RESEARCH CENTER
Rennes - Bretagne-Atlantique

FIELD

Activity Report 2018

Section Scientific Foundations

Edition: 2019-03-07

ALGORITHMICS, PROGRAMMING, SOFTWARE AND ARCHITECTURE

APPLIED MATHEMATICS, COMPUTATION AND SIMULATION

DIGITAL HEALTH, BIOLOGY AND EARTH

NETWORKS, SYSTEMS AND SERVICES, DISTRIBUTED COMPUTING

PERCEPTION, COGNITION AND INTERACTION

CAIRN Project-Team

3. Research Program

3.1. Panorama

The development of complex applications is traditionally split in three stages: a theoretical study of the algorithms, an analysis of the target architecture and the implementation. When facing new emerging applications such as high-performance, low-power and low-cost mobile communication systems or smart sensor-based systems, it is mandatory to strengthen the design flow by a joint study of both algorithmic and architectural issues.

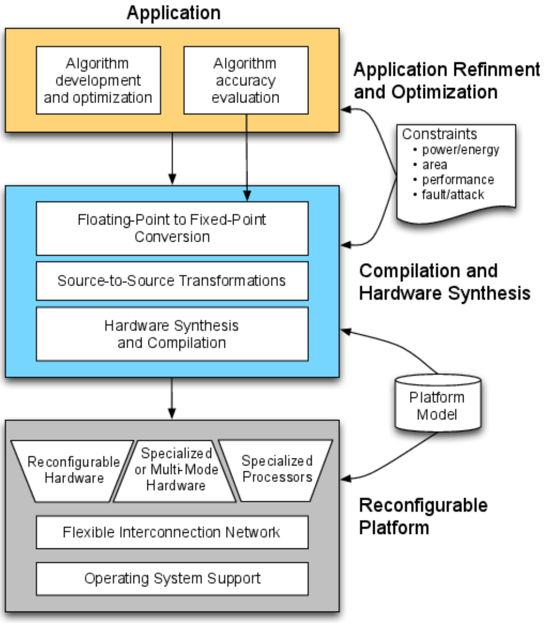
Application
Algorithm development and optimization
Algorithm accuracy evaluation
Application Refinment and Optimization
Constraints
• power/energy
• area
• performance
• fault/attack
Floating-Point to Fixed-Point Conversion
Source-to-Source Transformations
Hardware Synthesis and Compilation
Compilation and Hardware Synthesis
Platform Model
Reconfigurable Hardware
Specialized or Multi-Mode Hardware
Specialized Processors
Flexible Interconnection Network
Operating System Support
Reconfigurable Platform

Figure 1. CAIRN's general design flow and related research themes

Figure 1 shows the global design flow we propose to develop. This flow is organized in levels which refer to our three research themes: application optimization (new algorithms, fixed-point arithmetic, advanced representations of numbers), architecture optimization (reconfigurable and specialized hardware, application-specific processors, arithmetic operators and functions), and stepwise refinement and code generation (code transformations, hardware synthesis, compilation).

In the rest of this part, we briefly describe the challenges concerning **new reconfigurable platforms** in Section 3.2 and the issues on **compiler and synthesis tools** related to these platforms in Section 3.3 .

3.2. Reconfigurable Architecture Design

Nowadays, FPGAs are not only suited for application specific algorithms, but also considered as fully-featured computing platforms, thanks to their ability to accelerate massively parallelizable algorithms much faster than their processor counterparts [78]. They also support to be dynamically reconfigured. At runtime, partially reconfigurable regions of the logic fabric can be reconfigured to implement a different task, which allows for a better resource usage and adaptation to the environment. Dynamically reconfigurable hardware can also cope with hardware errors by relocating some of its functionalities to another, sane, part of the logic fabric. It could also provide support for a multi-tasked computation flow where hardware tasks are loaded on-demand at runtime. Nevertheless, current design flows of FPGA vendors are still limited by the use of one partial bitstream for each reconfigurable region and for each design. These regions are defined at design time and it is not possible to use only one bitstream for multiple reconfigurable regions nor multiple chips. The multiplicity of such bitstreams leads to a significant increase in memory. Recent research has been conducted in the domain of task relocation on a reconfigurable fabric. All of the related work was conducted on architectures from commercial vendors (e.g., Xilinx, Altera) which share the same limitations: the inner details of the bitstream are not publicly known, which limits applicability of the techniques. To circumvent this issue, most dynamic reconfiguration techniques are either generating multiple bitstreams for each location [62] or implementing an online filter to relocate the tasks [72]. Both of these techniques still suffer from memory footprint and from the online complexity of task relocation.

Increasing the level and grain of reconfiguration is a solution to counterbalance the FPGA penalties. Coarse-grained reconfigurable architectures (CGRA) provide operator-level configurable functional blocks and word-level datapaths [79], [67], [77]. Compared to FPGA, they benefit from a massive reduction in configuration memory and configuration delay, as well as for routing and placement complexity. This in turns results in an improvement in the computation volume over energy cost ratio, although with a loss of flexibility compared to bit-level operations. Such constraints have been taken into account in the design of DART[9], Adres [75] or polymorphous computing fabrics[11]. These works have led to commercial products such as the PACT/XPP [61] or Montium from Recore systems, without however a real commercial success yet. Emerging platforms like Xilinx/Zynq or Intel/Altera are about to change the game.

In the context of emerging heterogenous multicore architecture, CAIRN advocates for associating general-purpose processors (GPP), flexible network-on-chip and coarse-grain or fine-grain dynamically reconfigurable accelerators. We leverage our skills on microarchitecture, reconfigurable computing, arithmetic, and low-power design, to discover and design such architectures with a focus on: reduced energy per operation; improved application performance through acceleration; hardware flexibility and self-adaptive behavior; tolerance to faults, computing errors, and process variation; protections against side channel attacks; limited silicon area overhead.

3.3. Compilation and Synthesis for Reconfigurable Platforms

In spite of their advantages, reconfigurable architectures, and more generally hardware accelerators, lack efficient and standardized compilation and design tools. As of today, this still makes the technology impractical for large-scale industrial use. Generating and optimizing the mapping from high-level specifications to reconfigurable hardware platforms are therefore key research issues, which have received considerable interest over the last years [65], [80], [76], [74], [73]. In the meantime, the complexity (and heterogeneity) of these platforms has also been increasing quite significantly, with complex heterogeneous multi-cores architectures becoming a *de facto* standard. As a consequence, the focus of designers is now geared toward optimizing overall system-level performance and efficiency [71]. Here again, existing tools are not well suited, as they fail at providing a unified programming view of the programmable and/or reconfigurable components implemented on the platform.

In this context, we have been pursuing our efforts to propose tools whose design principles are based on a tight coupling between the compiler and the target hardware architectures. We build on the expertise of the team members in High Level Synthesis (HLS) [5], ASIP optimizing compilers [12] and automatic parallelization for massively parallel specialized circuits [2]. We first study how to increase the efficiency of standard programmable processors by extending their instruction set to speed-up compute intensive kernels. Our focus is on efficient and exact algorithms for the identification, selection and scheduling of such instructions [6]. We address compilation challenges by borrowing techniques from high-level synthesis, optimizing compilers and automatic parallelization, especially when dealing with nested loop kernels. In addition, and independently of the scientific challenges mentioned above, proposing such flows also poses significant software engineering issues. As a consequence, we also study how leading edge software engineering techniques (Model Driven Engineering) can help the Computer Aided Design (CAD) and optimizing compiler communities prototyping new research ideas [4].

Efficient implementation of multimedia and signal processing applications (in software for DSP cores or as special-purpose hardware) often requires, for reasons related to cost, power consumption or silicon area constraints, the use of fixed-point arithmetic, whereas the algorithms are usually specified in floating-point arithmetic. Unfortunately, fixed-point conversion is very challenging and time-consuming, typically demanding up to 50% of the total design or implementation time. Thus, tools are required to automate this conversion. For hardware or software implementation, the aim is to optimize the fixed-point specification. The implementation cost is minimized under a numerical accuracy or an application performance constraint. For DSP-software implementation, methodologies have been proposed [7] to achieve fixed-point conversion. For hardware implementation, the best results are obtained when the word-length optimization process is coupled with the high-level synthesis [68]. Evaluating the effects of finite precision is one of the major and often the most time consuming step while performing fixed-point refinement. Indeed, in the word-length optimization process, the numerical accuracy is evaluated as soon as a new word-length is tested, thus, several times per iteration of the optimization process. Classical approaches are based on fixed-point simulations [69]. Leading to long evaluation times, they can hardly be used to explore the design space. Therefore, our aim is to propose closed-form expressions of errors due to fixed-point approximations that are used by a fast analytical framework for accuracy evaluation [10].

CELTIQUE Project-Team (section vide)

CIDRE Project-Team

3. Research Program

3.1. Our perspective

For many aspects of our daily lives, we rely heavily on computer systems, many of which are based on massively interconnected devices that support a population of interacting and cooperating entities. As these systems become more open and complex, accidental and intentional failures become much more frequent and serious. We believe that the purpose of attacks against these systems is expressed at a high level (compromise of sensitive data, unavailability of services). However, these attacks are often carried out at a very low level (exploitation of vulnerabilities by malicious code, hardware attacks).

The CIDRE team is specialized in the defense of computer systems. We argue that to properly protect these systems we must have a complete understanding of the attacker's concrete capabilities. In other words, **to defend properly we must understand the attack**.

The CIDRE team therefore strives to have a global expertise in information systems: from hardware to distributed architectures. Our objective is to highlight security issues and propose preventive or reactive countermeasures in widely used and privacy-friendly systems.

3.2. Attack Comprehension

The first step before being able to offer secure systems is to understand and measure the real capabilities of the attacker. It's a cat and mouse game and in this game, the attacker is always one step ahead of the defender. The attacker is able to exploit for his own benefit all the services, machines, codes that are accessible to him, even on systems that seem highly protected.

Our first research axis therefore aims at highlighting both the effective attacker's means and the way an attack unfolds and spreads.

This knowledge is valuable for security experts who must react quickly during an attack. They need effective ways to understand how their systems may have been compromised.

The main scientific challenge is to be able to adapt to all the attacker's protections against automatic analysis that the attacker could imagine.

In this context, we are particularly interested in

- **highlighting attacks** on hardware that affect software security
- **providing expert support**
  - to analyze malicious code
  - to quickly investigate an intrusion on a system monitored by an intrusion detection system

3.3. Attack Detection

An attack has several phases. A first major phase is the approach phase, during which the attacker enters the system, locates the target and makes himself persistent, the attack is at this point a simple intrusion. In a second phase, the attack is actually launched.

The main objective of intrusion detection is to be able to detect the attacker during the first approach phase. For that purpose, an intrusion detection system (IDS) is based on probes that continuously monitor the system. These probes generate low level alerts (warnings) for any observation of an event that could be a sign of an intrusion. These low-level alerts are very numerous and their semantic value is low. In other words, an IDS generates a huge amount of low-level alerts that bring only few information and overwhelm the security analyst. In addition, many of these alerts are actually false positives, *i.e.* alerts raised when there is no real intrusion.

However, these low-level alerts can themselves be considered as security events by a higher-level IDS: an alert correlation system. These higher-level IDS seek to exploit known relationships between low-level alerts to generate meta-alerts with greater semantic value, *i.e.* with higher-level meaning. An alert correlation system allows to reduce the number of alerts (and especially, false positives) and to return to the security analysts a higher level analysis of the situation.

There are mainly two approaches to detect intrusions. The misuse-based detection and the anomaly-based detection. A misuse-based detection is actually a signature-based detection approach: it allows to detect only the attacks whose signature is available. From our point of view, while useful in practice, misuse-detection is intrinsically limited. Indeed, it requires to continuously update the database of signatures. We follow the alternative approach, namely the anomaly approach, which consists in detecting any deviation from a reference behavior. The main difficulty is thus to compute a model of this reference behavior. Such a model is only useful if it is sufficiently accurate. Otherwise, if the model is an over-approximation, it will be a source of false negatives, *i.e.* real intrusions not detected. If the model is a under-approximation, it will be a source of false positives, *i.e.* normal behaviors seen as intrusions.

In this context, our contributions in intrusion detection systems follow two separate axes: anomaly-based IDS and alert correlation systems. Our contribution in anomaly-based intrusion detection relies on:

- **Illegal Information Flow Detection:** we have proposed to detect information flows in the monitored system (either a node or a set of trusted nodes) that are allowed by the access control mechanism, but are illegal from the security policy point of view. This approach is particularly appealing to detect intrusions in a standalone node.
- **Anomaly-Based Detection in Distributed Applications:** our goal is to specify the normal behavior based on either a formal specification of the distributed application, or previous executions. This approach is particularly appealing to detect intrusions in industrial control systems since these systems exhibit well-defined behaviors at different levels: network level (network communication patterns, protocol specifications, etc.), control level (continuous and discrete process control laws), or even the state of the local resources (memory or CPU).
- **Online data analytics:** our goal is to estimate on the fly different statistics or metrics on distributed input streams to detect abnormal behavior with respect to a well-defined criterion such as the distance between different streams, their correlation or their entropy.

3.4. Attack Resistance

The first two axes of the team allowed us to measure the concrete technical means of the attacker. We claim that the attacker can always avoid the measures put in place to secure a system. We believe that another way to offer more secure systems is to take into account from the design phase that these systems will operate in the presence of an omnipotent attacker. The last research axis of the CIDRE team is focused on offering systems that are resistant to attackers, *i.e.* they can provide the expected services even in the presence of an attacker.

To achieve this goal, we explore two approaches:

- be able to take into account all possible actions of the attacker
- provide services based on the collaboration of a set of nodes that are not affected by the presence in minority of malicious nodes

We are interested in massively-used systems that are essential building blocks of security or privacy.

GALLINETTE Project-Team

3. Research Program

3.1. Scientific Context

Software quality is a requirement that is becoming more and more prevalent, by now far exceeding the traditional scope of embedded systems. The development of tools to construct software that respects a given specification is a major challenge facing computer science. *Proof assistants* such as Coq [50] provide a formal method whose central innovation is to produce *certified programs* by transforming the very activity of programming. Programming and proving are merged into a single development activity, informed by an elegant but rigid mathematical theory inspired by the correspondence between programming, logic and algebra: the *Curry-Howard correspondence*. For the certification of programs, this approach has shown its efficiency in the development of important pieces of certified software such as the C compiler of the CompCert project [79]. The extracted CompCert compiler is reliable and efficient, running only 15% slower than GCC 4 at optimisation level 2 (`gcc -O2`), a level of optimisation that was considered before to be highly unreliable.

Proof assistants can also be used to *formalise mathematical theories*: they not only provide a means of representing mathematical theories in a form amenable to computer processing, but their internal logic provides a language for reasoning about such theories. In the last decade, proof assistants have been used to verify extremely large and complicated proofs of recent mathematical results, sometimes requiring either intensive computations [61], [65] or intricate combinations of a multitude of mathematical theories [60]. But formalised mathematics is more than just proof checking and proof assistants can help with the organisation mathematical knowledge or even with the discovery of new constructions and proofs.

Unfortunately, the rigidity of the theory behind proof assistants impedes their expressiveness both as programming languages and as logical systems. For instance, a program extracted from Coq only uses a purely functional subset of OCaml, leaving behind important means of expression such as side-effects and objects. Limitations also appears in the formalisation of advanced mathematics: proof assistants do not cope well with classical axioms such as excluded middle and choice which are sometimes used crucially. The fact of the matter is that the development of proof assistants cannot be dissociated from a reflection on the nature of programs and proofs coming from the Curry-Howard correspondence. In the EPC Gallinette, we propose to address several drawbacks of proof assistants by pushing the boundaries of this correspondence.

In the 1970's, the Curry-Howard correspondence was seen as a perfect match between functional programs, intuitionistic logic, and Cartesian closed categories. It received several generalisations over the decades, and now it is more widely understood as a fertile correspondence between computation, logic, and algebra. Nowadays, the view of the Curry-Howard correspondence has evolved from a perfect match to a collection of theories meant to explain similar structures at work in logic and computation, underpinned by mathematical abstractions. By relaxing the requirement of a perfect match between programs and proofs, and instead emphasising the common foundations of both, the insights of the Curry-Howard correspondence may be extended to domains for which the requirements of programming and mathematics may in fact be quite different.

Consider the following two major theories of the past decades, which were until recently thought to be irreconcilable:

- **(Martin-Löf) Type theory:** introduced by Martin-Löf in 1971, this formalism [86] is both a programming language and a logical system. The central ingredient is the use of *dependent types* to allow fine-grained invariants to be expressed in program types. In 1985, Coquand and Huet developed a similar system called the *calculus of constructions*, which served as logical foundation of the first implementation of Coq. This kind of systems is still under active development, especially with the recent advent of homotopy type theory (HoTT) [108] which gives a new point of view on types and the notion of equality in type theory.

- **The theory of effects:** starting in the 1980's, Moggi [91] and Girard [58] put forward monads and co-monads as describing various compositional notions of computation. In this theory, programs can have side-effects (state, exceptions, input-output), logics can be non-intuitionistic (linear, classical), and different computational universes can interact (modal logics). Recently, the safe and automatic management of resources has also seen a coming of age (Rust, Modern C++) confirming the importance of linear logic for various programming concepts. It is now understood that the characteristic feature of the theory of effects is sensitivity to *evaluation order*, in contrast with type theory which is built around the assumption that evaluation order is irrelevant.

We now outline a series of scientific challenges aimed at understanding of type theory, effects, and their combination.

More precisely, three key axes of improvement have been identified:

1. Making the notion of equality closer to what is usually assumed when doing proofs on black board, with a balance between irrelevant equality for simple structures and equality up-to equivalences for more complex ones (Section 3.2). Such a notion of equality should allow one to implement traditional model transformations that enhance the logical power of the proof assistant using distinct compilation phases.
2. Advancing the foundations of effects within the Curry-Howard approach. The objective is to pave the way for the integration of effects in proof assistants and to prototype the corresponding implementation. This integration should allow for not only certified programming with effects, but also the expression of more powerful logics (Section 3.3).
3. Making more programming features (notably, object polymorphism) available in proof assistants, in order to scale to practical-sized developments. The objective is to enable programming styles closer to common practices. One of the key challenges here is to leverage gradual typing to dependent programming (Section 3.4).

To validate the new paradigms, we propose in Section 3.5 three particular application fields in which members of the team already have a strong expertise: code refactoring, constraint programming and symbolic computation.

3.2. Enhance the computational and logical power of proof assistants

The democratisation of proof assistants based on type theory has likely been impeded one central problem: the mismatch between the conception of equality in mathematics and its formalisation in type theory. Indeed, some basic principles that are used implicitly in mathematics—such as Church's principle of propositional extensionality, which says that two propositions are equal when they are logically equivalent—are not derivable in type theory. Even more problematically, from a computer science point of view, the basic concept of two functions being equal when they are equal at every "point" of their domain is also not derivable: rather, it must be added as an additional axiom. Of course, these principles are consistent with type theory so that working under the corresponding additional assumptions is safe. But the use of these assumptions in a definition potentially clutters its computational behaviour: since axioms are computational black boxes, computation gets stuck at the points of the code where they have been used.

We propose to investigate how expressive logical transformations such as forcing [71] and sheaf construction might be used to enhance the computational and logical power of proof assistants—with a particular emphasis on their implementation in the Coq proof assistant by the means of effective translations (or compilation phases). One of the main topics of this task, in connection to the ERC project CoqHoTT, is the integration in Coq of new concepts inspired by homotopy type theory [108] such as the univalence principle, and higher inductive types.

3.2.1. A definitional proof-irrelevant version of Coq.

In the Coq proof assistant, the sort $\mathbf{Prop}$ stands for the universe of types which are propositions. That is, when a term P has type $\mathbf{Prop}$, the only relevant fact is whether P is inhabited (that is true) or not (that is false). This property, known as *proof irrelevance,* can be expressed formally as: $\forall x\ y : P, x = y$. Originally, the *raison d'être* of the sort $\mathbf{Prop}$ was to characterise types with no computational meaning with the intention that terms of such types could be erased upon extraction. However, the assumption that every element of $\mathbf{Prop}$ should be proof irrelevant has never been integrated to the system. Indeed, in Coq, proof irrelevance for the sort $\mathbf{Prop}$ is not incorporated into the theory: it is only compatible with it, in the sense that its assumption does not give rise to an inconsistent theory. In fact, the exact status of the sort $\mathbf{Prop}$ in Coq has never been entirely clarified, which explains in part this lack of integration. Homotopy type theory brings fresh thinking on this issue and suggests turning $\mathbf{Prop}$ into the collection of terms that a certain static inference procedure tags as proof irrelevant. The goal of this task is to integrate this insight in the Coq system and to implement a definitional proof-irrelevant version of the sort $\mathbf{Prop}$.

3.2.2. Extend the Coq proof assistant with a computational version of univalence

The univalence principle is becoming widely accepted as a very promising avenue to provide new foundations for mathematics and type theory. However, this principle has not yet been incorporated into a proof assistant. Indeed, the very mathematical structures (known as ∞-groupoids) motivating the theory remain to this day an active area of research. Moreover, a correct and decidable type checking procedure for the whole theory raises both computational complexity and logical coherence issues. Observational type theory [33], as implemented in Epigram, provides a first-stage approximation to homotopy type theory, but only deals with functional extensionality and does not capture univalence. Coquand and his collaborators have obtained significant results on the computational meaning of univalence using cubical sets [40], [46]. Bickford has initiated a promising formalisation work [0] in the NuPRL system. However, a complete formalisation in intensional type theory remains an open problem.

Hence a major objective is to achieve a complete internalisation of univalence in intensional type theory, including an integration to a new version of Coq. We will strive to keep compatibility with previous versions, in particular from a performance point of view. Indeed, the additional complexity of homotopy type theory should not induce an overhead in the type checking procedure used by the software if we want our new framework to become rapidly adopted by the community. Concretely, we will make sure that the compilation time of Coq's Standard Library will be of the same order of magnitude.

3.2.3. Extend the logical power of type theory without axioms in a modular way

Extending the power of a logic using model transformations (*e.g.,* forcing transformation [72], [71] or the sheaf construction [101]) is a classic topic of mathematical logic [47], [77]. However, these ideas have not been much investigated in the setting of type theory, even though they may provide a useful framework for extending the logical power of proof assistant in a modular way. There is a good reason for this: with a syntactic notion of equality, the underlying structure of type theory does not conform to the structure of topos used in mathematical logic. A direct incorporation of the standard techniques is therefore not possible. However, a univalent notion of equality brings type theory closer to the required algebraic structure, as it corresponds to the notion of ∞-topos recently studied by Lurie [84]. The goal of this task is to revisit model transformations in the light of the univalence principle, and to obtain in this way new internal transformations in type theory which can in turn be seen as compilation phases. The general notion of an internal syntactical translation has already been investigated in the team [41].

3.2.4. Methodology: Extending type theory with different compilation phases

The Gallinette project advocates the use of distinct compilation phases as a methodology for the design of a new generation of proof assistants featuring modular extensions of a core logic. The essence of a compiler is the separation of the complexity of a translation process into modular stages, and the organization of their

[0] http://www.nuprl.org/wip/Mathematics/cubical!type!theory/index.html

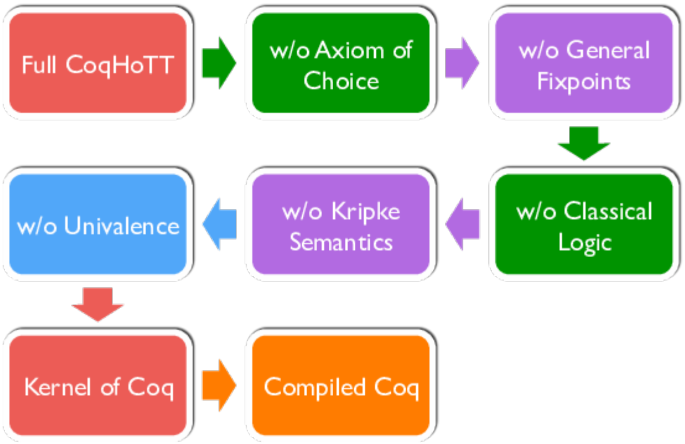

Figure 1. Multiple compilation phases to increase the logical and computational power of Coq.

re-composition. This idea finds a natural application in the design of complex proof assistants (Figure 1). For instance, the definition of type classes in Coq follows this pattern, and is morally given by the means of a translation into a type-class free kernel. More recently, a similar approach by compilation stages, using the forcing transformation, was used to relax the strict positivity condition guarding inductive types [72], [71]. We believe that this flavour of compilation-based strategies offers a promising direction of investigation for the propose of defining a decidable type checking algorithm for HoTT.

3.3. Semantic and logical foundations for effects in proof assistants based on type theory

We propose the incorporation of effects in the theory of proof assistants at a foundational level. Not only would this allow for certified programming with effects, but it would moreover have implications for both semantics and logic.

We mean *effects* in a broad sense that encompasses both Moggi's monads [91] and Girard's linear logic [58]. These two seminal works have given rise to respective theories of effects (monads) and resources (co-monads). Recent advances, however have unified these two lines of thought: it is now clear that the defining feature of effects, in the broad sense, is sensitivity to evaluation order [80], [51].

In contrast, the type theory that forms the foundations of proof assistants is based on pure λ calculus and is built on the assumption that evaluation order is irrelevant. Evaluation order is therefore the blind spot of type theory. In Moggi [92], integrating the dependent types of type theory with monads is *"the next difficult step [...] currently under investigation"*.

Any realistic program contains effects: state, exceptions, input-output. More generally, evaluation order may simply be important for complexity reasons. With this in mind, many works have focused on certified programming with effects: notably Ynot [96], and more recently $\mathrm{F}^{\star}$ [106] and Idris [42], which propose various ways for encapsulating effects and restricting the dependency of types on effectful terms. Effects are either specialised, such as the monads with Hoare-style pre- and post-conditions found in Ynot or $\mathrm{F}^{\star}$, or more general, such as the algebraic effects implemented in Idris. But whereas there are several experiments and projects pursuing the certification of programs with effects, each making its own choices on how effects and dependency should be merged, there is on the other hand a deficit of logical and semantic investigations.

We propose to develop the foundations of a type theory with effects taking into account the logical and semantic aspects, and to study their practical and theoretical consequences. A type theory that integrates effects would have logical, algebraic and computational implications when viewed through the Curry-Howard correspondence. For instance, effects such as control operators establish a link with classical proof theory [63]. Indeed, control operators provide computational interpretations of type isomorphisms such as $A \cong \neg\neg A$ and $\neg\forall x A \cong \exists x \neg A$ (e.g. [93]), whereas the conventional wisdom of type theory holds that such axioms are non-constructive (this is for instance the point of view that has been advocated so far in homotopy type theory [108]). Another example of an effect with logical content is state (more precisely memoization) which is used to provide constructive content to the classical dependent axiom of choice [39], [75], [67]. In the long term, a whole body of literature on the constructive content of classical proofs is to be explored and integrated, providing rich sources of inspiration: Kohlenbach's proof mining [74] and Simpson's reverse mathematics [104], for instance, are certainly interesting to investigate from the Curry-Howard perspective.

The goal is to develop a type theory with effects that accounts both for practical experiments in certified programming, and for clues from denotational semantics and logical phenomena, in a unified setting.

