about inventing a self-healing capability at run-time. This is complementary to the previous objective that mainly deals with development time issues. We will achieve this in two steps. First, we want to define frameworks for resilient software systems. Those frameworks will help to maintain the execution even in the presence of bugs—*i.e.* to let the system survive. As exposed below, this may mean for example to switch to some degraded modes. Next, we want to go a step further and to define solutions for automated runtime repair, that is, not simply compensating the erroneous behavior, but also determining the correct repair actions and applying them at run-time.

Mining best effort values. A well-known principle of software engineering is the "fail-fast" principle. In a nutshell, it states that as soon as something goes wrong, software should stop the execution before entering incorrect states. This is fine when a human user is in the loop, capable of understanding the error or at least rebooting the system. However, the notion of "failure-oblivious computing" [83] shows that in certain domains, software should run in a resilient mode (*i.e.* capable of recovering from errors) and/or best-effort mode—*i.e.* a slightly imprecise computation is better than stopping. Hence, we plan to investigate data mining techniques in order to learn best-effort values from past executions (*i.e.* somehow learning what is a correct state, or the opposite what is not a completely incorrect state). This knowledge will then be used to adapt the software state and flow in order to mitigate the error consequences, the exact opposite of fail-fast for systems with long-running cycles.

Embedding search based algorithms at runtime. Harman recently described the field of search-based software engineering [71]. We think that certain search based approaches can be embedded at runtime with the goal of automatically finding solutions that avoid crashing. We will create software infrastructures that allow automatically detecting and repairing faults at run-time. The methodology for achieving this task is based on three points: (1) empirical study of runtime faults; (2) learning approaches to characterize runtime faults; (3) learning algorithms to produce valid changes to the software runtime state. An empirical study will be performed to analyze those bug reports that are associated with runtime information (*e.g.* core dumps or stacktraces). After this empirical study, we will create a system that learns on previous repairs how to produce small changes that solve standard runtime bugs (*e.g.* adding an array bound check to throw a handled domain exception rather than a spurious language exception). To achieve this task, component models will be used to (1) encapsulate the monitoring and reparation meta-programs in appropriate components and (2) support runtime code modification using scripting, reflective or bytecode generation techniques.

3.3. Objective #2: Self-optimization - Sharing runtime behaviors to continuously adapt software

Complex distributed systems have to seamlessly adapt to a wide variety of deployment targets. This is due to the fact that developers cannot anticipate all the runtime conditions under which these systems are immersed. A major challenge for these software systems is to develop their capability to continuously reason about themselves and to take appropriate decisions and actions on the optimizations they can apply to improve themselves. This challenge encompasses research contributions in different areas, from environmental monitoring to real-time symptoms diagnosis, to automated decision making. The variety of distributed systems, the number of optimization parameters, and the complexity of decisions often resign the practitioners to design monolithic and static middleware solutions. However, it is now globally acknowledged that the development of dedicated building blocks does not contribute to the adoption of sustainable solutions. This is confirmed by the scale of actual distributed systems, which can—for example—connect several thousands of devices to a set of services hosted in the Cloud. In such a context, the lack of support for smart behaviours at different levels of the systems can inevitably lead to its instability or its unavailability. In June 2012, an outage of Amazon's Elastic Compute Cloud in North Virginia has taken down Netflix, Pinterest, and Instagram services. During hours, all these services failed to satisfy their millions of customers due to the lack of integration of a self-optimization mechanism going beyond the boundaries of Amazon.

The research contributions we envision within this area will therefore be organized as a reference model for engineering **self-optimized distributed systems** autonomously driven by *adaptive feedback control loops*, which will automatically enlarge their scope to cope with the complexity of the decisions to be taken. This solution introduces a multi-scale approach, which first privileges local and fast decisions to ensure the homeostasis [0] property of a single node, and then progressively propagates symptoms in the network in order to reason on a longer term and a larger number of nodes. Ultimately, domain experts and software developers can be automatically involved in the decision process if the system fails to find a satisfying solution. The research program for this objective will therefore focus on the study of mechanisms for **monitoring, taking decisions, and automatically reconfiguring software at runtime and at various scales**. As stated in the self-healing objective, we believe that there is no one-size-fits-all mechanism that can span all the scales of the system. We will therefore study and identify an optimal composition of various adaptation mechanisms in order to produce long-living software systems.

The novelty of this objective is to exploit the wisdom of crowds to define new middleware solutions that are able to continuously adapt software deployed in the wild. We intend to demonstrate the applicability of this approach to distributed systems that are deployed from mobile phones to cloud infrastructures. The key scientific challenges to address can be summarized as follows: *How does software behave once deployed in the wild? Is it possible to automatically infer the quality of experience, as it is perceived by users? Can the*

[0] Homeostasis is the property of a system that regulates its internal environment and tends to maintain a stable, relatively constant condition of properties [Wikipedia].

runtime optimizations be shared across a wide variety of software? How optimizations can be safely operated on large populations of software instances?

The remainder of this section further elaborates on the opportunities that can be considered within the frame of this objective.

3.3.1. Monitoring software in the wild

Once deployed, developers are generally no longer aware of how their software behave. Even if they heavily use testbeds and benchmarks during the development phase, they mostly rely on the bugs explicitly reported by users to monitor the efficiency of their applications. However, it has been shown that contextual artifacts collected at runtime can help to understand performance leaks and optimize the resilience of software systems [94]. Monitoring and understanding the context of software at runtime therefore represent the first building block of this research challenge. Practically, we intend to investigate crowd-sensing approaches, to smartly collect and process runtime metrics (*e.g.*, request throughput, energy consumption, user context). Crowd-sensing can be seen as a specific kind of crowdsourcing activity, which refers to the capability of lifting a (large) diffuse group of participants to delegate the task of retrieving trustable data from the field. In particular, crowd-sensing covers not only *participatory sensing* to involve the user in the sensing task (*e.g.*, surveys), but also *opportunistic sensing* to exploit mobile sensors carried by the user (*e.g.*, smartphones).

While reported metrics generally enclose raw data, the monitoring layer intends to produce meaningful indicators like the *Quality of Experience* (QoE) perceived by users. This QoE reflects representative symptoms of software requiring to trigger appropriate decisions in order to improve its efficiency. To diagnose these symptoms, the system has to process a huge variety of data including runtime metrics, but also history of logs to explore the sources of the reported problems and identify opportunities for optimizations. The techniques we envision at this level encompass machine learning, principal component analysis, and fuzzy logic [81] to provide enriched information to the decision level.

3.3.2. Collaborative decision-making approaches

Beyond the symptoms analysis, decisions should be taken in order to improve the *Quality of Service* (QoS). In our opinion, collaborative approaches represent a promising solution to effectively converge towards the most appropriate optimization to apply for a given symptom. In particular, we believe that exploiting the wisdom of the crowd can help the software to optimize itself by sharing its experience with other software instances exhibiting similar symptoms. The intuition here is that the body of knowledge that supports the optimization process cannot be specific to a single software instance as this would restrain the opportunities for improving the quality and the performance of applications. Rather, we think that any software instance can learn from the experience of others.

With regard to the state-of-the-art, we believe that a multi-levels decision infrastructure, inspired from distributed systems like Spotify [66], can be used to build a decentralized decision-making algorithm involving the surrounding peers before requesting a decision to be taken by more central control entity. In the context of collaborative decision-making, peer-based approaches therefore consist in quickly reaching a consensus on the decision to be adopted by a majority of software instances. Software instances can share their knowledge through a micro-economic model [63], that would weight the recommendations of experienced instances, assuming their age reflects an optimal configuration.

Beyond the peer level, the adoption of algorithms inspired from evolutionary computations, such as genetic programming, at an upper level of decision can offer an opportunity to test and compare several alternative decisions for a given symptom and to observe how does the crowd of applications evolves. By introducing some diversity within this population of applications, some instances will not only provide a satisfying QoS, but will also become naturally resilient to unforeseen situations.

3.3.3. Smart reconfigurations in the large

Any decision taken by the crowd requires to propagate back to and then operated by the software instances. While simplest decisions tend to impact software instances located on a single host (*e.g.*, laptop, smartphone), this process can also exhibit more complex reconfiguration scenarios that require the orchestration of various actions that have to be safely coordinated across a large number of hosts. While it is generally acknowledged that centralized approaches raise scalability issues, we think that self-optimization should investigate different reconfiguration strategies to propagate and apply the appropriate actions. The investigation of such strategies can be addressed in two steps: the consideration of *scalable data propagation protocols* and the identification of *smart reconfiguration mechanisms*.

With regard to the challenge of scalable data propagation protocols, we think that research opportunities encompass not only the exploitation of gossip-based protocols [65], but also the adoption of publish/subscribe abstractions [73] in order to decouple the decision process from the reconfiguration. The fundamental issue here is the definition of a communication substrate that can accommodate the propagation of decisions with relaxed properties, inspired by *Delay Tolerant Networks* (DTN), in order to reach weakly connected software instances. We believe that the adoption of asynchronous communication protocols can provide the sustainable foundations for addressing various execution environments including harsh environments, such as developing countries, which suffer from a partial connectivity to the network. Additionally, we are interested in developing the principle of *social networks of applications* in order to seamlessly group and organize software instances according to their similarities and acquaintances. The underlying idea is that grouping application instances can contribute to the identification of optimization profiles not only contributing to the monitoring layer, but also interested in similar reconfigurations. Social networks of applications can contribute to the anticipation of reconfigurations by exploiting the symptoms of similar applications to improve the performance of others before that problems actually happen.

With regard to the challenge of smart reconfiguration mechanisms, we are interested in building on our established experience of adaptive middleware [8] in order to investigate novel approaches to efficient application reconfigurations. In particular, we are interested in adopting seamless micro-updates and micro-reboot techniques to provide in-situ reconfiguration of pieces of software. Additionally, the provision of safe and secured reconfiguration mechanisms is clearly a key issue that requires to be carefully addressed in order to avoid malicious exploitation of dynamic reconfiguration mechanisms against the software itself. In this area, although some reconfiguration mechanisms integrate transaction models [74], most of them are restricted to local reconfigurations, without providing any support for executing distributed reconfiguration transactions. Additionally, none of the approached published in the literature include security mechanisms to preserve from unauthorized or malicious reconfigurations.