3.3.1. Models for integrating effects with dependent types

A crucial step is the integration of dependent types with effects, a topic which has remained *"currently under investigation"* [92] ever since the beginning. The difficulty resides in expressing the dependency of types on terms that can perform side-effects during the computation. On the side of denotational semantics, several extensions of categorical models for effects with dependent types have been proposed [30], [109] using axioms that should correspond to restrictions in terms of expressivity but whose practical implications, however, are not immediately transparent. On the side of logical approaches [67], [68], [78], [90], one first considers a drastic restriction to terms that do not compute, which is then relaxed by semantic means. On the side of systems for certified programming such as F$^{☆}$, the type system ensures that types only depend on pure and terminating terms.

Thus, the recurring idea is to introduce restrictions on the dependency in order to establish an encapsulation of effects. In our approach, we seek a principled description of this idea by developing the concept of *semantic value* (thunkables, linears) which arose from foundational considerations [57], [103], [94] and whose relevance was highlighted in recent works [81], [100]. The novel aspect of our approach is to seek a proper extension of type theory which would provide foundations for a classical type theory with axiom of choice in the style of Herbelin [67], but which moreover could be generalised to effects other than just control by exploiting an abstract and adaptable notion of semantic value.

3.3.2. Intuitionistic depolarisation

In our view, the common idea that evaluation order does not matter for pure and termination computations should serve as a bridge between our proposals for dependent types in the presence of effects and traditional type theory. Building on the previous goal, we aim to study the relationship between semantic values, purity, and parametricity theorems [102], [59]. Our goal is to characterise parametricity as a form of intuitionistic *depolarisation* following the method by which the first game model of full linear logic was given (Melliès [87], [88]). We have two expected outcomes in mind: enriching type theory with intensional content without losing its properties, and giving an explanation of the dependent types in the style of Idris and F$^{☆}$ where purity- and termination-checking play a role.

3.3.3. Developing the rewriting theory of calculi with effects

An integrated type theory with effects requires an understanding of evaluation order from the point of view of rewriting. For instance, rewriting properties can entail the decidability of some conversions, allowing the automation of equational reasoning in types [28]. They can also provide proofs of computational consistency (that terms are not all equivalent) by showing that extending calculi with new constructs is conservative [105]. In our approach, the λ-calculus is replaced by a calculus modelling the evaluation in an abstract machine [52]. We have shown how this approach generalises the previous semantic and proof-theoretic approaches [34], [80], [82], and overcomes their shortcomings [95].

One goal is to prove computational consistency or decidability of conversions purely using advanced rewriting techniques following a technique introduced in [105]. Another goal is the characterisation of weak reductions: extensions of the operational semantics to terms with free variables that preserve termination, whose iteration is equivalent to strong reduction [29], [55]. We aim to show that such properties derive from generic theorems of higher-order rewriting [111], so that weak reduction can easily be generalised to richer systems with effects.

3.3.4. Direct models and categorical coherence

Proof theory and rewriting are a source of *coherence theorems* in category theory, which show how calculations in a category can be simplified with an embedding into a structure with stronger properties [85], [76]. We aim to explore such results for categorical models of effects [80], [51]. Our key insight is to consider the reflection between *indirect and direct models* [57], [94] as a coherence theorem: it allows us to embed the traditional models of effects into structures for which the rewriting and proof-theoretic techniques from the previous section are effective.

Building on this, we are further interested in connecting operational semantics to 2-category theory, in which a second dimension is traditionally considered for modelling conversions of programs rather than equivalences. This idea has been successfully applied for the λ-calculus [73], [69] but does not scale yet to more realistic models of computation. In our approach, it has already been noticed that the expected symmetries coming from categorical dualities are better represented, motivating a new investigation into this long-standing question.

3.3.5. Models of effects and resources

The unified theory of effects and resources [51] prompts an investigation into the semantics of safe and automatic resource management, in the style of Modern C++ and Rust. Our goal is to show how advanced semantics of effects, resources, and their combination arise by assembling elementary blocks, pursuing the methodology applied by Melliès and Tabareau in the context of continuations [89]. For instance, by combining control flow (exceptions, return) with linearity allows us to describe in a precise way the “Resource Acquisition Is Initialisation” idiom in which the resource safety is ensured with scope-based destructors. A further step would be to reconstruct uniqueness types and borrowing using similar ideas.

3.4. Language extensions for the scaling of proof assistants

The development of tools to construct software systems that respect a given specification is a major challenge of current and future research in computer science. Certified programming with dependent types has recently attracted a lot of interest, and Coq is the *de facto* standard for such endeavours, with an increasing number of users, pedagogical resources, and large-scale projects. Nevertheless, significant work remains to be done to make Coq more usable from a software engineering point of view. The Gallinette team proposes to make progress on three lines of work: (i) the development of gradual certified programming, (ii) the integration of imperative features and object polymorphism in Coq, and (iii) the development of robust tactics for proof engineering for the scaling of formalised libraries.

3.4.1. Gradual Certified Programming

One of the main issues faced by a programmer starting to internalise in a proof assistant code written in a more permissive world is that type theory is constrained by a strict type discipline which lacks flexibility. Concretely, as soon that you start giving more a precise type/specification to a function, the rest of the code interacting with this functions needs to be more precise too. To address this issue, the Gallinette team will put strong efforts into the development of gradual typing in type theory to allow progressive integration of code that comes from a more permissive world.

Indeed, on the way to full verification, programmers can take advantage of a gradual approach in which some properties are simply asserted instead of proven, subject to dynamic verification. Tabareau and Tanter have made preliminary progress in this direction [107]. This work, however, suffers from a number of limitations, the most important being the lack of a mechanism for handling the possibility of runtime errors within Coq. Instead of relying on axioms, this project will explore the application of Section 3.3 to embed effects in Coq.

This way, instead of postulating axioms for parts of the development that are too hard/marginal to be dealt with, the system adds dynamic checks. Then, after extraction, we get a program that corresponds to the initial program but with dynamic check for parts that have not been proven, ensuring that the program will raise an error instead of going outside its specification.

This will yield new foundations of gradual certified programming, both more expressive and practical. We will also study how to integrate previous techniques with the extraction mechanism of Coq programs to OCaml, in order to exploit the exception mechanism of OCaml.

3.4.2. Imperative features and object polymorphism in the Coq proof assistant

3.4.2.1. Imperative features.

Abstract data types (ADTs) become useful as the size of programs grows since they provide for a modular approach, allowing abstractions about data to be expressed and then instantiated. Moreover, ADTs are natural concepts in the calculus of inductive constructions. But while it is easy to declare an ADT, it is often difficult to implement an efficient one. Compare this situation with, for example, Okasaki's purely functional data structures [97] which implement ADTs like queues in languages with imperative features. Of course, Okasaki's queues enforce some additional properties for free, such as persistence, but the programmer may prefer to use and to study a simpler implementation without those additional properties. Also in certified symbolic computation (see 3.5.3), an efficient functional implementation of ADTs is often not available, and efficiency is a major challenge in this area. Relying on the theoretical work done in 3.3 , we will equip Coq with imperative features and we will demonstrate how they can be used to provide efficient implementations of ADTs. However, it is also often the case that imperative implementation are hard-to-reason-on, requiring for instance the use of separation logic. But in that case, we could take benefice of recent works on integration of separation logic in the Coq proof assistant and in particular the Iris project http://iris-project.org/.

3.4.2.2. Object polymorphism.

Object-oriented programming has evolved since its foundation based on the representation of computations as an exchange of messages between objects. In modern programming languages like Scala, which aims at a synthesis between object-oriented and functional programming, object-orientation concretely results in the use of hierarchies of interfaces ordered by the subtyping relation and the definition of interface implementations that can interoperate. As observed by Cook and Aldrich [49], [32], interoperability can be considered as the essential feature of objects and is a requirement for many modern frameworks and ecosystems: it means that two different implementations of the same interface can interoperate.

Our objective is to provide a representation of object-oriented programs, by focusing on subtyping and interoperability.

For subtyping, the natural solution in type theory is coercive subtyping [83], as implemented in Coq, with an explicit operator for coercions. This should lead to a shallow embedding, but has limitations: indeed, while it allows subtyping to be faithfully represented, it does not provide a direct means to represent union and intersection types, which are often associated with subtyping (for instance intersection types are present in Scala). A more ambitious solution would be to resort to subsumptive subtyping (or semantic subtyping [56]): in its more general form, a type algebra is extended with boolean operations (union, intersection, complementing) to get a boolean algebra with operators (the original type constructors). Subtying is then interpreted as the natural partial order of the boolean algebra.

We propose to use the type class machinery of Coq to implement semantic subtyping for dependent type theory. Using type class resolution, we can emulate inference rules of subsumptive subtyping without modifying Coq internally. This has also another advantage. As subsumptive subtyping for dependent types should be undecidable in general, using type class resolution allows for an incomplete yet extensible decision procedure.

3.4.3. Robust tactics for proof engineering for the scaling of formalised libraries

When developing certified software, a major part of the effort is spent not only on writing proof scripts, but on *rewriting* them, either for the purpose of code maintenance or because of more significant changes in the base definitions. Regrettably, proof scripts suffer more often than not from a bad programming style, and too many proof developers casually neglect the most elementary principles of well-behaved programmers. As a result, many proof scripts are very brittle, user-defined tactics are often difficult to extend, and sometimes even lack a clear specification. Formal libraries are thus generally very fragile pieces of software. One reason for this unfortunate situation is that proof engineering is very badly served by the tools currently available to the users of the Coq proof assistant, starting with its tactic language. One objective of the Gallinette team is to develop better tools to write proof scripts.

Completing and maintaining a large corpus of formalised mathematics requires a well-designed tactic language. This language should both accommodate the possible specific needs of the theories at stake, and help with diagnostics at refactoring time. Coq's tactic language is in fact two-leveled. First, it includes a basic tactic language, to organise the deductive steps in a proof script and to perform the elementary bureaucracy. Its second layer is a meta-programming language, which allows user to defined their own new tactics at toplevel. Our first direction of work consists in the investigation of the appropriate features of the *basic tactic language*. For instance, the design of the Ssreflect tactic language, and its support for the small scale reflection methodology [62], has been a key ingredient in at least two large scale formalisation endeavours: the Four Colour Theorem [61] and of the Odd Order Theorem [60]. Building on our experience with the Ssreflect tactic language, we will contribute to the ongoing work on the basic tactic language for Coq. The second objective of this task is to contribute to the design of a *typed tactic language*. In particular, we will build on the work of Ziliani and his collaborators [110], extending it with reasoning about the effects that tactics have on the "state of a proof" (e.g. number of sub-goals, metavariables in context). We will also develop a novel approach for incremental type checking of proof scripts, so that programmers gain access to a richer discovery- engineering interaction with the proof assistant.

3.5. Practical experiments

The first three axes of the EPC Gallinette aim at developing a new generation of proof assistants. But we strongly believe that foundational investigations must go hand in hand with practical experiments. Therefore, we expect to benefit from existing expertise and collaborations in the team to experiment our extensions of Coq on real world developments. It should be noticed that those practical experiments are strongly guided by the deep history of research on software engineering of team members.

3.5.1. Certified Code Refactoring

In the context of refactoring of C programs, we intend to formalise program transformations that are written in an imperative style to test the usability of our addition of effects in the proof assistant. This subject has been chosen based on the competence of members of the team.

We are currently working on the formalisation of refactoring tools in Coq [45]. Automatic refactoring of programs in industrial languages is difficult because of the large number of potential interactions between language features that are difficult to predict and to test. Indeed, all available refactoring tools suffer from bugs : they fail to ensure that the generated program has the same behaviour as the input program. To cope with that difficulty, we have chosen to build a refactoring tool with Coq : a program transformation is written in the Coq programming language, then proven correct on all possible inputs, and then an OCaml executable program is generated by the platform. We rely on the CompCert C formalisation of the C language. CompCert is currently the most complete formalisation of an industrial language, which justifies that choice. We have three goals in that project :

- Build a refactoring tool that programmers can rely on and make it available in a popular platform (such as Eclipse, IntelliJ or Frama-C).
- Explore large, drastic program transformations such as replacing a design architecture for an other one, by applying a sequence of small refactoring operations (as we have done for Java and Haskell programs before [48], [44], [31]), while ensuring behaviour preservation.
- Explore the use of enhancements of proof systems on large developments. For instance, refactoring tools are usually developed in the imperative/object paradigm, so the extension of Coq with side effects or with object features proposed in the team can find a direct use-case here.

3.5.2. Certified Constraint Programming

We plan to make use of the internalisation of the object-oriented paradigm in the context of constraint programming. Indeed, this domain is made of very complex algorithms that are often developed using object-oriented programming (as it is the case for instance for CHOCO, which is developed in the Tasc Group at IMT Atlantique, Nantes). We will in particular focus on filtering algorithms in constraint solvers, for which research publications currently propose new algorithms with manual proofs. Their formalisation in Coq is challenging. Another interesting part of constraint solving to formalise is the part that deals with program generation (as opposed to extraction). However, when there are numerous generated pieces of code, it is not realistic to prove their correctness manually, and it can be too difficult to prove the correctness of a generator. So we intend to explore a middle path that consists in generating a piece of code along with its corresponding proof (script or proof term). A target application could be interval constraints (for instance Allen interval algebra or region connection calculus) that can generate thousands of specialised filtering algorithms for a small number of variables [37].

Finally, Rémi Douence has already worked (articles publishing [64], [98], [54], PhD Thesis advising [99]) with different members of the Tasc team. Currently, he supervises with Nicolas Beldiceanu the PhD Thesis of Ekaterina Arafailova in the Tasc team. She studies finite transducers to model time-series constraints [38], [36], [35]. This work requires proofs, manually done for now, we would like to explore when these proofs could be mechanised.

3.5.3. Certified Symbolic Computation

We will investigate how the addition of effects in the Coq proof assistant can facilitate the marriage of computer algebra with formal proofs. Computer algebra systems on one hand, and proof assistants on the other hand, are both designed for doing mathematics with the help of a computer, by the means of symbolic computations. These two families of systems are however very different in nature: computer algebra systems allow for implementations faithful to the theoretical complexity of the algorithms, whereas proof assistants have the expressiveness to specify exactly the semantic of the data-structures and computations.

Experiments have been run that link computer algebra systems with Coq [53], [43]. These bridges rely on the implementation of formal proof-producing core algorithms like normalisation procedures. Incidentally, they require non trivial maintenance work to survive the evolution of both systems. Other proof assistants like the Isabelle/HOL system make use of so-called reflection schemes: the proof assistant can produce code in an external programming language like SML, but also allows to import the values output by these extracted programs back inside the formal proofs. This feature extends the trusted base of code quite significantly but it has been used for major achievements like a certified symbolic/numeric ODE solver [70].

We would like to bring Coq closer to the efficiency and user-friendliness of computer algebra systems: for now it is difficult to use the Coq programming language so that certified implementations of computer algebra algorithms have the right, observable, complexity when they are executed inside Coq. We see the addition of effects to the proof assistant as an opportunity to ease these implementations, for instance by making use of caching mechanisms or of profiling facilities. Such enhancements should enable the verification of computation-intensive mathematical proofs that are currently beyond reach, like the validation of Helfgott's proof of the weak Goldbach conjecture [66].

HYCOMES Project-Team

3. Research Program

3.1. Hybrid Systems Modeling

Systems industries today make extensive use of mathematical modeling tools to design computer controlled physical systems. This class of tools addresses the modeling of physical systems with models that are simpler than usual scientific computing problems by using only Ordinary Differential Equations (ODE) and Difference Equations but not Partial Differential Equations (PDE). This family of tools first emerged in the 1980's with SystemBuild by MatrixX (now distributed by National Instruments) followed soon by Simulink by Mathworks, with an impressive subsequent development.

In the early 90's control scientists from the University of Lund (Sweden) realized that the above approach did not support component based modeling of physical systems with reuse [0]. For instance, it was not easy to draw an electrical or hydraulic circuit by assembling component models of the various devices. The development of the Omola language by Hilding Elmqvist was a first attempt to bridge this gap by supporting some form of Differential Algebraic Equations (DAE) in the models. Modelica quickly emerged from this first attempt and became in the 2000's a major international concerted effort with the Modelica Consortium [0]. A wider set of tools, both industrial and academic, now exists in this segment [0]. In the EDA sector, VHDL-AMS was developed as a standard [11] and also allows for differential algebraic equations. Several domain-specific languages and tools for mechanical systems or electronic circuits also support some restricted classes of differential algebraic equations. Spice is the historic and most striking instance of these domain-specific languages/tools [0]. The main difference is that equations are hidden and the fixed structure of the differential algebraic results from the physical domain covered by these languages.

Despite these tools are now widely used by a number of engineers, they raise a number of technical difficulties. The meaning of some programs, their mathematical semantics, can be tainted with uncertainty. A main source of difficulty lies in the failure to properly handle the discrete and the continuous parts of systems, and their interaction. How the propagation of mode changes and resets should be handled? How to avoid artifacts due to the use of a global ODE solver causing unwanted coupling between seemingly non interacting subsystems? Also, the mixed use of an equational style for the continuous dynamics with an imperative style for the mode changes and resets is a source of difficulty when handling parallel composition. It is therefore not uncommon that tools return complex warnings for programs with many different suggested hints for fixing them. Yet, these "pathological" programs can still be executed, if wanted so, giving surprising results — See for instance the Simulink examples in [19], [5] and [14].

Indeed this area suffers from the same difficulties that led to the development of the theory of synchronous languages as an effort to fix obscure compilation schemes for discrete time equation based languages in the 1980's. Our vision is that hybrid systems modeling tools deserve similar efforts in theory as synchronous languages did for the programming of embedded systems.

3.2. Background on non-standard analysis

Non-Standard analysis plays a central role in our research on hybrid systems modeling [5], [19], [15], [14]. The following text provides a brief summary of this theory and gives some hints on its usefulness in the context of hybrid systems modeling. This presentation is based on our paper [1], a chapter of Simon Bliudze's PhD thesis [24], and a recent presentation of non-standard analysis, not axiomatic in style, due to the mathematician Lindström [48].

[0] http://www.lccc.lth.se/media/LCCC2012/WorkshopSeptember/slides/Astrom.pdf
[0] https://www.modelica.org/
[0] SimScape by Mathworks, Amesim by LMS International, now Siemens PLM, and more.
[0] http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/MANUALS/spice3.html

Non-standard numbers allowed us to reconsider the semantics of hybrid systems and propose a radical alternative to the *super-dense time semantics* developed by Edward Lee and his team as part of the Ptolemy II project, where cascades of successive instants can occur in zero time by using $\mathbb{R}_+ \times \mathbb{N}$ as a time index. In the non-standard semantics, the time index is defined as a set $\mathbb{T} = \{n\partial \mid n \in {}^*\mathbb{N}\}$, where ∂ is an *infinitesimal* and ${}^*\mathbb{N}$ is the set of *non-standard integers*. Remark that (1) $\mathbb{T}$ is dense in $\mathbb{R}_+$, making it "continuous", and (2) every $t \in \mathbb{T}$ has a predecessor in $\mathbb{T}$ and a successor in $\mathbb{T}$, making it "discrete". Although it is not effective from a computability point of view, the *non-standard semantics* provides a framework that is familiar to the computer scientist and at the same time efficient as a symbolic abstraction. This makes it an excellent candidate for the development of provably correct compilation schemes and type systems for hybrid systems modeling languages.

Non-standard analysis was proposed by Abraham Robinson in the 1960s to allow the explicit manipulation of "infinitesimals" in analysis [54], [41], [10]. Robinson's approach is axiomatic; he proposes adding three new axioms to the basic Zermelo-Fraenkel (ZFC) framework. There has been much debate in the mathematical community as to whether it is worth considering non-standard analysis instead of staying with the traditional one. We do not enter this debate. The important thing for us is that non-standard analysis allows the use of the non-standard discretization of continuous dynamics "as if" it was operational.

Not surprisingly, such an idea is quite ancient. Iwasaki et al. [43] first proposed using non-standard analysis to discuss the nature of time in hybrid systems. Bliudze and Krob [25], [24] have also used non-standard analysis as a mathematical support for defining a system theory for hybrid systems. They discuss in detail the notion of "system" and investigate computability issues. The formalization they propose closely follows that of Turing machines, with a memory tape and a control mechanism.

3.3. Contract-Based Design, Interfaces Theories, and Requirements Engineering

System companies such as automotive and aeronautic companies are facing significant difficulties due to the exponentially raising complexity of their products coupled with increasingly tight demands on functionality, correctness, and time-to-market. The cost of being late to market or of imperfections in the products is staggering as witnessed by the recent recalls and delivery delays that many major car and airplane manufacturers had to bear in the recent years. The specific root causes of these design problems are complex and relate to a number of issues ranging from design processes and relationships with different departments of the same company and with suppliers, to incomplete requirement specification and testing.

We believe the most promising means to address the challenges in systems engineering is to employ structured and formal design methodologies that seamlessly and coherently combine the various viewpoints of the design space (behavior, space, time, energy, reliability, ...), that provide the appropriate abstractions to manage the inherent complexity, and that can provide correct-by-construction implementations. The following technology issues must be addressed when developing new approaches to the design of complex systems:

- The overall design flows for heterogeneous systems and the associated use of models across traditional boundaries are not well developed and understood. Relationships between different teams inside a same company, or between different stake-holders in the supplier chain, are not well supported by solid technical descriptions for the mutual obligations.
- System requirements capture and analysis is in large part a heuristic process, where the informal text and natural language-based techniques in use today are facing significant challenges. Formal requirements engineering is in its infancy: mathematical models, formal analysis techniques and links to system implementation must be developed.
- Dealing with variability, uncertainty, and life-cycle issues, such as extensibility of a product family, are not well-addressed using available systems engineering methodologies and tools.

The challenge is to address the entire process and not to consider only local solutions of methodology, tools, and models that ease part of the design.

Contract-based design has been proposed as a new approach to the system design problem that is rigorous and effective in dealing with the problems and challenges described before, and that, at the same time, does not require a radical change in the way industrial designers carry out their task as it cuts across design flows of different type. Indeed, contracts can be used almost everywhere and at nearly all stages of system design, from early requirements capture, to embedded computing infrastructure and detailed design involving circuits and other hardware. Contracts explicitly handle pairs of properties, respectively representing the assumptions on the environment and the guarantees of the system under these assumptions. Intuitively, a contract is a pair $C = (A, G)$ of assumptions and guarantees characterizing in a formal way 1) under which context the design is assumed to operate, and 2) what its obligations are. Assume/Guarantee reasoning has been known for a long time, and has been used mostly as verification mean for the design of software [52]. However, contract based design with explicit assumptions is a philosophy that should be followed all along the design, with all kinds of models, whenever necessary. Here, specifications are not limited to profiles, types, or taxonomy of data, but also describe the functions, performances of various kinds (time and energy), and reliability. This amounts to enrich a component's interface with, on one hand, formal specifications of the behavior of the environment in which the component may be instantiated and, on the other hand, of the expected behavior of the component itself. The consideration of rich interfaces is still in its infancy. So far, academic researchers have addressed the mathematics and algorithmics of interfaces theories and contract-based reasoning. To make them a technique of choice for system engineers, we must develop:

- Mathematical foundations for interfaces and requirements engineering that enable the design of frameworks and tools;
- A system engineering framework and associated methodologies and tool sets that focus on system requirements modeling, contract specification, and verification at multiple abstraction layers.

A detailed bibliography on contract and interface theories for embedded system design can be found in [6]. In a nutshell, contract and interface theories fall into two main categories:

Assume/guarantee contracts. By explicitly relying on the notions of assumptions and guarantees, A/G-contracts are intuitive, which makes them appealing for the engineer. In A/G-contracts, assumptions and guarantees are just properties regarding the behavior of a component and of its environment. The typical case is when these properties are formal languages or sets of traces, which includes the class of safety properties [45], [32], [51], [13], [34]. Contract theories were initially developed as specification formalisms able to refuse some inputs from the environment [42]. A/G-contracts were advocated by the SPEEDS project [18]. They were further experimented in the framework of the CESAR project [37], with the additional consideration of *weak* and *strong* assumptions. This is still a very active research topic, with several recent contributions dealing with the timed [23] and probabilistic [28], [29] viewpoints in system design, and even mixed-analog circuit design [53].

Automata theoretic interfaces. Interfaces combine assumptions and guarantees in a single, automata theoretic specification. Most interface theories are based on Lynch Input/Output Automata [50], [49]. Interface Automata [57], [56], [58], [30] focus primarily on parallel composition and compatibility: Two interfaces can be composed and are compatible if there is at least one environment where they can work together. The idea is that the resulting composition exposes as an interface the needed information to ensure that incompatible pairs of states cannot be reached. This can be achieved by using the possibility, for an Interface Automaton, to refuse selected inputs from the environment in a given state, which amounts to the implicit assumption that the environment will never produce any of the refused inputs, when the interface is in this state. Modal Interfaces [3] inherit from both Interface Automata and the originally unrelated notion of Modal Transition System [47], [12], [26], [46]. Modal Interfaces are strictly more expressive than Interface Automata by decoupling the I/O orientation of an event and its deontic modalities (mandatory, allowed or forbidden). Informally, a *must* transition is available in every component that realizes the modal interface, while a *may* transition needs not be. Research on interface theories is still very active. For instance, timed [59], [20], [22], [39], [38], [21], probabilistic [28], [40] and energy-aware [31] interface theories have been proposed recently.

Requirements Engineering is one of the major concerns in large systems industries today, particularly so in sectors where certification prevails [55]. DOORS projects collecting requirements are poorly structured and cannot be considered a formal modeling framework today. They are nothing more than an informal documentation enriched with hyperlinks. As examples, medium size sub-systems may have a few thousands requirements and the Rafale fighter aircraft has above 250,000 of them. For the Boeing 787, requirements were not stable while subcontractors performed the development of the fly-by-wire and of the landing gear subsystems.

We see Contract-Based Design and Interfaces Theories as innovative tools in support of Requirements Engineering. The Software Engineering community has extensively covered several aspects of Requirements Engineering, in particular:

- the development and use of large and rich *ontologies*; and
- the use of Model Driven Engineering technology for the structural aspects of requirements and resulting hyperlinks (to tests, documentation, PLM, architecture, and so on).

Behavioral models and properties, however, are not properly encompassed by the above approaches. This is the cause of a remaining gap between this phase of systems design and later phases where formal model based methods involving behavior have become prevalent—see the success of Matlab/Simulink/Scade technologies. We believe that our work on contract based design and interface theories is best suited to bridge this gap.

PACAP Project-Team

3. Research Program

3.1. Motivation

Our research program is naturally driven by the evolution of our ecosystem. Relevant recent changes can be classified in the following categories: technological constraints, evolving community, and domain constraints. We hereby summarize these evolutions.

3.1.1. Technological constraints

Until recently, binary compatibility guaranteed portability of programs, while increased clock frequency and improved micro-architecture provided increased performance. However, in the last decade, advances in technology and micro-architecture started translating into more parallelism instead. Technology roadmaps even predict the feasibility of thousands of cores on a chip by 2020. Hundreds are already commercially available. Since the vast majority of applications are still sequential, or contain significant sequential sections, such a trend put an end to the automatic performance improvement enjoyed by developers and users. Many research groups consequently focused on parallel architectures and compiling for parallelism.

Still, the performance of applications will ultimately be driven by the performance of the sequential part. Despite a number of advances (some of them contributed by members of the team), sequential tasks are still a major performance bottleneck. Addressing it is still on the agenda of the proposed PACAP project-team.

In addition, due to power constraints, only part of the billions of transistors of a microprocessor can be operated at any given time (the *dark silicon* paradigm). A sensible approach consists in specializing parts of the silicon area to provide dedicated accelerators (not run simultaneously). This results in diverse and heterogeneous processor cores. Application and compiler designers are thus confronted with a moving target, challenging portability and jeopardizing performance.

Finally, we live in a world where billions of sensors, actuators, and computers play a crucial role in our life: flight control, nuclear plant management, defense systems, banking, or health care. These systems must be reliable, despite the fact that they are subject to faults (for example due to aging, charged particle hit, or random noise). Faults will soon become the new *de facto* standard. The evolution of the semiconductor industry predicts an exponential growth of the number of permanent faults [45]. Reliability considerations usually degrade performance. We will propose solutions to mitigate this impact (for example by limiting overheads to critical sections).

Note on technology.
Technology also progresses at a fast pace. We do not propose to pursue any research on technology *per se*. Recently proposed paradigms (non-Silicon, brain-inspired) have received lots of attention from the research community. We do *not* intend to invest in those paradigms, but we will continue to investigate compilation and architecture for more conventional programming paradigms. Still, several technological shifts may have consequences for us, and we will closely monitor their developments. They include for example non-volatile memory (impacts security, makes writes longer than loads), 3D-stacking (impacts bandwidth), and photonics (impacts latencies and connection network).

3.1.2. Evolving community

The PACAP project-team tackles performance-related issues, for conventional programming paradigms. In fact, programming complex environments is no longer the exclusive domain of experts in compilation and architecture. A large community now develops applications for a wide range of targets, including mobile “apps”, cloud, multicore or heterogeneous processors.

This also includes domain scientists (in biology, medicine, but also social sciences) who started relying heavily on computational resources, gathering huge amounts of data, and requiring a considerable amount of processing to analyze them. Our research is motivated by the growing discrepancy between on the one hand, the complexity of the workloads and the computing systems, and on the other hand, the expanding community of developers at large, with limited expertise to optimize and to map efficiently computations to compute nodes.

3.1.3. Domain constraints

Mobile, embedded systems have become ubiquitous. Many of them have real-time constraints. For this class of systems, correctness implies not only producing the correct result, but also doing so within specified deadlines. In the presence of heterogeneous, complex and highly dynamic systems, producing *tight* (i.e., useful) upper bound to the worst-case execution time has become extremely challenging. Our research will aim at improving the tightness as well as enlarging the set of features that can be safely analyzed.

The ever growing dependence of our economy on computing systems also implies that security has become of utmost importance. Many systems are under constant attacks from intruders. Protection has a cost also in terms of performance. We plan to leverage our background to contribute solutions that minimize this impact.

Note on Applications Domains.
PACAP works on fundamental technologies for computer science: processor architecture, performance-oriented compilation and guaranteed response time for real-time. The research results may have impact on any application domain that requires high performance execution (telecommunication, multimedia, biology, health, engineering, environment...), but also on many embedded applications that exhibit other constraints such as power consumption, code size and guaranteed response time.

We strive to extract from active domains the fundamental characteristics that are relevant to our research. For example, *big data* is of interest to PACAP because it relates to the study of hardware/software mechanisms to efficiently transfer huge amounts of data to the computing nodes. Similarly, the *Internet of Things* is of interest because it has implications in terms of ultra low-power consumption.

3.2. Research Objectives

Processor micro-architecture and compilation have been at the core of the research carried by the members of the project teams for two decades, with undeniable contributions. They continue to be the foundation of PACAP.

Heterogeneity and diversity of processor architectures now require new techniques to guarantee that the hardware is satisfactorily exploited by the software. One of our goals is to devise new static compilation techniques (cf. Section 3.2.1), but also build upon iterative [1] and split [2] compilation to continuously adapt software to its environment (Section 3.2.2). Dynamic binary optimization will also play a key role in delivering adapting software and increased performance.

The end of Moore's law and Dennard's scaling [0] offer an exciting window of opportunity, where performance improvements will no longer derive from additional transistor budget or increased clock frequency, but rather come from breakthroughs in micro-architecture (Section 3.2.3). Reconciling CPU and GPU designs (Section 3.2.4) is one of our objectives.

Heterogeneity and multicores are also major obstacles to determining tight worst-case execution times of real-time systems (Section 3.2.5), which we plan to tackle.

Finally, we also describe how we plan to address transversal aspects such as reliability (Section 3.2.6), power efficiency (Section 3.2.7), and security (Section 3.2.8).

[0] According to Dennard scaling, as transistors get smaller the power density remains constant, and the consumed power remains proportional to the area.

3.2.1. Static Compilation

Static compilation techniques continue to be relevant in addressing the characteristics of emerging hardware technologies, such as non-volatile memories, 3D-stacking, or novel communication technologies. These techniques expose new characteristics to the software layers. As an example, non-volatile memories typically have asymmetric read-write latencies (writes are much longer than reads) and different power consumption profiles. PACAP studies new optimization opportunities and develops tailored compilation techniques for upcoming compute nodes. New technologies may also be coupled with traditional solutions to offer new trade-offs. We study how programs can adequately exploit the specific features of the proposed heterogeneous compute nodes.

We propose to build upon iterative compilation [1] to explore how applications perform on different configurations. When possible, Pareto points are related to application characteristics. The best configuration, however, may actually depend on runtime information, such as input data, dynamic events, or properties that are available only at runtime. Unfortunately a runtime system has little time and means to determine the best configuration. For these reasons, we also leverage split-compilation [2]: the idea consists in pre-computing alternatives, and embedding in the program enough information to assist and drive a runtime system towards to the best solution.

3.2.2. Software Adaptation

More than ever, software needs to adapt to its environment. In most cases, this environment remains unknown until runtime. This is already the case when one deploys an application to a cloud, or an "app" to mobile devices. The dilemma is the following: for maximum portability, developers should target the most general device; but for performance they would like to exploit the most recent and advanced hardware features. JIT compilers can handle the situation to some extent, but binary deployment requires dynamic binary rewriting. Our work has shown how SIMD instructions can be upgraded from SSE to AVX transparently [3]. Many more opportunities will appear with diverse and heterogeneous processors, featuring various kinds of accelerators.

On shared hardware, the environment is also defined by other applications competing for the same computational resources. It becomes increasingly important to adapt to changing runtime conditions, such as the contention of the cache memories, available bandwidth, or hardware faults. Fortunately, optimizing at runtime is also an opportunity, because this is the first time the program is visible as a whole: executable and libraries (including library versions). Optimizers may also rely on dynamic information, such as actual input data, parameter values, etc. We have already developed a software platform [12] to analyze and optimize programs at runtime, and we started working on automatic dynamic parallelization of sequential code, and dynamic specialization.

We started addressing some of these challenges in ongoing projects such as Nano2017 PSAIC Collaborative research program with STMicroelectronics, as well as within the Inria Project Lab MULTICORE. The H2020 FET HPC project ANTAREX also addresses these challenges from the energy perspective. We further leverage our platform and initial results to address other adaptation opportunities. Efficient software adaptation requires expertise from all domains tackled by PACAP, and strong interaction between all team members is expected.

3.2.3. Research directions in uniprocessor micro-architecture

Achieving high single-thread performance remains a major challenge even in the multicore era (Amdahl's law). The members of the PACAP project-team have been conducting research in uniprocessor micro-architecture research for about 20 years covering major topics including caches, instruction front-end, branch prediction, out-of-order core pipeline, and value prediction. In particular, in recent years they have been recognized as world leaders in branch prediction [19][9] and in cache prefetching [7] and they have revived the forgotten concept of value prediction [11][10]. This research was supported by the ERC Advanced grant DAL (2011-2016) and also by Intel. We pursue research on achieving ultimate unicore performance. Below are several non-orthogonal directions that we have identified for mid-term research:

1. management of the memory hierarchy (particularly the hardware prefetching);
2. practical design of very wide issue execution cores;

3. speculative execution.

Memory design issues:
Performance of many applications is highly impacted by the memory hierarchy behavior. The interactions between the different components in the memory hierarchy and the out-of-order execution engine have high impact on performance.

The last *Data Prefetching Contest* held with ISCA 2015 has illustrated that achieving high prefetching efficiency is still a challenge for wide-issue superscalar processors, particularly those featuring a very large instruction window. The large instruction window enables an implicit data prefetcher. The interaction between this implicit hardware prefetcher and the explicit hardware prefetcher is still relatively mysterious as illustrated by Pierre Michaud's BO prefetcher (winner of DPC2) [7]. The first research objective is to better understand how the implicit prefetching enabled by the large instruction window interacts with the L2 prefetcher and then to understand how explicit prefetching on the L1 also interacts with the L2 prefetcher.

The second research objective is related to the interaction of prefetching and virtual/physical memory. On real hardware, prefetching is stopped by page frontiers. The interaction between TLB prefetching (and on which level) and cache prefetching must be analyzed.

The prefetcher is not the only actor in the hierarchy that must be carefully controlled. Significant benefits can also be achieved through careful management of memory access bandwidth, particularly the management of spatial locality on memory accesses, both for reads and writes. The exploitation of this locality is traditionally handled in the memory controller. However, it could be better handled if larger temporal granularity was available. Finally, we also intend to continue to explore the promising avenue of compressed caches. In particular we recently proposed the skewed compressed cache [13]. It offers new possibilities for efficient compression schemes.

Ultra wide-issue superscalar.
To effectively leverage memory level parallelism, one requires huge out-of-order execution structures as well as very wide issue superscalar processors. For the two past decades, implementing ever wider issue superscalar processors has been challenging. The objective of our research on the execution core is to explore (and revisit) directions that allow the design of a very wide-issue (8-to-16 way) out-of-order execution core while mastering its complexity (silicon area, hardware logic complexity, power/energy consumption).

The first direction that we are exploring is the use of clustered architectures [8]. Symmetric clustered organization allows to benefit from a simpler bypass network, but induce large complexity on the issue queue. One remarkable finding of our study [8] is that, when considering two large clusters (e.g. 8-wide), steering large groups of consecutive instructions (e.g. 64 μops) to the same cluster is quite efficient. This opens opportunities to limit the complexity of the issue queues (monitoring fewer buses) and register files (fewer ports and physical registers) in the clusters, since not all results have to be forwarded to the other cluster.

The second direction that we are exploring is associated with the approach that we developed with Sembrant et al. [15]. It reduces the number of instructions waiting in the instruction queues for the applications benefiting from very large instruction windows. Instructions are dynamically classified as ready (independent from any long latency instruction) or non-ready, and as urgent (part of a dependency chain leading to a long latency instruction) or non-urgent. Non-ready non-urgent instructions can be delayed until the long latency instruction has been executed; this allows to reduce the pressure on the issue queue. This proposition opens the opportunity to consider an asymmetric micro-architecture with a cluster dedicated to the execution of urgent instructions and a second cluster executing the non-urgent instructions. The micro-architecture of this second cluster could be optimized to reduce complexity and power consumption (smaller instruction queue, less aggressive scheduling...)

Speculative execution.
Out-of-order (OoO) execution relies on speculative execution that requires predictions of all sorts: branch, memory dependency, value...

The PACAP members have been major actors of branch prediction research for the last 20 years; and their proposals have influenced the design of most of the hardware branch predictors in current microprocessors. We will continue to steadily explore new branch predictor designs, as for instance [17].

In speculative execution, we have recently revisited value prediction (VP) which was a hot research topic between 1996 and 2002. However it was considered until recently that value prediction would lead to a huge increase in complexity and power consumption in every stage of the pipeline. Fortunately, we have recently shown that complexity usually introduced by value prediction in the OoO engine can be overcome [11][10][19][9]. First, very high accuracy can be enforced at reasonable cost in coverage and minimal complexity [11]. Thus, both prediction validation and recovery by squashing can be done outside the out-of-order engine, at commit time. Furthermore, we propose a new pipeline organization, EOLE ({Early | Out-of-order | Late} Execution), that leverages VP with validation at commit to execute many instructions outside the OoO core, in-order [10]. With EOLE, the issue-width in OoO core can be reduced without sacrificing performance, thus benefiting the performance of VP without a significant cost in silicon area and/or energy. In the near future, we will explore new avenues related to value prediction. These directions include register equality prediction and compatibility of value prediction with weak memory models in multiprocessors.

3.2.4. Towards heterogeneous single-ISA CPU-GPU architectures

Heterogeneous single-ISA architectures have been proposed in the literature during the 2000's [44] and are now widely used in the industry (Arm big.LITTLE, NVIDIA 4+1...) as a way to improve power-efficiency in mobile processors. These architectures include multiple cores whose respective micro-architectures offer different trade-offs between performance and energy efficiency, or between latency and throughput, while offering the same interface to software. Dynamic task migration policies leverage the heterogeneity of the platform by using the most suitable core for each application, or even each phase of processing. However, these works only tune cores by changing their complexity. Energy-optimized cores are either identical cores implemented in a low-power process technology, or simplified in-order superscalar cores, which are far from state-of-the-art throughput-oriented architectures such as GPUs.

We investigate the convergence of CPU and GPU at both architecture and compiler levels.

Architecture.
The architecture convergence between Single Instruction Multiple Threads (SIMT) GPUs and multicore processors that we have been pursuing [5] opens the way for heterogeneous architectures including latency-optimized superscalar cores and throughput-optimized GPU-style cores, which all share the same instruction set. Using SIMT cores in place of superscalar cores will enable the highest energy efficiency on regular sections of applications. As with existing single-ISA heterogeneous architectures, task migration will not necessitate any software rewrite and will accelerate existing applications.

Compilers for emerging heterogeneous architectures.
Single-ISA CPU+GPU architectures will provide the necessary substrate to enable efficient heterogeneous processing. However, it will also introduce substantial challenges at the software and firmware level. Task placement and migration will require advanced policies that leverage both static information at compile time and dynamic information at run-time. We are tackling the heterogeneous task scheduling problem at the compiler level.

3.2.5. Real-time systems

Safety-critical systems (e.g. avionics, medical devices, automotive...) have so far used simple unicore hardware systems as a way to control their predictability, in order to meet timing constraints. Still, many critical embedded systems have increasing demand in computing power, and simple unicore processors are not sufficient anymore. General-purpose multicore processors are not suitable for safety-critical real-time systems, because they include complex micro-architectural elements (cache hierarchies, branch, stride and value predictors) meant to improve average-case performance, and for which worst-case performance is difficult to predict. The prerequisite for calculating tight WCET is a deterministic hardware system that avoids dynamic, time-unpredictable calculations at run-time.

Even for multi and manycore systems designed with time-predictability in mind (Kalray MPPA manycore architecture [0], or the Recore manycore hardware [0]) calculating WCETs is still challenging. The following two challenges will be addressed in the mid-term:

1. definition of methods to estimate WCETs tightly on manycores, that smartly analyze and/or control shared resources such as buses, NoCs or caches;
2. methods to improve the programmability of real-time applications through automatic parallelization and optimizations from model-based designs.

3.2.6. Fault Tolerance

Technology trends suggest that, in tomorrow's computing world, failures will become commonplace due to many factors, and the expected probability of failure will increase with scaling. While well-known approaches, such as error correcting codes, exist to recover from failures and provide fault-free chips, the exponential growth of the number of faults will make them unaffordable in the future. Consequently, other approaches such as fine-grained disabling and reconfiguration of hardware elements (e.g. individual functional units or cache blocks) will become economically necessary. We are going to enter a new era: functionally correct chips with variable performance among chips and throughout their lifetime [45].

Transient and permanent faults may be detected by similar techniques, but correcting them generally involves different approaches. We are primarily interested in permanent faults, even though we do not necessarily disregard transient faults (e.g. the TMR approach in the next paragraph addresses both kinds of faults).

CPU.
Permanent faults can occur anywhere in the processor. The performance implications of faulty cells vary depending on how the array is used in a processor. Most of micro-architectural work aiming at assessing the performance implications of permanently faulty cells relies on simulations with random fault-maps. These studies are, therefore, limited by the fault-maps they use that may not be representative for the average and distributed performance. They also do not consider aging effects.

Considering the memory hierarchy, we have already studied [4] the impact of permanent faults on the average and worst-case performance based on analytical models. We will extend these models to cover other components and other designs, and to analyze the interaction between faulty components.

For identified critical hardware structures, such as the memory hierarchy, we will propose protection mechanisms by for instance using larger cells, or even by selecting a different array organization to mitigate the impact of faults.

Another approach to deal with faults is to introduce redundancy at the code level. We propose to consider static compilation techniques focusing on existing hardware. As an example, we plan to leverage SIMD extensions of current instruction sets to introduce redundancy in scalar code at minimum cost. With these instructions, it will be possible to protect the execution from both soft errors by using TMR (triple modular redundancy) with voters in the code itself, and permanent faults without the need of extra hardware support to deconfigure faulty functional units.

Reconfigurable Computing.
In collaboration with the CAIRN project-team, we propose to construct Coarse Grain Reconfigurable Architectures (CGRA) from a sea of basic arithmetic and memory elements organized into clusters and connected through a hierarchical interconnection network. These clusters of basic arithmetic operators (e.g. 8-bit arithmetic and logic units) would be able to be seamlessly configured to various accuracy and data types to adapt the consumed energy to application requirements taking advantage of approximate computations. We propose to add new kinds of error detection (and sometimes correction) directly at the operator level by taking advantage of the massive redundancy of the array. As an example, errors can be tracked and detected in a complex sequence of double floating-point operations by using a reduced-precision version of the same processing.

[0] http://www.kalrayinc.com
[0] http://www.recoresystems.com/

Such reconfigurable blocks will be driven by compilation techniques, in charge of computing checkpoints, detecting faults, and replaying computations when needed.

Dynamic compilation techniques will help better exploit faulty hardware, by allocating data and computations on correct resources. In case of permanent faults, we will provide a mechanism to reconfigure the hardware, for example by reducing the issue width of VLIW processors implemented in CGRA. Dynamic code generation (JIT compiler) will re-generate code for the new configuration, guaranteeing portability and optimal exploitation of the hardware.

3.2.7. Power efficiency

PACAP addresses power-efficiency at several levels. First, we design static and split compilation techniques to contribute to the race for Exascale computing (the general goal is to reach 10^{18} FLOP/s at less than 20 MW). Second, we focus on high-performance low-power embedded compute nodes. Within the ANR project Continuum, in collaboration with architecture and technology experts from LIRMM and the SME Cortus, we research new static and dynamic compilation techniques that fully exploit emerging memory and NoC technologies. Finally, in collaboration with the CAIRN project-team, we investigate the synergy of reconfigurable computing and dynamic code generation.

Green and heterogeneous high-performance computing.
Concerning HPC systems, our approach consists in mapping, runtime managing and autotuning applications for green and heterogeneous High-Performance Computing systems up to the Exascale level. One key innovation of the proposed approach consists of introducing a separation of concerns (where self-adaptivity and energy efficient strategies are specified aside to application functionalities) promoted by the definition of a Domain Specific Language (DSL) inspired by aspect-oriented programming concepts for heterogeneous systems. The new DSL will be introduced for expressing adaptivity/energy/performance strategies and to enforce at runtime application autotuning and resource and power management. The goal is to support the parallelism, scalability and adaptability of a dynamic workload by exploiting the full system capabilities (including energy management) for emerging large-scale and extreme-scale systems, while reducing the Total Cost of Ownership (TCO) for companies and public organizations.

High-performance low-power embedded compute nodes.
We will address the design of next generation energy-efficient high-performance embedded compute nodes. It focuses at the same time on software, architecture and emerging memory and communication technologies in order to synergistically exploit their corresponding features. The approach of the project is organized around three complementary topics: 1) compilation techniques; 2) multicore architectures; 3) emerging memory and communication technologies. PACAP will focus on the compilation aspects, taking as input the software-visible characteristics of the proposed emerging technology, and making the best possible use of the new features (non-volatility, density, endurance, low-power).

Hardware Accelerated JIT Compilation.
Reconfigurable hardware offers the opportunity to limit power consumption by dynamically adjusting the number of available resources to the requirements of the running software. In particular, VLIW processors can adjust the number of available issue lanes. Unfortunately, changing the processor width often requires recompiling the application, and VLIW processors are highly dependent of the quality of the compilation, mainly because of the instruction scheduling phase performed by the compiler. Another challenge lies in the high constraints of the embedded system: the energy and execution time overhead due to the JIT compilation must be carefully kept under control.

We started exploring ways to reduce the cost of JIT compilation targeting VLIW-based heterogeneous many-core systems. Our approach relies on a hardware/software JIT compiler framework. While basic optimizations and JIT management are performed in software, the compilation back-end is implemented by means of specialized hardware. This back-end involves both instruction scheduling and register allocation, which are known to be the most time-consuming stages of such a compiler.

3.2.8. Security

Security is a mandatory concern of any modern computing system. Various threat models have led to a multitude of protection solutions. Members of PACAP already contributed, thanks to the HAVEGE [48] random number generator, and code obfuscating techniques (the obfuscating just-in-time compiler [43], or thread-based control flow mangling [46]).

We partner with security experts who can provide intuition, know-how and expertise, in particular in defining threat models, and assessing the quality of the solutions. Our background in compilation and architecture helps design more efficient and less expensive protection mechanisms.

We already have ongoing research directions related to security. SECODE (Secure Codes to Thwart Cyber-physical Attacks) is a project started in January 2016, in collaboration with security experts from Télécom Paris Tech, Paris 8, Université Catholique de Louvain (Belgium), and University of Sabancı (Turkey). We also plan to partner with the Inria/CentraleSupelec CIDRE project-team to design a tainting technique based on a just-in-time compiler.

Compiler-based data protection.
We specify and design error correction codes suitable for an efficient protection of sensitive information in the context of Internet of Things (IoT) and connected objects. We partner with experts in security and codes to prototype a platform that demonstrates resilient software. PACAP's expertise is key to select and tune the protection mechanisms developed within the project, and to propose safe, yet cost-effective solutions from an implementation point of view.

JIT-based tainting.
Dynamic information flow control (DIFC, also known as *tainting*) is used to detect intrusions and to identify vulnerabilities. It consists in attaching metadata (called *taints* or *labels*) to information containers, and to propagate the taints when particular operations are applied to the containers: reads, writes, etc. The goal is then to guarantee that confidential information is never used to generate data sent to an untrusted container; conversely, data produced by untrusted entities cannot be used to update sensitive data.

The containers can be of various granularities: fine-grain approaches can deal with single variables, coarser-grain approaches consider a file as a whole. The CIDRE project-team has developed several DIFC monitors. kBlare is coarse-grain monitor in the Linux kernel. JBlare is a fine-grain monitor for Java applications. Fine-grain monitors provide a better precision at the cost of a significant overhead in execution time.

Combining the expertise of CIDRE in DIFC with our expertise in JIT compilation will help design hybrid approaches. An initial static analysis of the program prior to installation or execution will feed information to a dynamic analyzer that propagates taints during just-in-time compilation.

SUMO Project-Team

3. Research Program

3.1. Analysis and verification of quantitative systems

The overall objective of this axis is to develop the quantitative aspects of formal methods while maintaining the tractability of verification objectives and progressing toward the management of large systems. This covers the development of relevant modeling formalims, to nicely weave time, costs and probabilities with existing models for concurrency. We plan to further study time(d) Petri nets, networks of timed automata (with synchronous or asynchronous communications), stochastic automata, partially-observed Markov decision processes, etc. A second objective is to develop verification methods for such quantitative systems. This covers several aspects: quantitative verification questions (e.g. computing an optimal scheduling policy), Boolean questions on quantitative features (deciding whether some probability is greater than a threshold), robustness issues (will a system have the same behaviors if some parameter is slightly altered?), etc. Our goal is to explore the frontier between decidable and undecidable problems, or more pragmatically tractable and untractable problems. Of course, there is a tradeoff between the expressivity and the tractability of a model. Models that incorporate distributed aspects, probabilities, time, etc., are typically untractable. In such a case, abstraction or approximation techniques are a workaround that we will explore.

Here are some precise topics that we place in our agenda:

- analysis of diagnosability and opacity properties for stochastic systems;
- verification of time(d) Petri nets;
- robustness analysis for timed and/or stochastic systems;
- abstraction techniques for quantitative systems.

3.2. Control of quantitative systems

The main objective of this research axis is to explore the quantitative and/or distributed extensions of classical control problems. We envision control in its widest meaning of driving a system in order to guarantee or enforce some extra property (i.e. not guaranteed by the system alone), in a partially- or totally-observed setting. This property can either be logical (e.g. reachability or safety) or quantitative (e.g. reach some performance level). These problems have of course an offline facet (e.g. controller design, existence of a policy/strategy) and an online facet (e.g. algorithm to select some optimal action at runtime).

Our objectives comprise classical controler synthesis for discrete-event systems, with extensions to temporal/stochastic/reward settings. They also cover maintaining or maximizing extra properties such as diagnosability or opacity, for example in stochastic systems. We also target further analysis of POMDPs (partially-observed Markov decision processes), and multi-agent versions of policy synthesis relying on tools from game theory. We aim at adressing some control problems motivated by industrial applications, that raise issues like the optimal control of timed and stochastic discrete-event systems, with concerns like robustness to perturbations and multicriteria optimization. Finally, we also plan to work on modular testing, and on runtime enforcement techniques, in order to garantee extra logical and temporal properties to event flows.

3.3. Management of large or distributed systems

The generic terms of "supervision" or "management" of distributed systems cover problems like control, diagnosis, sensor placement, planning, optimization, (state) estimation, parameter identification, testing, etc. This research axis examines how classical settings for such problems can scale up to large or distributed systems. Our work will be driven by considerations like: how to take advantage of modularity, how to design approximate management algorithms, how to design relevant abstractions to make large systems more tractable, how to deal with models of unknown size, how to design mechanisms to obtain relevant models, etc.

As more specific objectives, let us mention:

- Parametric-size systems: how to verify properties of distributed systems with an unknown number of components;
- Approximate management methods: we will explore the extension of ideas developed for Bayesian inference in large-scale stochastic systems (such as turbo-algorithms) to the field of modular dynamic systems. When component interactions are sparse, even if exact management methods are unaccessible (for diagnosis, planning, control, etc.), good approximations based on local computations may be accessible;
- Model abstraction: we will explore techniques to design more tractable abstractions of stochastic dynamic systems defined on large sets of variables;
- Self-modelling, which consists in managing large-scale systems that are known by their building rules, but where the specific instance is only discovered on-the-fly at runtime. The model of the managed system is built on-line, following the needs of the management algorithms;
- Distributed control: we will tackle issues related to asynchronous communications between local controllers, and to abstraction techniques allowing to address large systems;
- Test and enforcement: we will tackle coverage issues for the test of large systems, and the test and enforcement of properties for timed models, or for systems handling data.

3.4. Data driven systems

Data-driven systems are systems whose behaviour depends both on explicit workflows (scheduling and durations of tasks, calls to possibly distant services, ...) and on the data processed by the system (stored data, parameters of a request, results of a request, ...). This family of systems covers workflows that convey data (business processes or information systems), transactional systems (web stores), large databases managed with rules (banking systems), collaborative environments (crowds, health systems), etc. These systems are distributed, modular, and open: they integrate components and sub-services distributed over the web, and accept requests from clients. Our objective is to provide validation and supervision tools for such systems. To achieve this goal, we have to solve several challenging tasks:

- provide realistic models, and sound automated abstraction techniques, to reason on models that are reasonable abstractions of real systems. These models should be able to encompass modularity, distribution, in a context where workflows and data aspects are tightly connected;
- address design of data driven systems in a declarative way: declarative models are another way to handle data-driven systems. Rather than defining the explicit workflows and their effects on data, rule-based models state how actions are enacted in terms of the shape (pattern matching) or value of the current data. We think that distributed rewriting rules or attributed grammars can provide a practical yet formal framework for maintenance, by providing a solution to update mandatory documentation during the lifetime of an artifact.
- provide tractable solutions for validation of models: frequent issues are safety questions (can a system reach some bad configuration?), but also liveness (workflows progess), ... These questions should not only remain decidable on our models, but also with efficient computational methods.
- address QoS management in large reconfigurable systems: data-driven distributed systems often have constraints in terms of QoS. This QoS questions adress performance issues, but also data quality. This calls for an analysis of quantitative features and for reconfiguration techniques to meet desired QoS.

TAMIS Project-Team

3. Research Program

3.1. Axis 1: Vulnerability analysis

This axis proposes different techniques to discover vulnerabilities in systems. The outcomes of this axis are (a) new techniques to discover system vulnerabilities as well as to analyze them, and (b) to understand the importance of the hardware support.

Most existing approaches used at the engineering level rely on testing and fuzzing. Such techniques consist in simulating the system for various input values, and then checking that the result conforms to a given standard. The problem being the large set of inputs to be potentially tested. Existing solutions propose to extract significant sets by mutating a finite set of inputs. Other solutions, especially concolic testing developed at Microsoft, propose to exploit symbolic executions to extract constraints on new values. We build on those existing work, and extend them with recent techniques based on dissimilarity distances and learning. We also account for the execution environment, and study techniques based on the combination of timing attacks with fuzzing techniques to discover and classify classes of behavior of the system under test.

Techniques such as model checking and static analysis have been used for verifying several types of requirements such as safety and reliability. Recently, several works have attempted to adapt model checking to the detection of security issues. It has clearly been identified that this required to work at the level of binary code. Applying formal techniques to such code requires the development of disassembly techniques to obtain a semantically well-defined model. One of the biggest issues faced with formal analysis is the state space explosion problem. This problem is amplified in our context as representations of data (such as stack content) definitively blow up the state space. We propose to use statistical model checking (SMC) of rare events to efficiently identify problematic behaviors.

We also seek to understand vulnerabilities at the architecture and hardware levels. Particularly, we evaluate vulnerabilities of the interfaces and how an adversary could use them to get access to core assets in the system. One particular mechanism to be investigated is the DMA and the so-called Trustzone. An ad-hoc technique to defend against adversarial DMA-access to memory is to keep key material exclusively in registers. This implies co-analyzing machine code and an accurate hardware model.

3.2. Axis 2: Malware analysis

Axis 1 is concerned with vulnerabilities. Such vulnerabilities can be exploited by an attacker in order to introduce malicious behaviors in a system. Another method to identify vulnerabilities is to analyze malware that exploits them. However, modern malware has a wide variety of analysis avoidance techniques. In particular, attackers obfuscate the code leading to a security exploit. For doing so, recent black hat research suggests hiding constants in program choices via polynomials. Such techniques hinder forensic analysis by making detailed analysis labor intensive and time consuming. The objective of research axis 2 is to obtain a full tool chain for malware analysis starting from (a) the observability of the malware via deobfuscation, and (b) the analysis of the resulting binary file. A complementary objective is to understand how hardware attacks can be exploited by malwares.

We first investigate obfuscation techniques. Several solutions exist to mitigate the packer problem. As an example, we try to reverse the packer and remove the environment evaluation in such a way that it performs the same actions and outputs the resulting binary for further analysis. There is a wide range of techniques to obfuscate malware, which includes flattening and virtualization. We will produce a taxonomy of both techniques and tools. We will first give a particular focus to control flow obfuscation via mixed Boolean algebra, which is highly deployed for malware obfuscation. We recently showed that a subset of them can be broken via SAT-solving and synthesis. Then, we will expand our research to other obfuscation techniques.

Once the malware code has been unpacked/deobfuscated, the resulting binary still needs to be fully understood. Advanced malware often contains multiple stages, multiple exploits and may unpack additional features based on its environment. Ensuring that one understands all interesting execution paths of a malware sample is related to enumerating all of the possible execution paths when checking a system for vulnerabilities. The main difference is that in one case we are interested in finding vulnerabilities and in the other in finding exploitative behavior that may mutate. Still, some of the techniques of Axis 1 can be helpful in analyzing malware. The main challenge for axis 2 is thus to adapt the tools and techniques to deal with binary programs as inputs, as well as the logic used to specify malware behavior, including behavior with potentially rare occurrences. Another challenge is to take mutation into account, which we plan to do by exploiting mining algorithms.

Most recent attacks against hardware are based on fault injection which dynamically modifies the semantics of the code. We demonstrated the possibility to obfuscate code using constraint solver in such a way that the code becomes intentionally hostile while hit by a laser beam. This new form of obfuscation opens a new challenge for secure devices where malicious programs can be designed and uploaded that defeat comprehensive static analysis tools or code reviews, due to their multi-semantic nature. We have shown on several products that such an attack cannot be mitigated with the current defenses embedded in Java cards. In this research, we first aim at extending the work on fault injection, then at developing new techniques to analyze such hostile code. This is done by proposing formal models of fault injection, and then reusing results from our work on obfuscation/deobfuscation.

3.3. Axis 3: Building a secure network stack

Christian Grothoff, who leads this axis, got a position in Bern in 2017. This axis followed him, although TAMIS still held during 2018 expertise and members to finish ongoing work with the team.

TEA Project-Team

3. Research Program

3.1. Previous Works

The challenges of team TEA support the claim that sound Cyber-Physical System design (including embedded, reactive, and concurrent systems altogether) should consider multi-form time models as a central aspect. In this aim, architectural specifications found in software engineering are a natural focal point to start from. Architecture descriptions organize a system model into manageable components, establish clear interfaces between them, collect domain-specific constraints and properties to help correct integration of components during system design. The definition of a formal design methodology to support heterogeneous or multi-form models of time in architecture descriptions demands the elaboration of sound mathematical foundations and the development of formal calculi and methods to instrument them. This constitutes the research program of team TEA.

System design based on the “synchronous paradigm” has focused the attention of many academic and industrial actors on abstracting non-functional implementation details from system design. This elegant design abstraction focuses on the logic of interaction in reactive programs rather than their timed behavior, allowing to secure functional correctness while remaining an intuitive programming model for embedded systems. Yet, it corresponds to embedded technologies of single cores and synchronous buses from the 90s, and may hardly cover the semantic diversity of distribution, parallelism, heterogeneity, of cyber-physical systems found in 21st century Internet-connected, true-time^{TM}-synchronized clouds, of tomorrow’s grids.

By contrast with a synchronous hypothesis, yet from the same era, the polychronous MoCC is inherently capable of describing multi-clock abstractions of GALS systems. Polychrony is implemented in the data-flow specification language Signal, available in the Eclipse project POP [0] and in the CCSL standard [0] available from the TimeSquare project. Both provide tooled infrastructures to refine high-level specifications into real-time streaming applications or locally synchronous and globally asynchronous systems, through a series of model analysis, verification, and synthesis services. These tool-supported refinement and transformation techniques can assist the system engineer from the earliest design stages of requirement specification to the latest stages of synthesis, scheduling and deployment. These characteristics make polychrony much closer to the required semantic for compositional, refinement-based, architecture-driven, system design.

While polychrony was a step ahead of the traditional synchronous hypothesis, CCSL is a leap forward from synchrony and polychrony. The essence of CCSL is “multi-form time” toward addressing all of the domain-specific physical, electronic and logical aspects of cyber-physical system design.

3.2. Modeling Times

To make a sense and eventually formalize the semantics of time in system design, we should most certainly rely on algebraic representations of time found in previous works and introduce the paradigm of "time systems" (type systems to represent time) in a way reminiscent to CCSL. Just as a type system abstracts data carried along operations in a program, a time system abstracts the causal interaction of that program module or hardware element with its environment, its pre and post conditions, its assumptions and guarantees, either logical or numerical, discrete or continuous. Some fundamental concepts of the time systems we envision are present in the clock calculi found in data-flow synchronous languages like Signal or Lustre, yet bound to a particular model of concurrency, hence time.

[0] *Polychrony on Polarsys*, https://www.polarsys.org/projects/polarsys.pop

[0] *Clock Constraints in UML/MARTE CCSL*. C. André, F. Mallet. RR-6540. Inria, 2008. http://hal.inria.fr/inria-00280941

In particular, the principle of refinement type systems [0], is to associate information (data-types) inferred from programs and models with properties pertaining, for instance, to the algebraic domain on their value, or any algebraic property related to its computation: effect, memory usage, pre-post condition, value-range, cost, speed, time, temporal logic [0]. Being grounded on type and domain theories, a time system should naturally be equipped with program analysis techniques based on type inference (for data-type inference) or abstract interpretation (for program properties inference) to help establish formal relations between heterogeneous component "types". Just as a time calculus may formally abstract timed concurrent behaviors of system components, timed relations (abstraction and refinement) represent interaction among components.

Scalability and compositionality requires the use of assume-guarantee reasoning to represent them, and to facilitate composition by behavioral sub-typing, in the spirit of the (static) contract-based formalism proposed by Passerone et al. [0]. Verification problems encompassing heterogeneously timed specifications are common and of great variety: checking correctness between abstract and concrete time models relates to desynchronisation (from synchrony to asynchrony) and scheduling analysis (from synchrony to hardware). More generally, they can be perceived from heterogeneous timing viewpoints (e.g. mapping a synchronous-time software on a real-time middle-ware or hardware).

This perspective demands capabilities not only to inject time models one into the other (by abstract interpretation, using refinement calculi), to compare time abstractions one another (using simulation, refinement, bi-simulation, equivalence relations) but also to prove more specific properties (synchronization, determinism, endochrony). All this formalization effort will allow to effectively perform the tooled validation of common cross-domain properties (e.g. cost v.s. power v.s. performance v.s. software mapping) and tackle equally common yet tough case studies such as these linking battery capacity, to on-board CPU performance, to static software schedulability, to logical software correctness and plant controllability (by the choice of the correct sampling period across system components).

3.3. Modeling Architectures

To address the formalization of such cross-domain case studies, modeling the architecture formally plays an essential role. An architectural model represents components in a distributed system as boxes with well-defined interfaces, connections between ports on component interfaces, and specifies component properties that can be used in analytical reasoning about the model. Several architectural modeling languages for embedded systems have emerged in recent years, including the SAE AADL [0], SysML [0], UML MARTE [0].

In system design, an architectural specification serves several important purposes. First, it breaks down a system model into manageable components to establish clear interfaces between components. In this way, complexity becomes manageable by hiding details that are not relevant at a given level of abstraction. Clear, formally defined, component interfaces allow us to avoid integration problems at the implementation phase. Connections between components, which specify how components affect each other, help propagate the effects of a change in one component to the linked components.

Most importantly, an architectural model is a repository to share knowledge about the system being designed. This knowledge can be represented as requirements, design artifacts, component implementations, held together by a structural backbone. Such a repository enables automatic generation of analytical models for different aspects of the system, such as timing, reliability, security, performance, energy, etc. Since all the models are generated from the same source, the consistency of assumptions w.r.t. guarantees, of abstractions w.r.t. refinements, used for different analyses becomes easier, and can be properly ensured in a design methodology based on formal verification and synthesis methods.

[0] *Abstract Refinement Types*. N. Vazou, P. Rondon, and R. Jhala. European Symposium on Programming. Springer, 2013.

[0] *LTL types FRP*. A. Jeffrey. Programming Languages meets Program Verification.

[0] *A contract-based formalism for the specification of heterogeneous systems*. L. Benvenistu, et al. FDL, 2008

[0] *Architecture Analysis and Design Language*, AS-5506. SAE, 2004. http://standards.sae.org/as5506b

[0] *System modeling Language*. OMG, 2007. http://www.omg.org/spec/SysML

[0] *UML Profile for MARTE*. OMG, 2009. http://www.omg.org/spec/MARTE

Related works in this aim, and closer in spirit to our approach (to focus on modeling time) are domain-specific languages such as Prelude [0] to model the real-time characteristics of embedded software architectures. Conversely, standard architecture description languages could be based on algebraic modeling tools, such as interface theories with the ECDAR tool [0].

In project TEA, it takes form by the normalization of the AADL standard's formal semantics and the proposal of a time specification annex in the form of related standards, such as CCSL, to model concurrency time and physical properties, and PSL, to model timed traces.

3.4. Scheduling Theory

Based on sound formalization of time and CPS architectures, real-time scheduling theory provides tools for predicting the timing behavior of a CPS which consists of many interacting software and hardware components. Expressing parallelism among software components is a crucial aspect of the design process of a CPS. It allows for efficient partition and exploitation of available resources.

The literature about real-time scheduling [0] provides very mature schedulability tests regarding many scheduling strategies, preemptive or non-preemptive scheduling, uniprocessor or multiprocessor scheduling, etc. Scheduling of data-flow graphs has also been extensively studied in the past decades.

A milestone in this prospect is the development of abstract affine scheduling techniques [0]. It consists, first, of approximating task communication patterns (e.g. between Safety-Critical Java threads) using cyclo-static data-flow graphs and affine functions. Then, it uses state of the art ILP techniques to find optimal schedules and to concretize them as real-time schedules in the program implementations [0][0].

Abstract scheduling, or the use of abstraction and refinement techniques in scheduling borrowed to the theory of abstract interpretation [0] is a promising development toward tooled methodologies to orchestrate thousands of heterogeneous hardware/software blocks on modern CPS architectures (just consider modern cars or aircrafts). It is an issue that simply defies the state of the art and known bounds of complexity theory in the field, and consequently requires a particular focus.

To develop the underlying theory of this promising research topic, we first need to deepen the theoretical foundation to establish links between scheduling analysis and abstract interpretation. A theory of time systems would offer the ideal framework to pursue this development. It amounts to representing scheduling constraints, inferred from programs, as types or contract properties. It allows to formalize the target time model of the scheduler (the architecture, its middle-ware, its real-time system) and defines the basic concepts to verify assumptions made in one with promises offered by the other: contract verification or, in this case, synthesis.

3.5. Verified programming for system design

The IoT is a network of devices that sense, actuate and change our immediate environment. Against this fundamental role of sensing and actuation, design of edge devices often treats action and event timing to be primarily software implementation issues: programming models for IoT abstract even the most rudimentary information regarding timing, sensing and the effects of actuation. As a result, applications programming interfaces for IoT allow wiring systems fast without any meaningful assertions about correctness, reliability or resilience.

We make the case that the "API glue" must give way to the logical closure of interface types. Interfaces can be governed by a calculus – a refinement type calculus – to enable reasoning on time, sensing and actuation, in a way that provides both deep specification refinement, for mechanized verification of requirements, and multi-layered abstraction, to support compositionality and scalability, from one end of the system to the other.

[0] *The Prelude language*. LIFL and ONERA, 2012. http://www.lifl.fr/~forget/prelude.html

[0] *PyECDAR, timed games for timed specifications*. Inria, 2013. https://project.inria.fr/pyecdar

[0] *A survey of hard real-time scheduling for multiprocessor systems*. R. I. Davis and A. Burns. *ACM Computing Survey* 43(4), 2011.

[0] *Buffer minimization in EDF scheduling of data-flow graphs*. A. Bouakaz and J.-P. Talpin. LCTES, ACM, 2013.

[0] *ADFG for the synthesis of hard real-time applications*. A. Bouakaz, J.-P. Talpin, J. Vitek. ACSD, IEEE, June 2012.

[0] *Design of SCJ Level 1 Applications Using Affine Abstract Clocks*. A. Bouakaz and J.-P. Talpin. SCOPES, ACM, 2013.

[0] *La vérification de programmes par interprétation abstraite*. P. Cousot. Séminaire au Collège de France, 2008.

Our project seeks to elevate the “function as type” paradigm to that of “system as type”: to define a refinement type calculus for reasoning on networked devices and integrate them as cyber-physical systems [0]. Our companion paper [0] outlines our progress in this aim and plans towards building a verified programming environment for networked IoT devices: we propose a type-driven approach to verifying and building safe and secure IoT applications.

Accounting for such constrains in a more principled fashion demands reasoning about the composition of all the software and hardware components of the application. Our proposed framework takes a step in this direction by (1) using refinement types to make make physical constraints explicit and (2) imposing an event-driven programming discipline to simplify the reasoning of system-wide properties to that of an event queue. In taking this approach, our framework makes it possible for developers to build verified IoT application by making it a type error for code to violate physical constraints.

[0] Refinement types for system design. Jean-Pierre Talpin. FDL’18 keynote.

[0] Steps toward verified programming of embedded computing systems. Jean-Pierre Talpin, Jean-Joseph Marty, Deian Stefan, Shravan Nagarayan, Rajesh Gupta, DATE’18.

I4S Project-Team

3. Research Program

3.1. Vibration analysis

In this section, the main features for the key monitoring issues, namely identification, detection, and diagnostics, are provided, and a particular instantiation relevant for vibration monitoring is described.

It should be stressed that the foundations for identification, detection, and diagnostics, are fairly general, if not generic. Handling high order linear dynamical systems, in connection with finite elements models, which call for using subspace-based methods, is specific to vibration-based SHM. Actually, one particular feature of model-based sensor information data processing as exercised in I4S, is the combined use of black-box or semi-physical models together with physical ones. Black-box and semi-physical models are, for example, eigenstructure parameterizations of linear MIMO systems, of interest for modal analysis and vibration-based SHM. Such models are intended to be identifiable. However, due to the large model orders that need to be considered, the issue of model order selection is really a challenge. Traditional advanced techniques from statistics such as the various forms of Akaike criteria (AIC, BIC, MDL, ...) do not work at all. This gives rise to new research activities specific to handling high order models.

Our approach to monitoring assumes that a model of the monitored system is available. This is a reasonable assumption, especially within the SHM areas. The main feature of our monitoring method is its intrinsic ability to the early warning of small deviations of a system with respect to a reference (safe) behavior under usual operating conditions, namely without any artificial excitation or other external action. Such a normal behavior is summarized in a reference parameter vector θ_0, for example a collection of modes and mode-shapes.

3.1.1. Identification

The behavior of the monitored continuous system is assumed to be described by a parametric model $\{\mathbf{P}_\theta \, , \, \theta \in \Theta\}$, where the distribution of the observations $(Z_0, ..., Z_N)$ is characterized by the parameter vector $\theta \in \Theta$.

For reasons closely related to the vibrations monitoring applications, we have been investigating subspace-based methods, for both the identification and the monitoring of the eigenstructure (λ, ϕ_λ) of the state transition matrix F of a linear dynamical state-space system :

$$\left\{ \begin{array}{rcl} X_{k+1} & = & F \; X_k + V_{k+1} \\ Y_k & = & H \; X_k + W_k \end{array} \right. , \tag{1}$$

namely the $(\lambda, \varphi_\lambda)$ defined by :

$$\det \; (F - \lambda \; I) = 0, \quad (F - \lambda \; I) \; \phi_\lambda = 0, \quad \varphi_\lambda \stackrel{\Delta}{=} H \; \phi_\lambda \tag{2}$$

The (canonical) parameter vector in that case is :

$$\theta \stackrel{\Delta}{=} \left(\begin{array}{c} \Lambda \\ \mathrm{vec}\Phi \end{array} \right) \tag{3}$$

where Λ is the vector whose elements are the eigenvalues λ, Φ is the matrix whose columns are the φ_λ's, and vec is the column stacking operator.

Subspace-based methods is the generic name for linear systems identification algorithms based on either time domain measurements or output covariance matrices, in which different subspaces of Gaussian random vectors play a key role [57].

Let $R_i \stackrel{\Delta}{=} \mathbf{E}\left(Y_k\ Y_{k-i}^T\right)$ and:

$$\mathcal{H}_{p+1,q} \stackrel{\Delta}{=} \begin{pmatrix} R_1 & R_2 & \vdots & R_q \\ R_2 & R_3 & \vdots & R_{q+1} \\ \vdots & \vdots & \vdots & \vdots \\ R_{p+1} & R_{p+2} & \vdots & R_{p+q} \end{pmatrix} \stackrel{\Delta}{=} \text{Hank}\left(R_i\right) \tag{4}$$

be the output covariance and Hankel matrices, respectively; and: $G \stackrel{\Delta}{=} \mathbf{E}\left(X_k Y_{k-1}^T\right)$. Direct computations of the R_i's from the equations (4) lead to the well known key factorizations :

$$\begin{aligned} R_i &= HF^{i-1}G \\ \mathcal{H}_{p+1,q} &= \mathcal{O}_{p+1}(H,F)\ \mathcal{C}_q(F,G) \end{aligned} \tag{5}$$

where:

$$\mathcal{O}_{p+1}(H,F) \stackrel{\Delta}{=} \begin{pmatrix} H \\ HF \\ \vdots \\ HF^p \end{pmatrix} \quad \text{and} \quad \mathcal{C}_q(F,G) \stackrel{\Delta}{=} (G\ FG\ \cdots\ F^{q-1}G) \tag{6}$$

are the observability and controllability matrices, respectively. The observation matrix H is then found in the first block-row of the observability matrix $\mathcal{O}$. The state-transition matrix F is obtained from the shift invariance property of $\mathcal{O}$. The eigenstructure (λ, ϕ_λ) then results from (5).

Since the actual model order is generally not known, this procedure is run with increasing model orders.

3.1.2. Detection

Our approach to on-board detection is based on the so-called asymptotic statistical local approach. It is worth noticing that these investigations of ours have been initially motivated by a vibration monitoring application example. It should also be stressed that, as opposite to many monitoring approaches, our method does not require repeated identification for each newly collected data sample.

For achieving the early detection of small deviations with respect to the normal behavior, our approach generates, on the basis of the reference parameter vector θ_0 and a new data record, indicators which automatically perform :

- The early detection of a slight mismatch between the model and the data;
- A preliminary diagnostics and localization of the deviation(s);
- The tradeoff between the magnitude of the detected changes and the uncertainty resulting from the estimation error in the reference model and the measurement noise level.

These indicators are computationally cheap, and thus can be embedded. This is of particular interest in some applications, such as flutter monitoring.

Choosing the eigenvectors of matrix F as a basis for the state space of model (4) yields the following representation of the observability matrix:

$$\mathcal{O}_{p+1}(\theta) = \begin{pmatrix} \Phi \\ \Phi\Delta \\ \vdots \\ \Phi\Delta^p \end{pmatrix} \tag{7}$$

where $\Delta \stackrel{\Delta}{=} \mathrm{diag}(\Lambda)$, and Λ and Φ are as in (6). Whether a nominal parameter θ_0 fits a given output covariance sequence $(R_j)_j$ is characterized by:

$$\mathcal{O}_{p+1}(\theta_0) \text{ and } \mathcal{H}_{p+1,q} \text{ have the same left kernel space.} \tag{8}$$

This property can be checked as follows. From the nominal θ_0, compute $\mathcal{O}_{p+1}(\theta_0)$ using (10), and perform e.g. a singular value decomposition (SVD) of $\mathcal{O}_{p+1}(\theta_0)$ for extracting a matrix U such that:

$$U^T\, U = I_s \text{ and } U^T\, \mathcal{O}_{p+1}(\theta_0) = 0 \tag{9}$$

Matrix U is not unique (two such matrices relate through a post-multiplication with an orthonormal matrix), but can be regarded as a function of θ_0. Then the characterization writes:

$$U(\theta_0)^T\, \mathcal{H}_{p+1,q} = 0 \tag{10}$$

3.1.2.1. *Residual associated with subspace identification.*

Assume now that *a reference* θ_0 *and a new sample* $Y_1, \cdots, Y_N$ *are available.* For checking whether the data agree with θ_0, the idea is to compute the empirical Hankel matrix $\widehat{\mathcal{H}}_{p+1,q}$:

$$\widehat{\mathcal{H}}_{p+1,q} \stackrel{\Delta}{=} \mathrm{Hank}\left(\widehat{R}_i\right), \quad \widehat{R}_i \stackrel{\Delta}{=} 1/(N-i) \sum_{k=i+1}^{N} Y_k\, Y_{k-i}^T \tag{11}$$

and to define the residual vector:

$$\zeta_N(\theta_0) \stackrel{\Delta}{=} \sqrt{N}\ \mathrm{vec}\left(U(\theta_0)^T\, \widehat{\mathcal{H}}_{p+1,q}\right) \tag{12}$$

Let θ be the actual parameter value for the system which generated the new data sample, and $\mathbf{E}_\theta$ be the expectation when the actual system parameter is θ. From (13), we know that $\zeta_N(\theta_0)$ has zero mean when no change occurs in θ, and nonzero mean if a change occurs. Thus $\zeta_N(\theta_0)$ plays the role of a residual.

As in most fault detection approaches, the key issue is to design a *residual*, which is ideally close to zero under normal operation, and has low sensitivity to noises and other nuisance perturbations, but high sensitivity to small deviations, before they develop into events to be avoided (damages, faults, ...). The originality of our approach is to :

- *Design* the residual basically as a *parameter estimating function*,
- *Evaluate* the residual thanks to a kind of central limit theorem, stating that the residual is asymptotically Gaussian and reflects the presence of a deviation in the parameter vector through a change in its own mean vector, which switches from zero in the reference situation to a non-zero value.

The central limit theorem shows [51] that the residual is asymptotically Gaussian :

$$\zeta_N \xrightarrow[N \to \infty]{} \begin{cases} \mathcal{N}(0, \Sigma) & \text{under } \mathbf{P}_{\theta_0} \ , \\ \\ \mathcal{N}(\mathcal{J}\,\eta, \Sigma) & \text{under } \mathbf{P}_{\theta_0+\eta/\sqrt{N}} \ , \end{cases} \tag{13}$$

where the asymptotic covariance matrix Σ can be estimated, and manifests the deviation in the parameter vector by a change in its own mean value. Then, deciding between $\eta = 0$ and $\eta \neq 0$ amounts to compute the following χ^2-test, provided that $\mathcal{J}$ is full rank and Σ is invertible :

$$\chi^2 = \bar{\zeta}^T \, \mathbf{F}^{-1} \, \bar{\zeta} \gtrless \lambda \ . \tag{14}$$

where

$$\bar{\zeta} \triangleq \mathcal{J}^T \, \Sigma^{-1} \, \zeta_N \quad \text{and} \quad \mathbf{F} \triangleq \mathcal{J}^T \, \Sigma^{-1} \, \mathcal{J} \tag{15}$$

3.1.3. Diagnostics

A further monitoring step, often called *fault isolation*, consists in determining which (subsets of) components of the parameter vector θ have been affected by the change. Solutions for that are now described. How this relates to diagnostics is addressed afterwards.

The question: *which (subsets of) components of θ have changed ?*, can be addressed using either nuisance parameters elimination methods or a multiple hypotheses testing approach [50].

In most SHM applications, a complex physical system, characterized by a generally non identifiable parameter vector Φ has to be monitored using a simple (black-box) model characterized by an identifiable parameter vector θ. A typical example is the vibration monitoring problem for which complex finite elements models are often available but not identifiable, whereas the small number of existing sensors calls for identifying only simplified input-output (black-box) representations. In such a situation, two different diagnosis problems may arise, namely diagnosis in terms of the black-box parameter θ and diagnosis in terms of the parameter vector Φ of the underlying physical model.

The isolation methods sketched above are possible solutions to the former. Our approach to the latter diagnosis problem is basically a detection approach again, and not a (generally ill-posed) inverse problem estimation approach.

The basic idea is to note that the physical sensitivity matrix writes $\mathcal{J} \, \mathcal{J}_{\Phi\theta}$, where $\mathcal{J}_{\Phi\theta}$ is the Jacobian matrix at Φ_0 of the application $\Phi \mapsto \theta(\Phi)$, and to use the sensitivity test for the components of the parameter vector Φ. Typically this results in the following type of directional test :

$$\chi^2_\Phi = \zeta^T \, \Sigma^{-1} \, \mathcal{J} \, \mathcal{J}_{\Phi\theta} \, (\mathcal{J}^T_{\Phi\theta} \, \mathcal{J}^T \, \Sigma^{-1} \, \mathcal{J} \, \mathcal{J}_{\Phi\theta})^{-1} \, \mathcal{J}^T_{\Phi\theta} \, \mathcal{J}^T \, \Sigma^{-1} \, \zeta \gtrless \lambda \ . \tag{16}$$

It should be clear that the selection of a particular parameterization Φ for the physical model may have a non-negligible influence on such type of tests, according to the numerical conditioning of the Jacobian matrices $\mathcal{J}_{\Phi\theta}$.

3.2. Thermal methods

3.2.1. Infrared thermography and heat transfer

This section introduces the infrared radiation and its link with the temperature, in the next part different measurement methods based on that principle are presented.

3.2.1.1. *Infrared radiation*

Infrared is an electromagnetic radiation having a wavelength between $0.2\mu m$ and $1\ mm$, this range begin in uv spectrum and it ends on the microwaves domain, see Figure 1 .

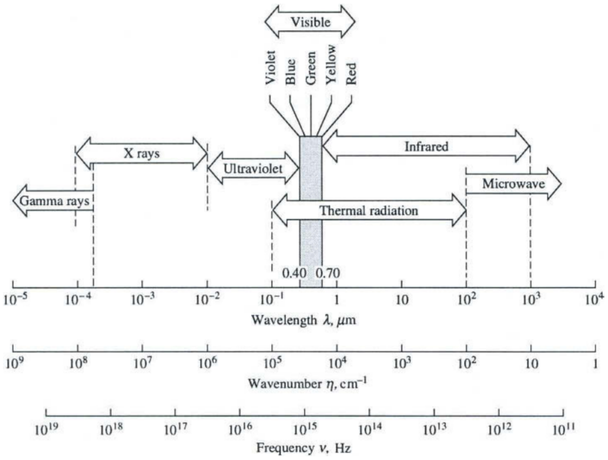

Figure 1. Electromagnetic spectrum - Credit MODEST, M.F. (1993). Radiative Heat Transfer. Academic Press.

For scientific purposes, infrared can be divided in three ranges of wavelength in which the application varies, see Table 1 .

Table 1. Wavelength bands in the infrared according to ISO 20473:2007

Band name	**wavelength**	**Uses ∖ definition**
Near infrared (PIR, IR-A, NIR)	$0.7-3\mu$m	Reflected solar heat flux
Mid infrared (MIR, IR-B)	$3-50\mu$m	Thermal infrared
Far infrared (LIR, IR-C, FIR)	$50-1000\mu$m	Astronomy

Our work is concentrated in the mid infrared spectral band. Keep in mind that Table 1 represents the ISO 20473 division scheme, in the literature boundaries between bands can move slightly.

The Plank's law, proposed by Max Planck in 1901, allows to compute the black body emission spectrum for various temperatures (and only temperatures), see Figure 2 left. The black body is a theoretical construction, it represents perfect energy emitter at a given temperature, cf. Equation (20).

$$M^o_{\lambda,T} = \frac{C_1\lambda^{-5}}{\exp^{\frac{C_2}{\lambda T}} -1} \tag{17}$$

With λ the wavelength in m and T as the temperature in Kelvin. The C_1 and C_2 constants, respectively in W.m^2 and m.K are defined as follow:

$$\begin{aligned} C_1 &= 2hc^2\pi \\ C_2 &= h\frac{c}{k} \end{aligned} \tag{18}$$

with

- c, the electromagnetic wave speed (in vacuum c is the light speed in m.s^{-1}).
- $k = 1.381e^{-23}$ J.K^{-1} The Boltzmann (Entropy definition from Ludwig Boltzmann 1873). It can be seen as a proportionality factor between the temperature and the energy of a system.
- $h \approx 6,62606957e^{-34}$ J.s The Plank constant. It is the link between the photons energy and their frequency.

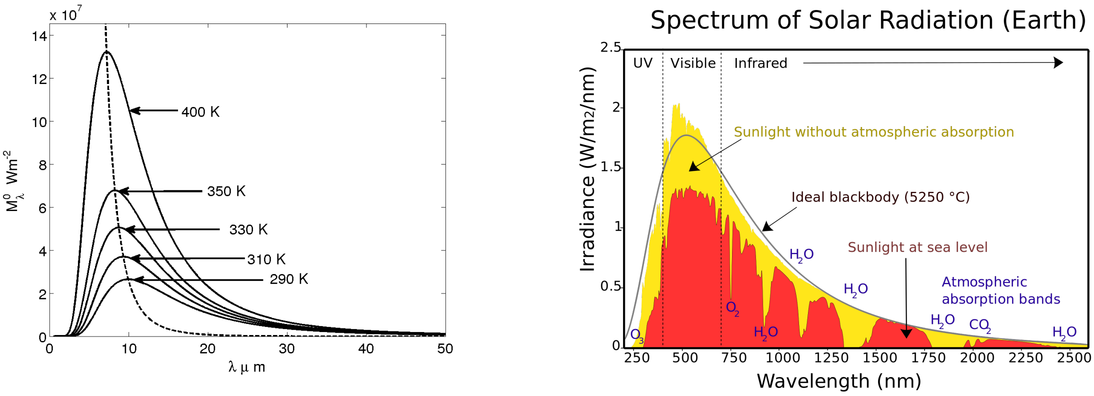

Figure 2. Left: Plank's law at various temperatures - Right: Energy spectrum of the atmosphere

By generalizing the Plank's law with the Stefan Boltzmann law (proposed first in 1879 and then in 1884 by Joseph Stefan and Ludwig Boltzmann), it is possible to address mathematically the energy spectrum of real body at each wavelength depending on the temperature, the optical condition and the real body properties, which is the base of the infrared thermography.

For example, Figure 2 right presents the energy spectrum of the atmosphere at various levels, it can be seen that the various properties of the atmosphere affect the spectrum at various wavelengths. Other important point is that the infrared solar heat flux can be approximated by a black body at 5523,15 K.

3.2.1.2. Infrared Thermography

The infrared thermography is a way to measure the thermal radiation received from a medium. With that information about the electromagnetic flux, it is possible to estimate the surface temperature of the body, see section 3.2.1.1 . Various types of detector can assure the measure of the electromagnetic radiation.

Those different detectors can take various forms and/or manufacturing process. For our research purposes, we use uncooled infrared camera using a matrix of microbolometers detectors. A microbolometer, as a lot of transducers, converts a radiation in electric current used to represent the physical quantity (here the heat flux).

This field of activity includes the use and the improvement of vision system, like in [3].

3.2.2. Heat transfer theory

Once the acquisition process is done, it is useful to model the heat conduction inside the cartesian domain Ω. Note that in opaque solid medium the heat conduction is the only mode of heat transfer. Proposed by Jean Baptiste Biot in 1804 and experimentally demonstrated by Joseph Fourier in 1821, the Fourier Law describes the heat flux inside a solid, cf Equation (22).

$$\varphi = k\nabla T \quad X \in \Omega \tag{19}$$

Where k is the thermal conductivity in W.m^{-1}.K ^{o}, ∇ is the gradient operator and φ is the heat flux density in Wm^{-2}. This law illustrates the first principle of thermodynamic (law of conservation of energy) and implies the second principle (irreversibility of the phenomenon). From this law it can be seen that the heat flux always goes from hot area to cold area.
An energy balance with respect to the first principle yields to the expression of the heat conduction in all point of the domain Ω, cf Equation (23). This equation has been proposed by Joseph Fourier in 1811.

$$\rho C \frac{\partial T(X,t)}{\partial t} = \nabla \cdot (k \nabla T) + P \quad X \in \Omega \tag{20}$$

With $\nabla.()$ the divergence operator, C the specific heat capacity in J.kg^{-1}.^{o}K^{-1}, ρ the volumetric mass density in kg. m^{-3}, X the space variable $X = \{x, y, z\}$ and P a possible internal heat production in W.m^{-3}.
To solve the system (23), it is necessary to express the boundaries conditions of the system. With the developments presented in section 3.2.1.1 and the Fourier's law, it is possible, for example, to express the thermal radiation and the convection phenomenon which can occur at $\partial\Omega$ the system boundaries, cf Equation (24).

$$\varphi = k\nabla T \cdot n = \underbrace{h\left(T_{fluid} - T_{Boundarie}\right)}_{\text{Convection}} + \underbrace{\epsilon\sigma_s\left(T^4_{environement} - T^4_{Boundary}\right)}_{\text{Radiation}} + \varphi_0 \quad X \in \partial\Omega \tag{21}$$

Equation (24) is the so called Robin condition on the boundary $\partial\Omega$, where n is the normal, h the convective heat transfer coefficient in W.m^{-2}.K^{-1} and φ_0 an external energy contribution W.m^{-2}, in cases where the external energy contribution is artificial and controlled we call it active thermography (spotlight etc...), otherwise it is called passive thermography (direct solar heat flux).
The systems presented in the different sections above (3.2.1 to 3.2.2) are useful to build physical models in order to represents the measured quantity. To estimate key parameters, as the conductivity, model inversion is used, the next section will introduce that principle.

3.2.3. Inverse model for parameters estimation

Lets take any model A which can for example represent the conductive heat transfer in a medium, the model is solved for a parameter vector P and it yields another vector b, cf Equation (25). For example if A represents the heat transfer, b can be the temperature evolution.

$$AP = b \tag{22}$$

With A a matrix of size $n \times m$, P a vector of size m and b of size n, preferentially $n >> P$. This model is called direct model, the inverse model consist to find a vector P which satisfy the results b of the direct model. For that we need to inverse the matrix A, cf Equation (26).

$$P = A^{-1}b \tag{23}$$

Here we want to find the solution AP which is closest to the acquired measures M, Equation (27).

$$AP \approx \mathcal{M} \tag{24}$$

To do that it is important to respect the well posed condition established by Jacques Hadamard in 1902

- A solution exists.
- The solution is unique.
- The solution's behavior changes continuously with the initial conditions.

Unfortunately those condition are rarely respected in our field of study. That is why we dont solve directly the system (27) but we minimise the quadratic coast function (28) which represents the Legendre-Gauss least square algorithm for linear problems.

$$min_P \left(\|AP - \mathcal{M}\|^2\right) = min_P \left(\mathcal{F}\right) \tag{25}$$

Where $\mathcal{F}$ can be a product of matrix.

$$\mathcal{F} = [AP - \mathcal{M}]^T [AP - \mathcal{M}]$$

In some cases the problem is still ill-posed and need to be regularized for example using the Tikhonov regularization. An elegant way to minimize the cost function $\mathcal{F}$ is compute the gradient, Equation (29) and find where it is equal to zero.

$$\nabla\mathcal{F}(P) = 2\left[-\frac{\partial AP^T}{\partial P}\right][AP - \mathcal{M}] = 2J(P)^T [AP - \mathcal{M}] \tag{26}$$

Where J is the sensitivity matrix of the model A with respect to the parameter vector P.
Until now the inverse method proposed is valid only when the model A is linearly dependent of its parameter P, for the heat equation it is the case when the external heat flux has to be estimated, φ_0 in Equation (24). For all the other parameters, like the conductivity k the model is non-linearly dependant of its parameter P. For such case the use of iterative algorithm is needed, for example the Levenberg-Marquardt algorithm, cf Equation (30).

$$P^{k+1} = P^k + \left[(J^k)^T J^k + \mu^k \Omega^k\right]^{-1} (J^k)^T \left[\mathcal{M} - A(P^k)\right] \tag{27}$$

Equation (30) is solved iteratively at each loop k. Some of our results with such linear or non linear method can be seen in [4] or [2], more specifically [1] is a custom implementation of the Levenberg-Marquardt algorithm based on the adjoint method (developed by Jacques Louis Lions in 1968) coupled to the conjugate gradient algorithm to estimate wide properties field in a medium.

3.3. Reflectometry-based methods for electrical engineering and for civil engineering

The fast development of electronic devices in modern engineering systems involves more and more connections through cables, and consequently, with an increasing number of connection failures. Wires and connectors are subject to ageing and degradation, sometimes under severe environmental conditions. In many applications, the reliability of electrical connexions is related to the quality of production or service, whereas in critical applications reliability becomes also a safety issue. It is thus important to design smart diagnosis systems able to detect connection defects in real time. This fact has motivated research projects on methods for fault diagnosis in this field. Some of these projects are based on techniques of reflectometry, which consist in injecting waves into a cable or a network and in analyzing the reflections. Depending on the injected waveforms and on the methods of analysis, various techniques of reflectometry are available. They all have the common advantage of being non destructive.

At Inria the research activities on reflectometry started within the SISYPHE EPI several years ago and now continue in the I4S EPI. Our most notable contribution in this area is a method based on the *inverse scattering* theory for the computation of *distributed characteristic impedance* along a cable from reflectometry measurements [14], [11], [56]. It provides an efficient solution for the diagnosis of *soft* faults in electrical cables, like in the example illustrated in Figure 3 . While most reflectometry methods for fault diagnosis are based on the detection and localization of impedance discontinuity, our method yielding the spatial profile of the characteristic impedance is particularly suitable for the diagnosis of soft faults *with no or weak impedance discontinuities*.

Fault diagnosis for wired networks have also been studied in Inria [58], [54]. The main results concern, on the one hand, simple star-shaped networks from measurements made at a single node, on the other hand, complex networks of arbitrary topological structure with complete node observations.

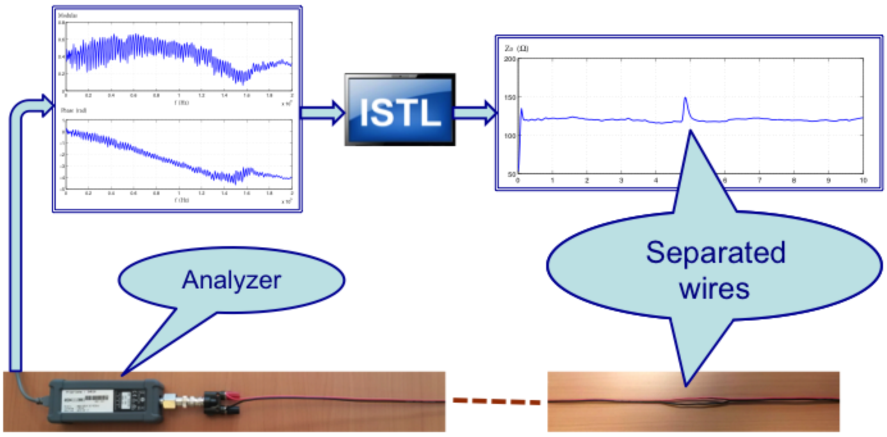

Figure 3. Inverse scattering software (ISTL) for cable soft fault diagnosis.

Though initially our studies on reflectometry were aiming at applications in electrical engineering, since the creation of the I4S team, we are also investigating applications in the field of civil engineering, by using electrical cables as sensors for monitoring changes in mechanical structures.

What follows is about some basic elements on mathematical equations of electric cables and networks, the main approach we follow in our study, and our future research directions.

3.3.1. Mathematical model of electric cables and networks

A cable excited by a signal generator can be characterized by the telegrapher's equations [55]

$$\begin{aligned} &\frac{\partial}{\partial z}V(t,z) + L(z)\frac{\partial}{\partial t}I(t,z) + R(z)I(t,z) = 0 \\ &\frac{\partial}{\partial z}I(t,z) + C(z)\frac{\partial}{\partial t}V(t,z) + G(z)V(t,z) = 0 \end{aligned} \tag{28}$$

where t represents the time, z is the longitudinal coordinate along the cable, $V(t,z)$ and $I(t,z)$ are respectively the voltage and the current in the cable at the time instant t and at the position z, $R(z), L(z), C(z)$ and $G(z)$ denote respectively the series resistance, the inductance, the capacitance and the shunt conductance per unit length of the cable at the position z. The left end of the cable (corresponding to $z = a$) is connected to a voltage source $V_s(t)$ with internal impedance R_s. The quantities $V_s(t)$, R_s, $V(t,a)$ and $I(t,a)$ are related by

$$V(t,a) = V_s(t) - R_s I(t,a). \tag{29}$$

At the right end of the cable (corresponding to $z = b$), the cable is connected to a load of impedance R_L, such that

$$V(t,b) \quad = R_L I(t,b). \tag{30}$$

One way for deriving the above model is to spatially discretize the cable and to characterize each small segment with 4 basic lumped parameter elements for the j-th segment: a resistance ΔR_j, an inductance ΔL_j, a capacitance ΔC_j and a conductance ΔG_j. The entire circuit is described by a system of ordinary differential equations. When the spatial discretization step size tends to zero, the limiting model leads to the telegrapher's equations.

A wired network is a set of cables connected at some nodes, where loads and sources can also be connected. Within each cable the current and voltage satisfy the telegrapher's equations, whereas at each node the current and voltage satisfy the Kirchhoff's laws, unless in case of connector failures.

3.3.2. *The inverse scattering theory applied to cables*

The inverse scattering transform was developed during the 1970s-1980s for the analysis of some nonlinear partial differential equations [53]. The visionary idea of applying this theory to solving the cable inverse problem goes also back to the 1980s [52]. After having completed some theoretic results directly linked to practice [14], [56], we started to successfully apply the inverse scattering theory to cable soft fault diagnosis, in collaboration with GEEPS-SUPELEC [11].

To link electric cables to the inverse scattering theory, the telegrapher's equations are transformed in a few steps to fit into a particular form studied in the inverse scattering theory. The Fourier transform is first applied to obtain a frequency domain model, the spatial coordinate z is then replaced by the propagation time

$$x(z) = \int_0^z \sqrt{L(s)C(s)}ds$$

and the frequency domain variables $V(\omega,x), I(\omega,x)$ are replaced by the pair

$$\begin{aligned} \nu_1(\omega,x) &= \frac{1}{2}\left[Z_0^{-\frac{1}{2}}(x)U(\omega,x) - Z_0^{\frac{1}{2}}(x)I(\omega,x)\right] \\ \nu_2(\omega,x) &= \frac{1}{2}\left[Z_0^{-\frac{1}{2}}(x)U(\omega,x) + Z_0^{\frac{1}{2}}(x)I(\omega,x)\right] \end{aligned} \tag{31}$$

with

$$Z_0(x) = \sqrt{\frac{L(x)}{C(x)}}. \tag{32}$$

These transformations lead to the Zakharov-Shabat equations

$$\begin{aligned} \frac{d\nu_1(\omega,x)}{dx} + ik\nu_1(\omega,x) &= q^*(x)\nu_1(\omega,x) + q^+(x)\nu_2(\omega,x) \\ \frac{d\nu_2(\omega,x)}{dx} - ik\nu_2(\omega,x) &= q^-(x)\nu_1(\omega,x) - q^*(x)\nu_2(\omega,x) \end{aligned} \tag{33}$$

with

$$\begin{aligned} q^{\pm}(x) &= -\frac{1}{4}\frac{d}{dx}\left[ln\frac{L(x)}{C(x)}\right] \mp \frac{1}{2}\left[\frac{R(x)}{L(x)} - \frac{G(x)}{C(x)}\right] \\ &= -\frac{1}{2Z_0(x)}\frac{d}{dx}Z_0(x) \mp \frac{1}{2}\left[\frac{R(x)}{L(x)} - \frac{G(x)}{C(x)}\right] \\ q^{*}(x) &= \frac{1}{2}\left[\frac{R(x)}{L(x)} + \frac{G(x)}{C(x)}\right]. \end{aligned} \tag{34}$$

These equations have been well studied in the inverse scattering theory, for the purpose of determining partly the "potential functions" $q^{\pm}(x)$ and $q^{*}(x)$ from the scattering data matrix, which turns out to correspond to the data typically collected with reflectometry instruments. For instance, it is possible to compute the function $Z_0(x)$ defined in (35), often known as the characteristic impedance, from the reflection coefficient measured at one end of the cable. Such an example is illustrated in Figure 3 . Any fault affecting the characteristic impedance, like in the example of Figure 3 caused by a slight geometric deformation, can thus be efficiently detected, localized and characterized.

3.4. Research Program

The research will first focus on the extension and implementation of current techniques as developed in I4S and IFSTTAR. Before doing any temperature rejection on large scale structures as planned, we need to develop good and accurate models of thermal fields. We also need to develop robust and efficient versions of our algorithms, mainly the subspace algorithms before envisioning linking them with physical models. Briefly, we need to mature our statistical toolset as well as our physical modeling before mixing them together later on.

3.4.1. Vibration analysis and monitoring

3.4.1.1. Direct vibration modeling under temperature changes

This task builds upon what has been achieved in the CONSTRUCTIF project, where a simple formulation of the temperature effect has been exhibited, based on relatively simple assumptions. The next step is to generalize this modeling to a realistic large structure under complex thermal changes. Practically, temperature and resulting structural prestress and pre strains of thermal origin are not uniform and civil structures are complex. This leads to a fully 3D temperature field, not just a single value. Inertia effects also forbid a trivial prediction of the temperature based on current sensor outputs while ignoring past data. On the other side, the temperature is seen as a nuisance. That implies that any damage detection procedure has first to correct the temperature effect prior to any detection.

Modeling vibrations of structures under thermal prestress does and will play an important role in the static correction of kinematic measurements, in health monitoring methods based on vibration analysis as well as in durability and in the active or semi-active control of civil structures that by nature are operated under changing environmental conditions. As a matter of fact, using temperature and dynamic models the project aims at correcting the current vibration state from induced temperature effects, such that damage detection algorithms rely on a comparison of this thermally corrected current vibration state with a reference state computed or measured at a reference temperature. This approach is expected to cure damage detection algorithms from the environmental variations.

I4S will explore various ways of implementing this concept, notably within the FUI SIPRIS project.

3.4.1.2. Damage localization algorithms (in the case of localized damages such as cracks)

During the CONSTRUCTIF project, both feasibility and efficiency of some damage detection and localization algorithms were proved. Those methods are based on the tight coupling of statistical algorithms with finite element models. It has been shown that effective localization of some damaged elements was possible, and this was validated on a numerical simulated bridge deck model. Still, this approach has to be validated on real structures.

On the other side, new localization algorithms are currently investigated such as the one developed conjointly with University of Boston and tested within the framework of FP7 ISMS project. These algorithms will be implemented and tested on the PEGASE platform as well as all our toolset.

When possible, link with temperature rejection will be done along the lines of what has been achieved in the CONSTRUCTIF project.

3.4.1.3. Uncertainty quantification for system identification algorithms

Some emphasis will be put on expressing confidence intervals for system identification. It is a primary goal to take into account the uncertainty within the identification procedure, using either identification algorithms derivations or damage detection principles. Such algorithms are critical for both civil and aeronautical structures monitoring. It has been shown that confidence intervals for estimation parameters can theoretically be related to the damage detection techniques and should be computed as a function of the Fisher information matrix associated to the damage detection test. Based on those assumptions, it should be possible to obtain confidence intervals for a large class of estimates, from damping to finite elements models. Uncertainty considerations are also deeply investigated in collaboration with Dassault Aviation in Mellinger PhD thesis or with Northeastern University, Boston, within Gallegos PhD thesis.

3.4.2. Reflectometry-based methods for civil engineering structure health monitoring

The inverse scattering method we developed is efficient for the diagnosis of all soft faults affecting the characteristic impedance, the major parameter of a cable. In some particular applications, however, faults would rather affect the series resistance (ohmic loss) or shunt conductance (leakage loss) than the characteristic impedance. The first method we developed for the diagnosis of such losses had some numerical stability problems. The new method is much more reliable and efficient. It is also important to develop efficient solutions for long cables, up to a few kilometers.

For wired networks, the methods we already developed cover either the case of simple networks with a single node measurement or the case of complex networks with complete node measurements. Further developments are still necessary for intermediate situations.

In terms of applications, the use of electric cables as sensors for the monitoring of various structures is still at its beginning. We believe that this new technology has a strong potential in different fields, notably in civil engineering and in materials engineering.

3.4.3. Non Destructive testing of CFRP bonded on concrete through active thermography

Strengthening or retrofitting of reinforced concrete structures by externally bonded fiber-reinforced polymer (FRP) systems is now a commonly accepted and widespread technique. However, the use of bonding techniques always implies following rigorous installation procedures. The number of carbon fiber-reinforced polymer (CFRP) sheets and the glue layer thickness are designed by civil engineers to address strengthening objectives. Moreover, professional crews have to be trained accordingly in order to ensure the durability and long-term performance of the FRP reinforcements. Conformity checking through an ‘in situ’ verification of the bonded FRP systems is then highly desirable. The quality control programme should involve a set of adequate inspections and tests. Visual inspection and acoustic sounding (hammer tap) are commonly used to detect delaminations (disbonds). Nevertheless, these techniques are unable to provide sufficient information about the depth (in case of multilayered composite) and width of the disbonded areas. They are also incapable of evaluating the degree of adhesion between the FRP and the substrate (partial delamination, damage of the resin and poor mechanical properties of the resin). Consequently, rapid and efficient inspection methods are required. Among the non-destructive (NDT) methods currently under study, active infrared thermography is investigated due to its ability to be used in the field. In such context and to reach the aim of having an in situ efficient NDT method, we carried out experiments and subsequent data analysis using thermal excitation. Image processing, inverse thermal modelling and 3D numerical simulations are used and then applied to experimental data obtained in laboratory conditions.

3.4.4. IRSHM: Multi-Sensing system for outdoor thermal monitoring

Ageing of transport infrastructures combined with traffic and climatic solicitations contribute to the reduction of their performances. To address and quantify the resilience of civil engineering structure, investigations on robust, fast and efficient methods are required. Among research works carried out at IFSTTAR, methods for long term monitoring face an increasing demand. Such works take benefits of this last decade technological progresses in ICT domain.

Thanks to IFSTTAR years of experience in large scale civil engineering experiment, I4S is able to perform very long term thermal monitoring of structures exposed to environmental condition, as the solar heat flux, natural convection or seasonal perturbation. Informations system are developed to asses the data acquisition and researchers work on the quantification of the data to detect flaws emergence on structure, those techniques are also used to diagnose thermal insulation of buildings or monitoring of guided transport infrastructures, Figure 4 left. Experiments are carried out on a real transport infrastructure open to traffic and buildings. The detection of the inner structure of the deck is achieved by image processing techniques (as FFT), principal component thermography (PCT), Figure 4 right, or characterization of the inner structure thanks to an original image processing approach.

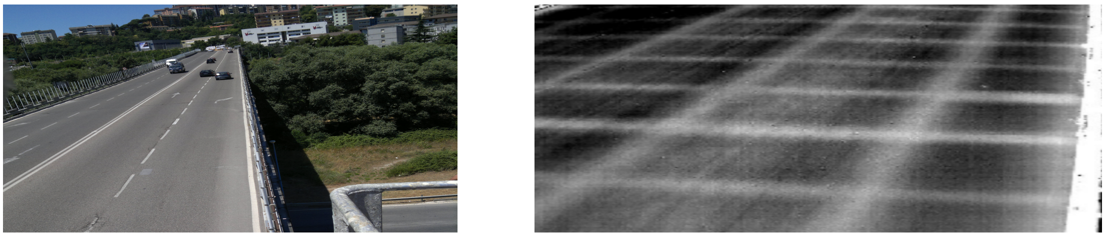

Figure 4. Left: Image in the visible spectrum of the deck surface - Right: PCT result on a bridge deck

For the next few years, I4S is actively implied in the SenseCity EQUIPEX (http://sense-city.ifsttar.fr/) where our informations systems are used to monitor a mini-city replica, Figure 5 .

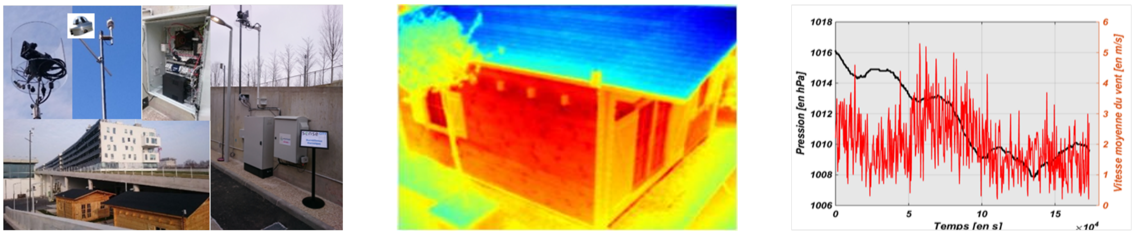

Figure 5. Various view and results of the SenseCity experimentation site - (site and hardware view, IR imaging, Environmental Monitoring)

3.4.5. R5G: The 5th Generation Road

The road has to reinvent itself periodically in response to innovations, societal issues and rising user expectations. The 5th Generation Road (R5G) focuses firmly on the future and sets out to be automated, safe, sustainable and suited to travel needs. Several research teams are involved in work related to this flagship project for IFSTTAR, which is a stakeholder in the Forever Open Road. Through its partnership with the COSYS (IFSTTAR) department, I4S is fully involved in the development of the 5th Generation Road.

Most of the innovations featured in R5G are now mature, for example communication and few solutions for energy exchange between the infrastructure, the vehicle and the network manager; recyclable materials with the potential for self-diagnosis and repair, a pavement surface that remains permanently optimal irrespective of climatic variations… Nevertheless, implementing them on an industrial scale at a reasonable cost still represents a real challenge. Consultation with the stakeholders (researchers, industry, road network owners and users) has already established the priorities for the creation of full-scale demonstrators. The next stages are to achieve synergy between the technologies tested by the demonstrators, to manage the interfaces and get society to adopt R5G.

MINGUS Project-Team

3. Research Program

3.1. Research Program

The MINGUS project is devoted to the mathematical and numerical analysis of models arising in plasma physics and nanotechnology. The main goal is to construct and analyze numerical methods for the approximation of PDEs containing multiscale phenomena. Specific multiscale numerical schemes will be proposed and analyzed in different regimes (namely highly-oscillatory and dissipative). The ultimate goal is to dissociate the physical parameters (generically denoted by ε) from the numerical parameters (generically denoted by h) with a uniform accuracy. Such a task requires mathematical prerequisite of the PDEs.

Then, for a given stiff (highly-oscillatory or dissipative) PDE, the methodology of the MINGUS team will be the following

- Mathematical study of the asymptotic behavior of multiscale models.
  This part involves averaging and asymptotic analysis theory to derive asymptotic models, but also long-time behavior of the considered models.
- Construction and analysis of multiscale numerical schemes.
  This part is the core of the project and will be deeply inspired for the mathematical prerequisite. In particular, our ultimate goal is the design of *Uniformly Accurate* (UA) schemes, whose accuracy is independent of ε.
- Validation on physically relevant problems.
  The last goal of the MINGUS project is to validate the new numerical methods, not only on toy problems, but also on realistic models arising in physics of plasmas and nanotechnologies. We will benefit from the Selalib software library which will help us to scale-up our new numerical methods to complex physics.

3.1.1. Dissipative problems

In the dissipative context, the asymptotic analysis is quite well understood in the deterministic case and multiscale numerical methods have been developed in the last decades. Indeed, the so-called Asymptotic-Preserving schemes has retained a lot of attention all over the world, in particular in the context of collisional kinetic equations. But, there is still a lot of work to do if one is interesting in the derivation high order asymptotic models, which enable to capture the original solution for all time. Moreover, this analysis is still misunderstood when more complex systems are considered, involving non homogeneous relaxation rates or stochastic terms for instance. Following the methodology we aim at using, we first address the mathematical analysis before deriving multiscale efficient numerical methods.

A simple model of dissipative systems is governed by the following differential equations

$$\left\{\begin{array}{lll} \dfrac{dx^\varepsilon(t)}{dt} & = \mathcal{G}(x^\varepsilon(t), y^\varepsilon(t)), & x^\varepsilon(0) = x_0, \\[2ex] \dfrac{dy^\varepsilon(t)}{dt} & = -\dfrac{y^\varepsilon(t)}{\varepsilon} + \mathcal{H}(x^\varepsilon(t), y^\varepsilon(t)), & y^\varepsilon(0) = y_0, \end{array}\right. \tag{35}$$

for given initial condition $(x_0, y_0) \in \mathbb{R}^2$ and given smooth functions $\mathcal{G}, \mathcal{H}$ which possibly involve stochastic terms.

3.1.1.1. Asymptotic analysis of dissipative PDEs (F. Castella, P. Chartier, A. Debussche, E. Faou, M. Lemou)

Derivation of asymptotic problems

Our main goal is to analyze the asymptotic behavior of dissipative systems of the form (3) when ε goes to zero. The *center manifold theorem* [35] is of great interest but is largely unsatisfactory from the following points of view

- a constructive approach of h and x_0^ε is clearly important to identify the high-order asymptotic models: this would require expansions of the solution by means of B-series or word-series [37] allowing the derivation of error estimates between the original solution and the asymptotic one.
- a better approximation of the transient phase is strongly required to capture the solution for small time: extending the tools developed in averaging theory, the main goal is to construct a suitable change of variable which enables to approximate the original solution for all time.

Obviously, even at the ODE level, a deep mathematical analysis has to be performed to understand the asymptotic behavior of the solution of (3). But, the same questions arise at the PDE level. Indeed, one certainly expects that dissipative terms occurring in collisional kinetic equations (2) may be treated theoretically along this perspective. The key new point indeed is to see the center manifold theorem as a change of variable in the space on unknowns, while the standard point of view leads to considering the center manifold as an asymptotic object.

Stochastic PDEs
We aim at analyzing the asymptotic behavior of stochastic collisional kinetic problems, that is equation of the type (2). The noise can describe creation or absorption (as in (2)), but it may also be a forcing term or a random magnetic field. In the parabolic scaling, one expect to obtain parabolic SPDEs at the limit. More precisely, we want to understand the fluid limits of kinetic equations in the presence of noise. The noise is smooth and non delta correlated. It contains also a small parameter and after rescaling converges formally to white noise. Thus, this adds another scale in the multiscale analysis. Following the pioneering work [38], some substantial progresses have been done in this topic.

More realistic problems may be addressed such as high field limit describing sprays, or even hydrodynamic limit. The full Boltzmann equation is a very long term project and we wish to address simpler problems such as convergences of BGK models to a stochastic Stokes equation.

The main difficulty is that when the noise acts as a forcing term, which is a physically relevant situation, the equilibria are affected by the noise and we face difficulties similar to that of high field limit problems. Also, a good theory of averaging lemma in the presence of noise is lacking. The methods we use are generalization of the perturbed test function method to the infinite dimensional setting. We work at the level of the generator of the infinite dimensional process and prove convergence in the sense of the martingale problems. A further step is to analyse the speed of convergence. This is a prerequisite if one wants to design efficient schemes. This requires more refined tools and a good understanding of the Kolmogorov equation.

3.1.1.2. Numerical schemes for dissipative problems (All members)

The design of numerical schemes able to reproduce the transition from the microscopic to macroscopic scales largely matured with the emergence of the Asymptotic Preserving schemes which have been developed initially for collisional kinetic equations (actually, for solving (2) when $\beta \to 0$). Several techniques have flourished in the last decades. As said before, AP schemes entail limitations which we aim at overcoming by deriving

- AP numerical schemes whose numerical cost diminishes as $\beta \to 0$,
- Uniformly accurate numerical schemes, whose accuracy is independent of β.

Time diminishing methods
The main goal consists in merging Monte-Carlo techniques [33] with AP methods for handling *automatically* multiscale phenomena. As a result, we expect that the cost of the so-obtained method decreases when the asymptotic regime is approached; indeed, in the collisional (i.e. dissipative) regime, the deviational part becomes negligible so that a very few number of particles will be generated to sample it. A work in this direction has been done by members of the team.

We propose to build up a method which permits to realize the transition from the microscopic to the macroscopic description without domain decomposition strategies which normally oblige to fix and tune an interface in the physical space and some threshold parameters. Since it will permit to go over domain decomposition and AP techniques, this approach is a very promising research direction in the numerical approximation of multiscale kinetic problems arising in physics and engineering.

Uniformly accurate methods
To overcome the accuracy reduction observed in AP schemes for intermediate regimes, we intend to construct and analyse multiscale numerical schemes for (3) whose error is uniform with respect to ε. The construction of such a scheme requires a deep mathematical analysis as described above. Ideally one would like to develop schemes that preserve the center manifold (without computing the latter!) as well as schemes that resolve numerically the stiffness induced by the fast convergence to equilibrium (the so-called transient phase). First, our goal is to extend the strategy inspired by the central manifold theorem in the ODE case to the PDE context, in particular for collisional kinetic equations (2) when $\beta \to 0$. The design of Uniformly Accurate numerical schemes in this context would require to generalize two-scale techniques introduced in the framework of highly-oscillatory problems [36].

Multiscale numerical methods for stochastic PDEs
AP schemes have been developed recently for kinetic equations with noise in the context of Uncertainty Quantification UQ [41]. These two aspects (multiscale and UQ) are two domains which usually come within the competency of separate communities. UQ has drawn a lot of attention recently to control the propagation of data pollution; undoubtedly UQ has a lot of applications and one of our goals will be to study how sources of uncertainty are amplified or not by the multiscale character of the model. We also wish to go much further and by developing AP schemes when the noise is also rescaled and the limit is a white noise driven SPDE, as described in section (3.1.1.1). For simple nonlinear problem, this should not present much difficulties but new ideas will definitely be necessary for more complicated problems when noise deeply changes the asymptotic equation.

3.1.2. Highly-oscillatory problems

As a generic model for highly-oscillatory systems, we will consider the equation

$$\frac{du^\varepsilon(t)}{dt} = \mathcal{F}(t/\varepsilon, u^\varepsilon(t)), \quad u^\varepsilon(0) = u_0, \tag{36}$$

for a given u_0 and a given periodic function $\mathcal{F}$ (of period P w.r.t. its first variable) which possibly involves stochastic terms. Solution u^ε exhibits high-oscillations in time superimposed to a slow dynamics. Asymptotic techniques -resorting in the present context to *averaging* theory [45]- allow to decompose

$$u^\varepsilon(t) = \Phi_{t/\varepsilon} \circ \Psi_t \circ \Phi_0^{-1}(u_0), \tag{37}$$

into a fast solution component, the εP-periodic change of variable $\Phi_{t/\varepsilon}$, and a slow component, the flow Ψ_t of a non-stiff *averaged* differential equation. Although equation (5) can be satisfied only up to a small remainder, various methods have been recently introduced in situations where (4) is posed in $\mathbb{R}^n$ or for the Schrödinger equation (1).

In the asymptotic behavior $\varepsilon \to 0$, it can be advantageous to replace the original singularly perturbed model (for instance (1) or (2)) by an approximate model which does not contain stiffness any longer. Such reduced models can be derived using asymptotic analysis, namely averaging methods in the case of highly-oscillatory problems. In this project, we also plan to go beyond the mere derivation of limit models, by searching for better approximations of the original problem. This step is of mathematical interest *per se* but it also paves the way of the construction of multiscale numerical methods.

3.1.2.1. Asymptotic analysis of highly-oscillatory PDEs (All members)

Derivation of asymptotic problems

We intend to study the asymptotic behavior of highly-oscillatory evolution equations of the form (4) posed in an infinite dimensional Banach space.

Recently, the stroboscopic averaging has been extended to the PDE context, considering nonlinear Schrödinger equation (1) in the highly-oscillatory regime. A very exciting way would be to use this averaging strategy for highly-oscillatory kinetic problem (2) as those encountered in strongly magnetized plasmas. This turns out to be a very promising way to re-derive gyrokinetic models which are the basis of tokamak simulations in the physicists community. In contract with models derived in the literature (see [34]) which only capture the average with respect to the oscillations, this strategy allows for the complete recovery of the exact solution from the asymptotic (non stiff) model. This can be done by solving companion transport equation that stems naturally from the decomposition (5).

Long-time behavior of Hamiltonian systems
The study of long-time behavior of nonlinear Hamiltonian systems have received a lot of interest during the last decades. It enables to put in light some characteristic phenomena in complex situations, which are beyond the reach of numerical simulations. This kind of analysis is of great interest since it can provide very precise properties of the solution. In particular, we will focus on the dynamics of nonlinear PDEs when the initial condition is close to a stationary solution. Then, the long-time behavior of the solution is studied through mainly three axis

- *linear stability*: considering the linearized PDE, do we have stability of a stationary solution ? Do we have linear Landau damping around stable non homogeneous stationary states?
- *nonlinear stability*: under a criteria, do we have stability of a stationary solution in energy norm like in [42], and does this stability persist under numerical discretization? For example one of our goals is to address the question of the existence and stability of discrete travelling wave in space and time.
- do we have existence of damped solutions for the full nonlinear problem ? Around homogeneous stationary states, solutions scatter towards a modified stationary state (see [43], [39]). The question of existence of Landau damping effects around non homogeneous states is still open and is one of our main goal in the next future.

Asymptotic behavior of stochastic PDEs
The study of SPDEs has known a growing interest recently, in particular with the fields medal of M. Hairer in 2014. In many applications such as radiative transfer, molecular dynamics or simulation of optical fibers, part of the physical interactions are naturally modeled by adding supplementary random terms (the noise) to the initial deterministic equations. From the mathematical point of view, such terms change drastically the behavior of the system.

- In the presence of noise, highly-oscillatory dispersive equations presents new problems. In particular, to study stochastic averaging of the solution, the analysis of the long time behavior of stochastic dispersive equations is required, which is known to be a difficult problem in the general case. In some cases (for instance highly-oscillatory Schrödinger equation (1) with a time white noise in the regime $\varepsilon \ll 1$), it is however possible to perform the analysis and to obtain averaged stochastic equations. We plan to go further by considering more difficult problems, such as the convergence of a stochastic Klein-Gordon-Zakharov system to as stochastic nonlinear Schrödinger equation.
- The long-time behavior of stochastic Schrödinger equations is of great interest to analyze mathematically the validity of the Zakharov theory for wave turbulence (see [44]). The problem of wave turbulence can be viewed either as a deterministic Hamiltonian PDE with random initial data or a randomly forced PDEs where the stochastic forcing is concentrated in some part of the spectrum (in this sense it is expected to be a hypoelliptic problem). One of our goals is to test the validity the Zakharov equation, or at least to make rigorous the spectrum distribution spreading observed in the numerical experiments.

3.1.2.2. Numerical schemes for highly-oscillatory problems (All members)

This section proposes to explore numerical issues raised by highly-oscillatory nonlinear PDEs for which (4) is a prototype. Simulating a highly-oscillatory phenomenon usually requires to adapt the numerical parameters in order to solve the period of size ε so as to accurately simulate the solution over each period, resulting in a unacceptable execution cost. Then, it is highly desirable to derive numerical schemes able to advance the solution by a time step independent of ε. To do so, our goal is to construct *Uniformly Accurate* (UA) numerical schemes, for which the numerical error can be estimated by Ch^p (h being any numerical parameters) with C independent of ε and p the order of the numerical scheme.

Recently, such numerical methods have been proposed by members of the team in the highly-oscillatory context [36]. They are mainly based on a separation of the fast and slow variables, as suggested by the decomposition (5). An additional ingredient to prove the uniformly accuracy of the method for (4) relies on the search for an appropriate initial data which enables to make the problem smooth with respect to ε.

Such an approach is assuredly powerful since it provides a numerical method which enables to capture the high oscillations in time of the solution (and not only its average) even with a large time step. Moreover, in the asymptotic regime, the potential gain is of order $1/\varepsilon$ in comparison with standard methods, and finally averaged models are not able to capture the intermediate regime since they miss important information of the original problem. We are strongly convinced that this strategy should be further studied and extended to cope with some other problems. The ultimate goal being to construct a scheme for the original equation which degenerates automatically into a consistent approximation of the averaged model, without resolving it, the latter can be very difficult to solve.

- Space oscillations:
  When rapidly oscillating coefficients in **space** (*i.e.* terms of the form $a(x, x/\varepsilon)$) occur in elliptic or parabolic equations, homogenization theory and numerical homogenization are usually employed to handle the stiffness. However, these strategies are in general not accurate for all $\varepsilon \in]0, 1]$. Then, the construction of numerical schemes which are able to handle both regimes in an uniform way is of great interest. Separating fast and slow *spatial* scales merits to be explored in this context. The delicate issue is then to extend the choice suitable initial condition to an *appropriate choice of boundary conditions* of the augmented problem.
- Space-time oscillations:
  For more complex problems however, the recent proposed approaches fail since the main oscillations cannot be identified explicitly. This is the case for instance when the magnetic field B depends on t or x in (2) but also for many other physical problems. We then have to deal with the delicate issue of space-time oscillations, which is known to be a very difficult problem from a mathematical and a numerical point of view. To take into account the space-time mixing, a periodic motion has to be detected together with a phase S which possibly depends on the time and space variables. These techniques originate from **geometric optics** which is a very popular technique to handle highly-frequency waves.
- Geometrical properties:
  The questions related to the geometric aspects of multiscale numerical schemes are of crucial importance, in particular when long-time simulations are addressed (see [40]). Indeed, one of the main questions of geometric integration is whether intrinsic properties of the solution may be passed onto its numerical approximation. For instance, if the model under study is Hamiltonian, then the exact flow is symplectic, which motivates the design of symplectic numerical approximation. For practical simulations of Hamiltonian systems, symplectic methods are known to possess very nice properties (see [40]). It is important to combine multiscale techniques to geometric numerical integration. All the problems and equations we intend to deal with will be addressed with a view to preserve intrinsic geometric properties of the exact solutions and/or to approach the asymptotic limit of the system in presence of a small parameter. An example of a numerical method developed by members of the team is the multi-revolution method.
- Quasi-periodic case:

So far, numerical methods have been proposed for the periodic case with single frequency. However, the quasi-periodic case [0] is still misunderstood although many complex problems involve multi-frequencies. Even if the quasi-periodic averaging is doable from a theoretical point of view in the ODE case (see [45]), it is unclear how it can be extended to PDEs. One of the main obstacle being the requirement, usual for ODEs like (4), for $\mathcal{F}$ to be analytic in the periodic variables, an assumption which is clearly impossible to meet in the PDE setting. An even more challenging problem is then the design of numerical methods for this problem.

- extension to stochastic PDEs:
  All these questions will be revisited within the stochastic context. The mathematical study opens the way to the derivation of efficient multiscale numerical schemes for this kind of problems. We believe that the theory is now sufficiently well understood to address the derivation and numerical analysis of multiscale numerical schemes. Multi-revolution composition methods have been recently extended to highly-oscillatory stochastic differential equations The generalization of such multiscale numerical methods to SPDEs is of great interest. The analysis and simulation of numerical schemes for highly-oscillatory nonlinear stochastic Schrödinger equation under diffusion-approximation for instance will be one important objective for us. Finally, an important aspect concerns the quantification of uncertainties in highly-oscillatory kinetic or quantum models (due to an incomplete knowledge of coefficients or imprecise measurement of datas). The construction of efficient multiscale numerical methods which can handle multiple scales as well as random inputs have important engineering applications (see (4.1)).

[0] replacing t/ε by $t\omega/\varepsilon$ in (4), with $\omega \in \mathbb{R}^d$ a vector of non-resonant frequencies

SIMSMART Team

3. Research Program

3.1. Research Program

Introduction. Computer simulation of physical systems is becoming increasingly reliant on highly complex models, as the constant surge of computational power is nurturing scientists into simulating the most detailed features of reality – from complex molecular systems to climate/weather forecast.

Yet, when modeling physical reality, bottom-up approaches are stumbling over intrinsic difficulties. First, the timescale separation between the fastest simulated microscopic features, and the macroscopic effective slow behavior becomes huge, implying that the fully detailed and direct long time simulation of many interesting systems (*e.g.* large molecular systems) are out of reasonable computational reach. Second, the chaotic dynamical behaviors of the systems at stake, coupled with such multi-scale structures, exacerbate the intricate uncertainty of outcomes, which become highly dependent on intrinsic chaos, uncontrolled modeling, as well as numerical discretization. Finally, the massive increase of observational data addresses new challenges to classical data assimilation, such as dealing with high dimensional observations and/or extremely long time series of observations.

SIMSMART Identity. Within this highly challenging applicative context, SIMSMART positions itself as a computational probability and statistics research team, with a mathematical perspective. Our approach is based on the use of *stochastic modeling* of complex physical systems, and on the use of *Monte Carlo simulation* methods, with a strong emphasis on dynamical models. The two main numerical tasks of interest to SIMSMART are the following: (i) simulating with pseudo-random number generators - a.k.a. *sampling* - dynamical models of random physical systems, (ii) sampling such random physical dynamical models given some real observations - a.k.a. *Bayesian data assimilation*. SIMSMART aims at providing an appropriate mathematical level of abstraction and generalization to a wide variety of Monte Carlo simulation algorithms in order to propose non-superficial answers to both *methodological and mathematical* challenges. The issues to be resolved include computational complexity reduction, statistical variance reduction, and uncertainty quantification.

SIMSMART's Objectives. The main objective of SIMSMART is to disrupt this now classical field of particle Monte Carlo simulation by creating deeper mathematical frameworks adapted to the challenging world of complex (*e.g.* high dimensional and/or multi-scale), and massively observed systems, as described in the beginning of this introduction.

To be more specific, we will classify SIMSMART objectives using the following four intertwined topics:

1. Objective 1: Rare events and random simulation.
2. Objective 2: High dimensional and advanced particle filtering.
3. Objective 3: Non-parametric approaches.
4. Objective 4: Model reduction and sparsity.

Rare events Objective 1 are ubiquitous in random simulation, either to accelerate the occurrence of physically relevant random slow phenomenons, or to estimate the effect of uncertain variables. Objective 1 will be mainly concerned with particle methods where *splitting* is used to enforce the occurrence of rare events.

The problem of high dimensional observations, the main topic in Objective 2, is a known bottleneck in filtering, especially in non-linear particle filtering, where linear data assimilation methods remain the state-of-the-art approaches.

The increasing size of recorded observational data and the increasing complexity of models also suggest to devote more effort into non-parametric data assimilation methods, the main issue of Objective 3.

In some contexts, for instance when one wants to compare solutions of a complex (*e.g.* high dimensional) dynamical systems depending on uncertain parameters, the construction of relevant reduced-order models becomes a key topic. This is the content of Objective 4.

With respect to volume of research activity, Objective 1, Objective 4 and the sum (Objective 2+Objective 3) are comparable.

Some new challenges in the simulation and data assimilation of random physical dynamical systems have become prominent in the last decade. A first issue (i) consists in the intertwined problems of simulating on large, macroscopic random times, and simulating *rare events*. The link between both aspects stems from the fact that many effective, large times dynamics can be approximated by sequences of rare events. A second, obvious, issue (ii) consists in managing *very abundant observational data*. A third issue (iii) consists in quantifying *uncertainty/sensitivity/variance* of outcomes with respect to models or noise. A fourth issue (iv) consists in managing *high dimensionality*, either when dealing with complex prior physical models, or with very large data sets. The related increase of complexity also requires, as a fifth issue (v), the construction of *reduced models* to speed-up comparative simulations. In a context of very abundant data, this may be replaced by a sixth issue (vi) where complexity constraints on modeling is replaced by the use of *non-parametric statistical inference*.

Hindsight suggests that all the latter challenges are related. Indeed, the contemporary digital condition, made of a massive increase in computational power and in available data, is resulting in a demand for more complex and uncertain models, for more extreme regimes, and for using inductive approaches relying on abundant data.

For simplicity, we have classified SIMSMART research into the following already mentioned four main objectives.

1. Objective 1: Rare events and random simulation, which mainly encompass item (i).
2. Objective 2: High dimension and advanced particle filtering, which encompass item (iv).
3. Objective 3: Non-parametric inference, which mainly encompass item (ii) and (vi).
4. Objective 4: Model reduction, which mainly encompasses item (vi).

Uncertainty quantification (item (iii)) in fact underlies each aspect since we are mainly interested in Monte Carlo approaches, so that uncertainty can be *modeled by an initial random variable and be incorporated in the state space of the physical model.*

DYLISS Project-Team

3. Research Program

3.1. Computer science – symbolic artificial intelligence

We develop methods that use an explicit representation of the relationships between heterogeneous data and knowledge in order to construct a space of hypotheses. Therefore, our objectives in computer science is mainly to develop accurate representations (oriented graphs, Boolean networks, automata, or expressive grammars) to iteratively capture the complexity of a biological system.

Integrating data with querying languages: Semantic web for life sciences The first level of complexity in the data integration process consists in confronting heterogeneous datasets. Both the size and the heretogeneity of life science data make their integration and analysis by domain experts impractical and prone to the streetlight effect (they will pick up the models that best match what they know or what they would like to discover). Our first objective involves the formalization and management of knowledge, that is, the explicitation of relations occurring in structured data. In this setting, our main goal is to facilitate and optimize the integration Semantic Web resources with local users data by relying on the implicit data scheme contained in biological data and Semantic Web resources.

Reasoning over structured data with constraint-based logical paradigms Another level of complexity in life science integration is that very few paradigms exist to model the behavior of a complex biological system. This leads biologists to perform and formulate hypotheses in order to interpret their data. Our strategy is to interpret such hypotheses as combinatorial optimization problems allowing to reduce the family of models compatible with data. To that goal, we collaborate with Potsdam University in order to use and challenge the most recent developments of Answer Set Programming (ASP) [43], a logical paradigm for solving constraint satisfiability and combinatorial optimization issues. Our goal is therefore to provide scalable and expressive formal models of queries on biological networks with the focus of integrating dynamical information as explicit logical constraints in the modeling process.

Characterizing biological sequences with formal syntactic models Our last goal is to identify and characterize the function and expression of genes in non-model species, such as enzymes and isoforms functions in biological networks or specific functional features of metagenomic samples. These are insufficiently precise because of the divergence of biological sequences, the complexity of molecular structures and biological processes, and the weak signals characterizing these elements. Our goal is therefore to develop accurate formal syntactic models (automata, grammars, abstract gene models) enabling us to represent sequence conservation, sets of short and degenerated patterns and crossing or distant dependencies. This requires both to determine classes of formal syntactic models allowing to handle biological complexity, and to automatically characterize the functional potential embodied in biological sequences with these models.

3.2. Scalable methods to query data heterogenity

Confronted to large and complex data sets (raw data are associated with graphs depicting explicit or implicit links and correlations) almost all scientific fields have been impacted by the *big data issue*especially genomics and astronomy [47]. In our opinion, life sciences cumulates several features that are very specific and prevent the direct application of big data strategies that proved successful in other domains such as experimental physics: the existence of **several scales of granularity** from microscopic to macroscopic and the associated issue of dependency propagation, datasets **incompleteness and uncertainty** including highly **heterogeneous** responses to a perturbation from one sample to another, and highly fragmented sources of information that **lacks interoperability** [42]. To explore this research field, we use techniques from symbolic data mining (Semantic Web technologies, symbolic clustering, constraint satisfaction and grammatical modelling) to take into account those life science features in the analysis of biological data.

3.2.1. Research topics

Facilitating data integration and querying The quantity and inner complexity of life science data require semantically-rich analysis methods. A major challenge is then to combine data (from local project as well as from reference databases) and symbolic knowledge seamlessly. Semantic Web technologies provide a relevant framework, as demonstrated by the success of Linked (Open) Data [34]. However, life science end users (1) find it difficult to learn the languages for representing and querying Semantic Web data, and consequently (2) miss the possibility they had to interact with their tabulated data (even when doing so was exceedingly slow and tedious). Our first objective in this axis is to develop accurate abstractions of datasets or knowledge repositories to facilitate their exploration with RDF-based technologies.

Scalability of semantic web queries. A bottleneck in data querying is given by the performance of federated SPARQL queries, which must be improved by several orders of magnitude to allow current massive data to be analyzed. In this direction, our research program focuses the combination of *linked data fragments* [48], query properties and dataset structure for decomposing federated SPARQL queries.

Building and compressing static maps of interacting compounds A final approach to handle heterogeneity is to gather multi-scale data knowledge into functional static map of biological models that can be analyzed and/or compressed. This requires to linking genomics, metabolomics, expression data and protein measurement of several phenotypes into unified frameworks. In this direction, our main goal is to develop families of constraints, inspired by symbolic dynamical systems, to link datasets together. We currently focus on health (personalized medicine) and environmental (role of non-coding regulations, compression) datasets.

3.2.2. Associated software tools

AskOmics platform *AskOmics* is an integration and interrogation software for linked biological data based on semantic web technologies [url]. *AskOmics* aims at bridging the gap between end user data and the Linked (Open) Data cloud (LOD cloud). It allows heterogeneous bioinformatics data (formatted as tabular files or directly in RDF) to be loaded into a Triple Store system using a user-friendly web interface. It helps end users to (1) take advantage of the information readily available in the LOD cloud for analyzing there own data and (2) contribute back to the linked data by representing their data and the associated metadata in the proper format as well as by linking them to other resources. An originality is the graphical interface that allows any dataset to be integrated in a local RDF datawarehouse and SPARQL query to be built transparently and iteratively by a non-expert user.

FinGoc-tools *The FinGoc tools* allow filtering interaction networks with graph-based optimization criteria in order to elucidate the main regulators of an observed phenotype. The main added-value of these tools is to make explicit the criteria used to highlight the role of the main regulators.(1) The KeyRegulatorFinder package searches key regulators of lists of molecules (like metabolites, enzymes or genes) by taking advantage of knowledge databases in cell metabolism and signaling [package]. (2) The PowerGrasp python package implements graph compression methods oriented toward visualization, and based on power graph analysis [package]. (3) The iggy package enables the repairing of an interaction graph with respect to expression data. [Python package]

3.3. Metabolism: from enzyme sequences to systems ecology

Our researches in bioinformatics in relation with metabolic processes are driven by the understanding of non-model (eukaryote) species. Their metabolism have acquired specific features that we wish to identify with computational methods. To that goal, we combine sequence analysis with metabolic network analysis, with the final goal to understand better the metabolism of communities of organisms.

3.3.1. Research topics

Genomic level: characterizing enzymatic functions of protein sequences Precise characterization of functional proteins, such as enzymes or transporters, is a key to better understand and predict the actors involved in a metabolic process. In order to improve the precision of functional annotations, we develop machine learning approaches taking a sample of functional sequences as input to infer a grammar representing their key syntactical characteristics, including dependencies between residues. Our first goal is to enable an automatic semi-supervised refinement of enzymes classification [5] by combining the Protomata-Learner [38] framework - which captures local dependencies - with formal concept analysis. More challenging, we are exploring the learn of grammars representing long-distance dependencies such as those exhibited by contacts of amino-acids that are far in the sequence but close in the 3D protein folding.

System level: enriching and comparing metabolic networks for non-model organisms Non-model organisms are associated with often incomplete and poorly annotated sequences, leading to draft networks of their metabolism which largely suffer from incompleteness. in former studies, the team has developed several methods to improve the quality of eukaryotes metabolic networks, by solving several variants of the so-called *Metabolic Network gap-filling problem* with logical programming approaches [9], [8]. The main drawback of these approaches is that they cannot scale to the reconstruction and comparison of families of metabolic networks. Our main objective is therefore to develop new tools for the comparison of species strains at the metabolic level.

Consortium level: exploring the diversity of community consortia A new emerging field is system ecology, which aims at building predictive models of species interactions within an ecosystem for deciphering cooperative and competitive relationships between species [41].This field raise two new issues (1) uncertainty on the species present in the ecosystem and (2) uncertainty about the global objective governing an ecosystem. To address these challenges, our first research focus is the inference of metabolic exchanges and relationships from transporter identification, based on our expertise in metabolic network gap-filling. A second very challenging focus is the prediction of transporters families by obtaining refined characterization of transporters, which are quite unexplored apart from specific databases [45].

3.3.2. Associated software tools

Protomata is a machine learning suite for the inference of automata characterizing (functional) families of proteins from available sequences by modeling alternative local dependencies. They are well suited to predict new family members with a high specificity [url]. The tool builds sequences alignments (partial and local), learns automata and searches for new family members in sequence databases. Applications of Protomata tools include automatic updating of the cyanolase database [38] and the refinement of the classification of HAD enzymes [5].

AuReMe workspace is designed for tractable reconstruction of metabolic networks [url]. The toolbox allows for the Automatic Reconstruction of Metabolic networks based on the combination of multiple heterogeneous data and knowledge sources [12]. The main added-values are the inclusion of graph-based tools relevant for the study of non-classical organisms (Meneco, Menetools, Shogen packages), the possibility to trace the reconstruction and curation procedures (Padmet package), and the exploration of reconstructed metabolic networks with wikis (wiki-export package). It has been used for reconstructing metabolic networks of micro and macro-algae [44], extremophile bacteria [39] and communities of organisms [3].

3.4. Regulation and signaling: detecting complex and discriminant signatures of phenotypes

On the contrary to metabolic networks, regulatory and signaling processes in biological systems involves agents interacting at different granularity levels (from genes, non-coding RNAs to protein complexes) and different time-scales. Our focus is on the reconstruction of large-scale networks involving multiple scales processes, from which controllers can be extracted with symbolic dynamical systems methods. A particular attention is paid to the characterization of products of genes (such as isoform) and of perturbations to identify discriminant signature of pathologies.

3.4.1. Research topics

Genomic level: characterizing gene structure with grammatical languages and conservation information The subject here is to accurately represent gene structure, including intron/exon structure, for predicting the products of genes, such as isoform transcripts, and comparing the expression potential of a eukaryotic gene according to its context (e.g. tissue) or according to the species. Our approach consists in designing grammatical and comparative-genomics based models for gene structures able to detect heterogeneous functional sites (splicing sites, regulatory binding sites...), functional regions (exons, promotors...) and global constraints (translation into proteins) [35]. Accurate gene models are defined by identifying general constraints shaping gene families and their structures conserved over evolution. Syntactic elements controlling gene expression (transcription factor binding sites controlling transcription; enhancers and silencers controlling splicing events...), that is, short, degenerated and overlapping functional sequences, are modeled by relying on the high capability of SVG grammars to deal with structure and ambiguity [46].

System level: extracting causal signatures of complex phenotypes with systems biology frameworks The main challenge we address is to set up a generic formalism to model inter-layer interactions in large-scale biological networks. To that goal, we have developed several types of abstractions: multi-experiments framework to learn and control signaling networks [10], multi-layer reactions in interaction graphs [36], and multi-layer information in large-scale Petri nets [33]. Our main issues are to scale these approaches to standardized large-scale repositories by relying on the interoperable Linked Open Data (LOD) resources and to enrich them with ad-hoc regulations extracted from sequence-based analysis. This will allow us to characterize changes in system attractors induced by mutations and how they may be included in pathology signatures.

3.4.2. Associated software tools

Logol software is designed for complex pattern modelling and matching [url] It is a swiss-army-knife for pattern matching on DNA/RNA/Protein sequences, based on expressive patterns which consist in a complex combination of motifs (such as degenerated strings) and structures (such as imperfect stem-loop ou repeats) [1]. *Logol* key features are the possibilities (i) to divide a pattern description into several sub-patterns, (ii) to model long range dependencies, and (iii) to enable the use of ambiguous models or to permit the inclusion of negative conditions in a pattern definition. Therefore, *Logol* encompasses most of the features of specialized tools (Vmatch, Patmatch, Cutadapt, HMM) and enables interplays between several classes of patterns (motifs and structures), including stem-loop identification in CRISPR.

Caspo software Cell ASP Optimizer (*Caspo*) constitutes a pipeline for automated reasoning on logical signaling networks (learning, classifying, designing experimental perturbations, identifying controllers, take time-series into account) [url]. The software handles inherent experimental noise but enumerating all different logical networks which are compatible with a set of experimental observations [10]. The main advantage is that is enables a complete study of logical network without requiring any linear constraint programs.

Cadbiom package aims at building and analyzing the asynchronous dynamics of enriched logical networks [url] It is based on Guarded transition semantic and allows synchronization events to be investigated in large-scale biological networks [33]. For instance, it was designed to allow controler of phenotypes in large-scale knowledge databases (PID) to be curated and analyzed [4].

FLUMINANCE Project-Team

3. Research Program

3.1. Estimation of fluid characteristic features from images

The measurement of fluid representative features such as vector fields, potential functions or vorticity maps, enables physicists to have better understanding of experimental or geophysical fluid flows. Such measurements date back to one century and more but became an intensive subject of research since the emergence of correlation techniques [50] to track fluid movements in pairs of images of a particles laden fluid or by the way of clouds photometric pattern identification in meteorological images. In computer vision, the estimation of the projection of the apparent motion of a 3D scene onto the image plane, referred to in the literature as optical-flow, is an intensive subject of researches since the 80's and the seminal work of B. Horn and B. Schunk [61]. Unlike to dense optical flow estimators, the former approach provides techniques that supply only sparse velocity fields. These methods have demonstrated to be robust and to provide accurate measurements for flows seeded with particles. These restrictions and their inherent discrete local nature limit too much their use and prevent any evolutions of these techniques towards the devising of methods supplying physically consistent results and small scale velocity measurements. It does not authorize also the use of scalar images exploited in numerous situations to visualize flows (image showing the diffusion of a scalar such as dye, p ollutant, light index refraction, fluorescein,...). At the opposite, variational techniques enable in a well-established mathematical framework to estimate spatially continuous velocity fields, which should allow more properly to go towards the measurement of smaller motion scales. As these methods are defined through PDE's systems they allow quite naturally constraints to be included such as kinematic properties or dynamic laws governing the observed fluid flows. Besides, within this framework it is also much easier to define characteristic features estimation procedures on the basis of physically grounded data model that describes the relation linking the observed luminance function and some state variables of the observed flow. The Fluminance group has allowed a substantial progress in this direction with the design of dedicated dense estimation techniques to estimate dense fluid motion fields. See [7] for a detailed review. More recently problems related to scale measurement and uncertainty estimation have been investigated [55]. Dynamically consistent and highly robust techniques have been also proposed for the recovery of surface oceanic streams from satellite images [52]. Very recently parameter-free approaches relying on uncertainty concept has been devised [53]. This technique outperforms the state of the art.

3.2. Data assimilation and Tracking of characteristic fluid features

Real flows have an extent of complexity, even in carefully controlled experimental conditions, which prevents any set of sensors from providing enough information to describe them completely. Even with the highest levels of accuracy, space-time coverage and grid refinement, there will always remain at least a lack of resolution and some missing input about the actual boundary conditions. This is obviously true for the complex flows encountered in industrial and natural conditions, but remains also an obstacle even for standard academic flows thoroughly investigated in research conditions.

This unavoidable deficiency of the experimental techniques is nevertheless more and more compensated by numerical simulations. The parallel advances in sensors, acquisition, treatment and computer efficiency allow the mixing of experimental and simulated data produced at compatible scales in space and time. The inclusion of dynamical models as constraints of the data analysis process brings a guaranty of coherency based on fundamental equations known to correctly represent the dynamics of the flow (e.g. Navier Stokes equations) [11]. Conversely, the injection of experimental data into simulations ensures some fitting of the model with reality.

To enable data and models coupling to achieve its potential, some difficulties have to be tackled. It is in particular important to outline the fact that the coupling of dynamical models and image data are far from being straightforward. The first difficulty is related to the space of the physical model. As a matter of fact, physical models describe generally the phenomenon evolution in a 3D Cartesian space whereas images provides generally only 2D tomographic views or projections of the 3D space on the 2D image plane. Furthermore, these views are sometimes incomplete because of partial occlusions and the relations between the model state variables and the image intensity function are otherwise often intricate and only partially known. Besides, the dynamical model and the image data may be related to spatio-temporal scale spaces of very different natures which increases the complexity of an eventual multiscale coupling. As a consequence of these difficulties, it is necessary generally to define simpler dynamical models in order to assimilate image data. This redefinition can be done for instance on an uncertainty analysis basis, through physical considerations or by the way of data based empirical specifications. Such modeling comes to define inexact evolution laws and leads to the handling of stochastic dynamical models. The necessity to make use and define sound approximate models, the dimension of the state variables of interest and the complex relations linking the state variables and the intensity function, together with the potential applications described earlier constitute very stimulating issues for the design of efficient data-model coupling techniques based on image sequences.

On top of the problems mentioned above, the models exploited in assimilation techniques often suffer from some uncertainties on the parameters which define them. Hence, a new emerging field of research focuses on the characterization of the set of achievable solutions as a function of these uncertainties. This sort of characterization indeed turns out to be crucial for the relevant analysis of any simulation outputs or the correct interpretation of operational forecasting schemes. In this context, stochastic modeling play a crucial role to model and process uncertainty evolution along time. As a consequence, stochastic parameterization of flow dynamics has already been present in many contributions of the Fluminance group in the last years and will remain a cornerstone of the new methodologies investigated by the team in the domain of uncertainty characterization.

This wide theme of research problems is a central topic in our research group. As a matter of fact, such a coupling may rely on adequate instantaneous motion descriptors extracted with the help of the techniques studied in the first research axis of the FLUMINANCE group. In the same time, this coupling is also essential with respect to visual flow control studies explored in the third theme. The coupling between a dynamics and data, designated in the literature as a Data Assimilation issue, can be either conducted with optimal control techniques [62], [63] or through stochastic filtering approaches [56], [59]. These two frameworks have their own advantages and deficiencies. We rely indifferently on both approaches.

3.3. Optimization and control of fluid flows with visual servoing

Fluid flow control is a recent and active research domain. A significant part of the work carried out so far in that field has been dedicated to the control of the transition from laminarity to turbulence. Delaying, accelerating or modifying this transition is of great economical interest for industrial applications. For instance, it has been shown that for an aircraft, a drag reduction can be obtained while enhancing the lift, leading consequently to limit fuel consumption. In contrast, in other application domains such as industrial chemistry, turbulence phenomena are encouraged to improve heat exchange, increase the mixing of chemical components and enhance chemical reactions. Similarly, in military and civilians applications where combustion is involved, the control of mixing by means of turbulence handling rouses a great interest, for example to limit infra-red signatures of fighter aircraft.

Flow control can be achieved in two different ways: passive or active control. Passive control provides a permanent action on a system. Most often it consists in optimizing shapes or in choosing suitable surfacing (see for example [54] where longitudinal riblets are used to reduce the drag caused by turbulence). The main problem with such an approach is that the control is, of course, inoperative when the system changes. Conversely, in active control the action is time varying and adapted to the current system's state. This approach requires an external energy to act on the system through actuators enabling a forcing on the flow through for instance blowing and suction actions [66], [58]. A closed-loop problem can be formulated as an optimal control issue where a control law minimizing an objective cost function (minimization of the drag, minimization of the actuators power, etc.) must be applied to the actuators [51]. Most of the works of the literature indeed comes back to open-loop control approaches [65], [60], [64] or to forcing approaches [57] with control laws acting without any feedback information on the flow actual state. In order for these methods to be operative, the model used to derive the control law must describe as accurately as possible the flow and all the eventual perturbations of the surrounding environment, which is very unlikely in real situations. In addition, as such approaches rely on a perfect model, a high computational costs is usually required. This inescapable pitfall has motivated a strong interest on model reduction. Their key advantage being that they can be specified empirically from the data and represent quite accurately, with only few modes, complex flows' dynamics. This motivates an important research axis in the Fluminance group.

3.4. Numerical models applied to hydrogeology and geophysics

The team is strongly involved in numerical models for hydrogeology and geophysics. There are many scientific challenges in the area of groundwater simulations. This interdisciplinary research is very fruitful with cross-fertilizing subjects.

In geophysics, a main concern is to solve inverse problems in order to fit the measured data with the model. Generally, this amounts to solve a linear or nonlinear least-squares problem.

Models of geophysics are in general coupled and multi-physics. For example, reactive transport couples advection-diffusion with chemistry. Here, the mathematical model is a set of nonlinear Partial Differential Algebraic Equations. At each timestep of an implicit scheme, a large nonlinear system of equations arise. The challenge is to solve efficiently and accurately these large nonlinear systems.

3.5. Numerical algorithms and high performance computing

Linear algebra is at the kernel of most scientific applications, in particular in physical or chemical engineering. The objectives are to analyze the complexity of these different methods, to accelerate convergence of iterative methods, to measure and improve the efficiency on parallel architectures, to define criteria of choice.

GENSCALE Project-Team

3. Research Program

3.1. Axis 1: Data Structure

The aim of this axis is to develop efficient data structures for representing the mass of genomic data generated by the sequencing machines. This research is motivated by the fact that the treatments of large genomes, such as mammalian or plant genomes, require high computing resources, and more specifically very important memory configuration. For example, the ABYSS software used 4.3TB of memory to assemble the white spruce genome [36]. The main reason for such memory consumption is that the data structures used in ABYSS are far from optimal (and this is also the case for many assembly software).

Our research focuses on the de-Bruijn graph structure. This well-known data structure, directly built from raw sequencing data, have many properties matching perfectly well with NGS (Next Generation Sequencing) processing requirements (see next section). Here, the question we are interested in is how to provide a low memory footprint implementation of the de-Bruijn graph to process very large NGS datasets, including metagenomic ones [3], [4].

Another research direction of this axis is the indexing of large sets of objects. A typical, but non exclusive, need is to annotate nodes of the de-Bruijn graph, that is potentially billions of items. Again, very low memory footprint indexing structures are mandatory to manage a very large quantity of objects [5].

3.2. Axis 2: Algorithms

The main goal of the GenScale team is to develop optimized tools dedicated to NGS processing. Optimization can be seen both in terms of space (low memory footprint) and in terms of time (fast execution time). The first point is mainly related to advanced data structures as presented in the previous section (axis 1). The second point relies on new algorithms and, when possible implementation on parallel structures (axis 3).

We do not have the ambition to cover the vast panel of software related to NGS needs. We particularly focused on the following areas:

- **NGS data Compression** De-Bruijn graphs are de facto a compressed representation of the NGS information from which very efficient and specific compressors can be designed. Furthermore, compressing the data using smart structures may speed up some downstream graph-based analyses since a graph structure is already built [1].
- **Genome assembly** This task remains very complicated, especially for large and complex genomes, such as plant genomes with polyploid and highly repeated structures. We worked both on the generation of contigs [3] and on the scaffolding step [26].
- **Detection of variants** This is often the first information we want to extract from billions of reads. Variant structures range from SNPs or short indels to large insertions/deletions and long inversions over the chromosomes. We developed original methods to find variants without any reference genome [7]. We also worked on the detection of structural variants using approaches of local assembly [6].
- **Metagenomics** We focussed our research on comparative metagenomics by providing methods able to compare hundreds of metagenomic samples together. This is achieved by combining very low memory data structures and efficient implementation and parallelization on large clusters [2].
- **Genome Wide Association Study (GWAS)** We tackle this problem with algorithms commonly used in data mining. From two cohorts of individuals (case and control) we can exhibit statistically significant *patterns* spaning over full genomes.

In addition, we also proposed new algorithmic solutions for analyzing third generation sequencing data, in order to benefit from their larger read size while taking into account their higher sequencing error rate [16].

3.3. Axis 3: Parallelism

This third axis investigates another lever to increase performances and scalability of NGS treatments. There are many levels of parallelism that can be used and/or combined to reduce the execution time of very time-consuming bioinformatics processes. A first level is the parallel nature of today processors that now house several cores. A second level is the grid structure that is present in all bioinformatics centers or in the cloud. This two levels are generally combined: a node of a grid is often a multicore system. Another possibility is to add hardware accelerators to a processor. A GPU board is a good example.

GenScale does not do explicit research on parallelism. It exploits the capacity of computing resources to support parallelism. The problem is addressed in two different directions. The first is an engineering approach that uses existing parallel tools to implement algorithms such as multithreading or MapReduce techniques [4]. The second is a parallel algorithmic approach: during the development step, the algorithms are constrained by parallel criteria [2]. This is particularly true for parallel algorithms targeting hardware accelerators.

SERPICO Project-Team

3. Research Program

3.1. Statistics and algorithms for computational microscopy

Fluorescence microscopy limitations are due to the optical aberrations, the resolution of the microscopy system, and the photon budget available for the biological specimen. Hence, new concepts have been defined to address challenging image restoration and molecule detection problems while preserving the integrity of samples. Accordingly, the main stream regarding denoising, deconvolution, registration and detection algorithms advocates appropriate signal processing framework to improve spatial resolution, while at the same time pushing the illumination to extreme low levels in order to limit photo-damages and phototoxicity. As a consequence, the question of adapting cutting-edge signal denoising and deconvolution, object detection, and image registration methods to 3D fluorescence microscopy imaging has retained the attention of several teams over the world.

In this area, the Serpico team has developed a strong expertise in key topics in computational imaging including image denoising and deconvolution, object detection and multimodal image registration. Several algorithms proposed by the team outperformed the state-of-the-art results, and some developments are compatible with "high-throughput microscopy" and the processing of several hundreds of cells. We especially promoted non local, non-parametric and patch-based methods to solve well-known inverse problems or more original reconstruction problems. A recent research direction consists in adapting the deep learning concept to solve challenging detection and reconstruction problems in microscopy. We have investigated convolution neural networks to detect small macromolecules in 3D noisy electron images with promising results. The next step consists in proposing smart paradigms and architectures to save memory and computations.

More generally, many inverse problems and image processing become intractable with modern 3D microscopy, because very large temporal series of volumes (200 to 1000 images per second for one 3D stack) are acquired for several hours. Novel strategies are needed for 3D image denoising, deconvolution and reconstruction since computation is extremely heavy. Accordingly, we will adapt the estimator aggregation approach developed for optical flow computation to meet the requirements of 3D image processing. We plan to investigate regularization-based aggregation energy over super-voxels to reduce complexity, combined to modern optimization algorithms. Finally, we will design parallelized algorithms that fast process 3D images, perform energy minimization in few seconds per image, and run on low-cost graphics processor boards (GPU).

3.2. From image data to motion descriptors: trajectory computation and dynamics analysis

Several particle tracking methods for intracellular analysis have been tailored to cope with different types of cellular and subcellular motion down to Brownian single molecule behavior. Many algorithms were carefully evaluated on the particle tracking challenge dataset published in the Nature Methods journal in 2014. Actually, there is no definitive solution to the particle tracking problem which remains application-dependent in most cases. The work of Serpico in particle motion analysis is significant in multiple ways, and inserts within a very active international context. One of the remaining key open issues is the tracking of objects with heterogeneous movements in crowded configurations. Moreover, particle tracking methods are not always adapted for motion analysis, especially when the density of moving features hampers the individual extraction of objects of interest undergoing complex motion. Estimating flow fields can be more appropriate to capture the complex dynamics observed in biological sequences. The existing optical flow methods can be classified into two main categories: i/ local methods impose a parametric motion model (e.g. local translation) in a given neighborhood; ii/ global methods estimate the dense motion field by minimizing a global energy functional composed of a data term and a regularization term.

The Serpico team has developed a strong expertise in key topics, especially in object tracking for fluorescence microscopy, optical flow computation and high-level analysis of motion descriptors and trajectories. Several algorithms proposed by the team are very competitive when compared to the state-of-the-art results, and our new paradigms offer promising ways for molecule traffic quantification and analysis. Amongst the problems that we currently address, we can mention: computation of 3D optical flow for large-size images, combination of two frame-based differential methods and sparse sets of trajectories, detection and analysis of unexpected local motion patterns in global coherent collective motion. Development of efficient numerical schemes will be central in the future but visualization methods are also crucial for evaluation and quality assessment. Another direction of research consists in exploiting deep learning to 3D optical flow so as to develop efficient numerical schemes that naturally capture complex motion patterns. Investigation in machine learning and statistics will be actually conducted in the team in the two first research axes to address a large range of inverse problems in bioimaging. Deep learning is an appealing approach since expertise of biologists, via iterative annotation of training data, will be included in the design of image analysis schemes.

3.3. Biological and biophysical models and spatial statistics for quantitative bioimaging

A number of stochastic mathematical models were proposed to describe various intracellular trafficking, where molecules and proteins are transported to their destinations via free diffusion, subdiffusion and ballistic motion representing movements along the cytoskeleton networks assisted by molecular motors. Accordingly, the study of diffusion and stochastic dynamics has known a growing interest in bio-mathematics, biophysics and cell biology with the popularization of fluorescence dynamical microscopy and super-resolution imaging. In this area, the competing teams mainly studied MSD and fluorescence correlation spectroscopy methods.

In the recent period, the Serpico team achieved important results for diffusion-related dynamics involved in exocytosis mechanisms. Robustness to noise has been well investigated, but robustness to environmental effects has yet to be effectively achieved. Particular attention has been given to the estimation of particle motion regime changes, but the available results are still limited for analysing short tracks. The analysis of spatiotemporal molecular interactions from set of 3D computed trajectories or motion vector fields (e.g., co-alignment) must be investigated to fully quantify specific molecular machineries. We have already made efforts in that directions this year (e.g., for colocalization) but important experiments are required to make our preliminary algorithms reliable enough and well adapted to specific transport mechanisms.

Accordingly, we will study quantification methods to represent interactions between molecules and trafficking around three lines of research. First, we will focus on 3D space-time global and local object-based co-orientation and co-alignment methods, in the line of previous work on colocalization, to quantify interactions between molecular species. In addition, given N tracks associated to N molecular species, interaction descriptors, dynamics models and stochastic graphical models representing molecular machines will be studied in the statistical data assimilation framework. Second, we will analyse approaches to estimate molecular mobility, active transport and motion regime changes from computed trajectories in the Lagrangian and Eulerian settings. We will focus on the concept of super-resolution to provide spatially high-resolved maps of diffusion and active transport parameters based on stochastic biophysical models and sparse image representation. Third, we plan to extend the aggregation framework dedicated to optical flow to the problem of diffusion-transport estimation. Finally, we will investigate data assimilation methods to better combine algorithms, models, and experiments in an iterative and virtuous circle. The overview of ultrastructural organization will be achieved by additional 3D electron microscopy technologies.

VISAGES Project-Team

3. Research Program

3.1. Research Program

The scientific foundations of our team concern the development of new processing algorithms in the field of medical image computing : image fusion (registration and visualization), image segmentation and analysis, management of image related information. Since this is a very large domain, which can be applied on numerous types of application; for seek of efficiency, the purpose of our methodological work primarily focuses on clinical aspects and for the most part on head and neck related diseases. In addition, we emphasize our research efforts on the neuroimaging domain. Concerning the scientific foundations, we have pushed our research efforts:

- In the field of image fusion and image registration (rigid and deformable transformations) with a special emphasis on new challenging registration issues, especially when statistical approaches based on joint histogram cannot be used or when the registration stage has to cope with loss or appearance of material (like in surgery or in tumor imaging for instance).
- In the field of image analysis and statistical modeling with a new focus on image feature and group analysis problems. A special attention was also to develop advanced frameworks for the construction of atlases and for automatic and supervised labeling of brain structures.
- In the field of image segmentation and structure recognition, with a special emphasis on the difficult problems of *i*) image restoration for new imaging sequences (new Magnetic Resonance Imaging protocols, 3D ultrasound sequences...), and *ii*) structure segmentation and labelling based on shape, multimodal and statistical information.
- Following past national projects where we had leading roles (e.g., Neurobase, NeuroLog, . . .), we wanted to enhance the development of distributed and heterogeneous medical image processing systems.

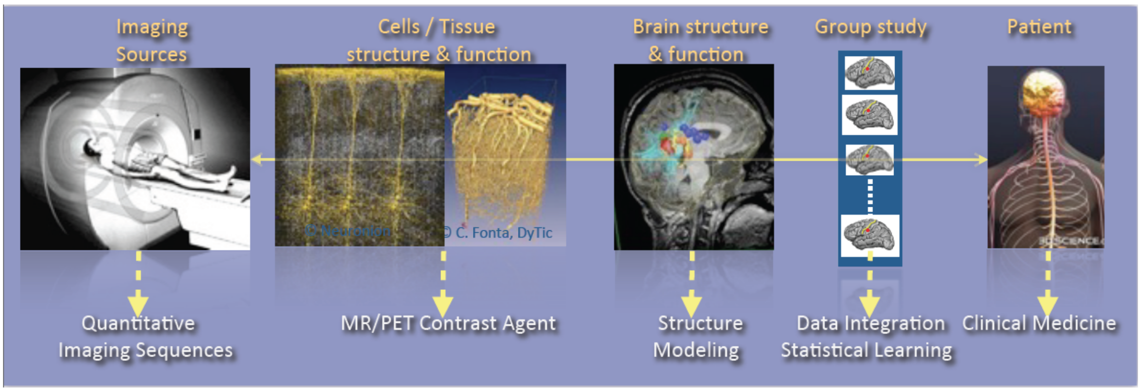
Imaging Sources
Cells / Tissue structure & function
Brain structure & function
Group study
Patient
© Neuronion
© C. Fonta, DyTic
Quantitative Imaging Sequences
MR/PET Contrast Agent
Structure Modeling
Data Integration Statistical Learning
Clinical Medicine

Figure 1. The major overall scientific foundation of the team concerns the integration of data from the Imaging source to the patient at different scales: from the cellular or molecular level describing the structure and function, to the functional and structural level of brain structures and regions, to the population level for the modelling of group patterns and the learning of group or individual imaging markers.

As shown in Fig. 1 , research activities of the VISAGES U1228 team are tightly coupling observations and models through integration of clinical and multi-scale data, phenotypes (cellular, molecular or structural patterns). We work on personalized models of central nervous system organs and pathologies, and intend to confront these models to clinical investigation studies for quantitative diagnosis, prevention of diseases, therapy planning and validation. These approaches are developed in a translational framework where the data integration process to build the models inherits from specific clinical studies, and where the models are assessed on prospective clinical trials for diagnosis and therapy planning. All of this research activity is conducted in tight links with the Neurinfo imaging platform environments and the engineering staff of the platform. In this context, some of our major challenges in this domain concern:

- The elaboration of new descriptors to study the brain structure and function (e.g., variation of brain perfusion with and without contrast agent, evolution in shape and size of an anatomical structure in relation with normal, pathological or functional patterns, computation of asymmetries from shapes and volumes).
- The integration of additional spatio-temporal imaging sequences covering a larger range of observation, from the molecular level to the organ through the cell (Arterial Spin Labeling, diffusion MRI, MR relaxometry, MR cell labeling imaging, PET molecular imaging, . . .). This includes the elaboration of new image descriptors coming from spatio-temporal quantitative or contrast-enhanced MRI.
- The creation of computational models through data fusion of molecular, cellular, structural and functional image descriptors from group studies of normal and/or pathological subjects.
- The evaluation of these models on acute pathologies especially for the study of degenerative, psychiatric or developmental brain diseases (e.g., Multiple Sclerosis, Epilepsy, Parkinson, Dementia, Strokes, Depression, Schizophrenia, . . .) in a translational framework.

In terms of methodological developments, we are particularly working on statistical methods for multidimensional image analysis, and feature selection and discovery, which include:

- The development of specific shape and appearance models, construction of atlases better adapted to a patient or a group of patients in order to better characterize pathologies;
- The development of advanced segmentation and modeling methods dealing with longitudinal and multidimensional data (vector or tensor fields), especially with the integration of new prior models to control the integration of multiscale data and aggregation of models;
- The development of new models and probabilistic methods to create water diffusion maps from MRI;
- The integration of machine learning procedures for classification and labeling of multidimensional features (from scalar to tensor fields and/or geometric features): pattern and rule inference and knowledge extraction are key techniques to help in the elaboration of knowledge in the complex domains we address;
- The development of new dimensionality reduction techniques for problems with massive data, which includes dictionary learning for sparse model discovery. Efficient techniques have still to be developed to properly extract from a raw mass of images derived data that are easier to analyze.

DIONYSOS Project-Team

3. Research Program

3.1. Introduction

The scientific foundations of our work are those of network design and network analysis. Specifically, this concerns the principles of packet switching and in particular of IP networks (protocol design, protocol testing, routing, scheduling techniques), and the mathematical and algorithmic aspects of the associated problems, on which our methods and tools are based.

These foundations are described in the following paragraphs. We begin by a subsection dedicated to Quality of Service (QoS) and Quality of Experience (QoE), since they can be seen as unifying concepts in our activities. Then we briefly describe the specific sub-area of model evaluation and about the particular multidisciplinary domain of network economics.

3.2. Quality of Service and Quality of Experience

Since it is difficult to develop as many communication solutions as possible applications, the scientific and technological communities aim towards providing general *services* allowing to give to each application or user a set of properties nowadays called "Quality of Service" (QoS), a terminology lacking a precise definition. This QoS concept takes different forms according to the type of communication service and the aspects which matter for a given application: for performance it comes through specific metrics (delays, jitter, throughput, etc.), for dependability it also comes through appropriate metrics: reliability, availability, or vulnerability, in the case for instance of WAN (Wide Area Network) topologies, etc.

QoS is at the heart of our research activities: We look for methods to obtain specific "levels" of QoS and for techniques to evaluate the associated metrics. Our ultimate goal is to provide tools (mathematical tools and/or algorithms, under appropriate software "containers" or not) allowing users and/or applications to attain specific levels of QoS, or to improve the provided QoS, if we think of a particular system, with an optimal use of the resources available. Obtaining a good QoS level is a very general objective. It leads to many different areas, depending on the systems, applications and specific goals being considered. Our team works on several of these areas. We also investigate the impact of network QoS on multimedia payloads to reduce the impact of congestion.

Some important aspects of the behavior of modern communication systems have subjective components: the quality of a video stream or an audio signal, *as perceived by the user*, is related to some of the previous mentioned parameters (packet loss, delays, ...) but in an extremely complex way. We are interested in analyzing these types of flows from this user-oriented point of view. We focus on the *user perceived quality*, in short, PQ, the main component of what is nowadays called Quality of Experience (in short, QoE), to underline the fact that, in this case, we want to center the analysis on the user. In this context, we have a global project called PSQA, which stands for Pseudo-Subjective Quality Assessment, and which refers to a technology we have developed allowing to automatically measure this PQ.

Another special case to which we devote research efforts in the team is the analysis of qualitative properties related to interoperability assessment. This refers to the act of determining if end-to-end functionality between at least two communicating systems is as required by the base standards for those systems. Conformance is the act of determining to what extent a single component conforms to the individual requirements of the standard it is based on. Our purpose is to provide such a formal framework (methods, algorithms and tools) for interoperability assessment, in order to help in obtaining efficient interoperability test suites for new generation networks, mainly around IPv6-related protocols. The interoperability test suites generation is based on specifications (standards and/or RFCs) of network components and protocols to be tested.

3.3. Stochastic modeling

The scientific foundations of our modeling activities are composed of stochastic processes theory and, in particular, Markov processes, queuing theory, stochastic graphs theory, etc. The objectives are either to develop numerical solutions, or analytical ones, or possibly discrete event simulation or Monte Carlo (and Quasi-Monte Carlo) techniques. We are always interested in model evaluation techniques for dependability and performability analysis, both in static (network reliability) and dynamic contexts (depending on the fact that time plays an explicit role in the analysis or not). We look at systems from the classical so-called *call level*, leading to standard models (for instance, queues or networks of queues) and also at the *burst level*, leading to *fluid models*.

In recent years, our work on the design of the topologies of WANs led us to explore optimization techniques, in particular in the case of very large optimization problems, usually formulated in terms of graphs. The associated methods we are interested in are composed of simulated annealing, genetic algorithms, TABU search, etc. For the time being, we have obtained our best results with GRASP techniques.

Network pricing is a good example of a multi-disciplinary research activity half-way between applied mathematics, economy and networking, centered on stochastic modeling issues. Indeed, the Internet is facing a tremendous increase of its traffic volume. As a consequence, real users complain that large data transfers take too long, without any possibility to improve this by themselves (by paying more, for instance). A possible solution to cope with congestion is to increase the link capacities; however, many authors consider that this is not a viable solution as the network must respond to an increasing demand (and experience has shown that demand of bandwidth has always been ahead of supply), especially now that the Internet is becoming a commercial network. Furthermore, incentives for a fair utilization between customers are not included in the current Internet. For these reasons, it has been suggested that the current flat-rate fees, where customers pay a subscription and obtain an unlimited usage, should be replaced by usage-based fees. Besides, the future Internet will carry heterogeneous flows such as video, voice, email, web, file transfers and remote login among others. Each of these applications requires a different level of QoS: for example, video needs very small delays and packet losses, voice requires small delays but can afford some packet losses, email can afford delay (within a given bound) while file transfer needs a good average throughput and remote login requires small round-trip times. Some pricing incentives should exist so that each user does not always choose the best QoS for her application and so that the final result is a fair utilization of the bandwidth. On the other hand, we need to be aware of the trade-off between engineering efficiency and economic efficiency; for example, traffic measurements can help in improving the management of the network but is a costly option. These are some of the various aspects often present in the pricing problems we address in our work. More recently, we have switched to the more general field of network economics, dealing with the economic behavior of users, service providers and content providers, as well as their relations.

DIVERSE Project-Team

3. Research Program

3.1. Scientific background

3.1.1. Model-driven engineering

Model-Driven Engineering (MDE) aims at reducing the accidental complexity associated with developing complex software-intensive systems (e.g., use of abstractions of the problem space rather than abstractions of the solution space) [99]. It provides DIVERSE with solid foundations to specify, analyze and reason about the different forms of diversity that occur through the development lifecycle. A primary source of accidental complexity is the wide gap between the concepts used by domain experts and the low-level abstractions provided by general-purpose programming languages [69]. MDE approaches address this problem through modeling techniques that support separation of concerns and automated generation of major system artifacts from models (*e.g.,* test cases, implementations, deployment and configuration scripts). In MDE, a model describes an aspect of a system and is typically created or derived for specific development purposes [51]. Separation of concerns is supported through the use of different modeling languages, each providing constructs based on abstractions that are specific to an aspect of a system. MDE technologies also provide support for manipulating models, for example, support for querying, slicing, transforming, merging, and analyzing (including executing) models. Modeling languages are thus at the core of MDE, which participates to the development of a sound *Software Language Engineering*, including a unified typing theory that integrate models as first class entities [102].

Incorporating domain-specific concepts and high-quality development experience into MDE technologies can significantly improve developer productivity and system quality. Since the late nineties, this realization has led to work on MDE language workbenches that support the development of domain-specific modeling languages (DSMLs) and associated tools (*e.g.,* model editors and code generators). A DSML provides a bridge between the field in which domain experts work and the implementation (programming) field. Domains in which DSMLs have been developed and used include, among others, automotive, avionics, and the emerging cyber-physical systems. A study performed by Hutchinson et al. [75] provides some indications that DSMLs can pave the way for wider industrial adoption of MDE.

More recently, the emergence of new classes of systems that are complex and operate in heterogeneous and rapidly changing environments raises new challenges for the software engineering community. These systems must be adaptable, flexible, reconfigurable and, increasingly, self-managing. Such characteristics make systems more prone to failure when running and thus development and study of appropriate mechanisms for continuous design and run-time validation and monitoring are needed. In the MDE community, research is focused primarily on using models at design, implementation, and deployment stages of development. This work has been highly productive, with several techniques now entering a commercialization phase. As software systems are becoming more and more dynamic, the use of model-driven techniques for validating and monitoring run-time behavior is extremely promising [83].

3.1.2. Variability modeling

While the basic vision underlying *Software Product Lines* (SPL) can probably be traced back to David Parnas seminal article [92] on the Design and Development of Program Families, it is only quite recently that SPLs are emerging as a paradigm shift towards modeling and developing software system families rather than individual systems [90]. SPL engineering embraces the ideas of mass customization and software reuse. It focuses on the means of efficiently producing and maintaining multiple related software products, exploiting what they have in common and managing what varies among them.

Several definitions of the *software product line* concept can be found in the research literature. Clements *et al.* define it as *a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and are developed from a common set of core assets in a prescribed way* [89]. Bosch provides a different definition [57]: *A SPL consists of a product line architecture and a set of reusable components designed for incorporation into the product line architecture. In addition, the PL consists of the software products developed using the mentioned reusable assets*. In spite of the similarities, these definitions provide different perspectives of the concept: *market-driven*, as seen by Clements *et al.*, and *technology-oriented* for Bosch.

SPL engineering is a process focusing on capturing the *commonalities* (assumptions true for each family member) and *variability* (assumptions about how individual family members differ) between several software products [63]. Instead of describing a single software system, a SPL model describes a set of products in the same domain. This is accomplished by distinguishing between elements common to all SPL members, and those that may vary from one product to another. Reuse of core assets, which form the basis of the product line, is key to productivity and quality gains. These core assets extend beyond simple code reuse and may include the architecture, software components, domain models, requirements statements, documentation, test plans or test cases.

The SPL engineering process consists of two major steps:

1. **Domain Engineering**, or *development for reuse*, focuses on core assets development.
2. **Application Engineering**, or *development with reuse*, addresses the development of the final products using core assets and following customer requirements.

Central to both processes is the management of **variability** across the product line [71]. In common language use, the term *variability* refers to *the ability or the tendency to change*. Variability management is thus seen as the key feature that distinguishes SPL engineering from other software development approaches [58]. Variability management is thus growingly seen as the cornerstone of SPL development, covering the entire development life cycle, from requirements elicitation [104] to product derivation [109] to product testing [87], [86].

Halmans *et al.* [71] distinguish between *essential* and *technical* variability, especially at requirements level. Essential variability corresponds to the customer's viewpoint, defining what to implement, while technical variability relates to product family engineering, defining how to implement it. A classification based on the dimensions of variability is proposed by Pohl *et al.* [94]: beyond **variability in time** (existence of different versions of an artifact that are valid at different times) and **variability in space** (existence of an artifact in different shapes at the same time) Pohl *et al.* claim that variability is important to different stakeholders and thus has different levels of visibility: **external variability** is visible to the customers while **internal variability**, that of domain artifacts, is hidden from them. Other classification proposals come from Meekel *et al.* [81] (feature, hardware platform, performances and attributes variability) or Bass *et al.* [49] who discusses about variability at the architectural level.

Central to the modeling of variability is the notion of *feature*, originally defined by Kang *et al.* as: *a prominent or distinctive user-visible aspect, quality or characteristic of a software system or systems* [77]. Based on this notion of *feature*, they proposed to use a *feature model* to model the variability in a SPL. A feature model consists of a *feature diagram* and other associated information: *constraints* and *dependency rules*. Feature diagrams provide a *graphical tree-like notation depicting the hierarchical organization of high level product functionalities* represented as features. The root of the tree refers to the complete system and is progressively decomposed into more refined features (tree nodes). Relations between nodes (features) are materialized by *decomposition edges* and *textual constraints*. Variability can be expressed in several ways. Presence or absence of a feature from a product is modeled using *mandatory* or *optional features*. Features are graphically represented as rectangles while some graphical elements (e.g., unfilled circle) are used to describe the variability (e.g., a feature may be optional).

Features can be organized into *feature groups*. Boolean operators *exclusive alternative (XOR)*, *inclusive alternative (OR)* or *inclusive (AND)* are used to select one, several or all the features from a feature group.

Dependencies between features can be modeled using *textual constraints*: *requires* (presence of a feature requires the presence of another), *mutex* (presence of a feature automatically excludes another). Feature attributes can be also used for modeling quantitative (e.g., numerical) information. Constraints over attributes and features can be specified as well.

Modeling variability allows an organization to capture and select which version of which variant of any particular aspect is wanted in the system [58]. To implement it cheaply, quickly and safely, redoing by hand the tedious weaving of every aspect is not an option: some form of automation is needed to leverage the modeling of variability [53], [65]. Model Driven Engineering (MDE) makes it possible to automate this weaving process [76]. This requires that models are no longer informal, and that the weaving process is itself described as a program (which is as a matter of facts an executable meta-model [84]) manipulating these models to produce for instance a detailed design that can ultimately be transformed to code, or to test suites [93], or other software artifacts.

3.1.3. Component-based software development

Component-based software development [103] aims at providing reliable software architectures with a low cost of design. Components are now used routinely in many domains of software system designs: distributed systems, user interaction, product lines, embedded systems, etc. With respect to more traditional software artifacts (e.g., object oriented architectures), modern component models have the following distinctive features [64]: description of requirements on services required from the other components; indirect connections between components thanks to ports and connectors constructs [79]; hierarchical definition of components (assemblies of components can define new component types); connectors supporting various communication semantics [61]; quantitative properties on the services [56].

In recent years component-based architectures have evolved from static designs to dynamic, adaptive designs (e.g., SOFA [61], Palladio [54], Frascati [85]). Processes for building a system using a statically designed architecture are made of the following sequential lifecycle stages: requirements, modeling, implementation, packaging, deployment, system launch, system execution, system shutdown and system removal. If for any reason after design time architectural changes are needed after system launch (e.g., because requirements changed, or the implementation platform has evolved, etc) then the design process must be reexecuted from scratch (unless the changes are limited to parameter adjustment in the components deployed).

Dynamic designs allow for *on the fly* redesign of a component based system. A process for dynamic adaptation is able to reapply the design phases while the system is up and running, without stopping it (this is different from a stop/redeploy/start process). Dynamic adaptation process supports *chosen adaptation*, when changes are planned and realized to maintain a good fit between the needs that the system must support and the way it supports them [78]. Dynamic component-based designs rely on a component meta-model that supports complex life cycles for components, connectors, service specification, etc. Advanced dynamic designs can also take platform changes into account at run-time, without human intervention, by adapting themselves [62], [106]. Platform changes and more generally environmental changes trigger *imposed adaptation*, when the system can no longer use its design to provide the services it must support. In order to support an eternal system [55], dynamic component based systems must separate architectural design and platform compatibility. This requires support for heterogeneity, since platform evolutions can be partial.

The Models@runtime paradigm denotes a model-driven approach aiming at taming the complexity of dynamic software systems. It basically pushes the idea of reflection one step further by considering the reflection layer as a real model “something simpler, safer or cheaper than reality to avoid the complexity, danger and irreversibility of reality [97]”. In practice, component-based (and/or service-based) platforms offer reflection APIs that make it possible to introspect the system (which components and bindings are currently in place in the system) and dynamic adaptation (by applying CRUD operations on these components and bindings). While some of these platforms offer rollback mechanisms to recover after an erroneous adaptation, the idea of Models@runtime is to prevent the system from actually enacting an erroneous adaptation. In other words, the “model at run-time” is a reflection model that can be uncoupled (for reasoning, validation, simulation purposes) and automatically resynchronized.

Heterogeneity is a key challenge for modern component based system. Until recently, component based techniques were designed to address a specific domain, such as embedded software for command and control, or distributed Web based service oriented architectures. The emergence of the Internet of Things paradigm calls for a unified approach in component based design techniques. By implementing an efficient separation of concern between platform independent architecture management and platform dependent implementations, *Models@runtime* is now established as a key technique to support dynamic component based designs. It provides DIVERSE with an essential foundation to explore an adaptation envelop at run-time.

Search Based Software Engineering [73] has been applied to various software engineering problems in order to support software developers in their daily work. The goal is to automatically explore a set of alternatives and assess their relevance with respect to the considered problem. These techniques have been applied to craft software architecture exhibiting high quality of services properties [70]. Multi Objectives Search based techniques [67] deal with optimization problem containing several (possibly conflicting) dimensions to optimize. These techniques provide DIVERSE with the scientific foundations for reasoning and efficiently exploring an envelope of software configurations at run-time.

3.1.4. Validation and verification

Validation and verification (V&V) theories and techniques provide the means to assess the validity of a software system with respect to a specific correctness envelop. As such, they form an essential element of DIVERSE's scientific background. In particular, we focus on model-based V&V in order to leverage the different models that specify the envelop at different moments of the software development lifecycle.

Model-based testing consists in analyzing a formal model of a system (*e.g.*, activity diagrams, which capture high-level requirements about the system, statecharts, which capture the expected behavior of a software module, or a feature model, which describes all possible variants of the system) in order to generate test cases that will be executed against the system. Model-based testing [105] mainly relies on model analysis, constraint solving [66] and search-based reasoning [80]. DIVERSE leverages in particular the applications of model-based testing in the context of highly-configurable systems and [107] interactive systems [82] as well as recent advances based on diversity for test cases selection [74].

Nowadays, it is possible to simulate various kinds of models. Existing tools range from industrial tools such as Simulink, Rhapsody or Telelogic to academic approaches like Omega [91], or Xholon [0]. All these simulation environments operate on homogeneous environment models. However, to handle diversity in software systems, we also leverage recent advances in heterogeneous simulation. Ptolemy [60] proposes a common abstract syntax, which represents the description of the model structure. These elements can be decorated using different directors that reflect the application of a specific model of computation on the model element. Metropolis [50] provides modeling elements amenable to semantically equivalent mathematical models. Metropolis offers a precise semantics flexible enough to support different models of computation. ModHel'X [72] studies the composition of multi-paradigm models relying on different models of computation.

Model-based testing and simulation are complemented by runtime fault-tolerance through the automatic generation of software variants that can run in parallel, to tackle the open nature of software-intensive systems. The foundations in this case are the seminal work about N-version programming [48], recovery blocks [95] and code randomization [52], which demonstrated the central role of diversity in software to ensure runtime resilience of complex systems. Such techniques rely on truly diverse software solutions in order to provide systems with the ability to react to events, which could not be predicted at design time and checked through testing or simulation.

3.1.5. Empirical software engineering

The rigorous, scientific evaluation of DIVERSE's contributions is an essential aspect of our research methodology. In addition to theoretical validation through formal analysis or complexity estimation, we also aim at applying state-of-the-art methodologies and principles of empirical software engineering. This approach encompasses a set of techniques for the sound validation contributions in the field of software engineering,

[0] http://www.primordion.com/Xholon/

ranging from statistically sound comparisons of techniques and large-scale data analysis to interviews and systematic literature reviews [100], [98]. Such methods have been used for example to understand the impact of new software development paradigms [59]. Experimental design and statistical tests represent another major aspect of empirical software engineering. Addressing large-scale software engineering problems often requires the application of heuristics, and it is important to understand their effects through sound statistical analyses [47].

3.2. Research axis

Figure 1 illustrates the four dimensions of software diversity, which form the core research axis of DIVERSE: the **diversity of languages** used by the stakeholders involved in the construction of these systems; the **diversity of features** required by the different customers; the **diversity of runtime environments** in which software has to run and adapt; the **diversity of implementations** that are necessary for resilience through redundancy. These four axes share and leverage the scientific and technological results developed in the area of model-driven engineering in the last decade. This means that all our research activities are founded on sound abstractions to reason about specific aspects of software systems, compose different perspectives and automatically generate parts of the system.

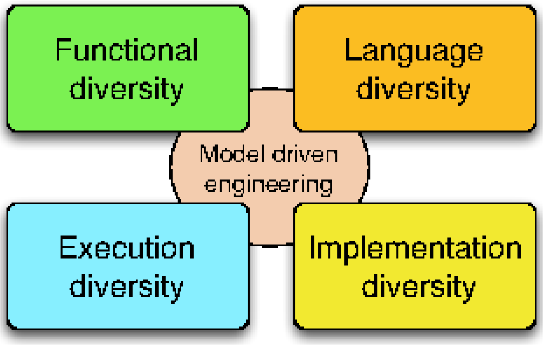

Figure 1. The four research axes of DIVERSE, which rely on a MDE scientific background

3.2.1. Software Language Engineering

The engineering of systems involves many different stakeholders, each with their own domain of expertise. Hence more and more organizations are adopting Domain Specific Modeling Languages (DSMLs) to allow domain experts to express solutions directly in terms of relevant domain concepts [99], [69]. This new trend raises new challenges about designing DSMLs, evolving a set of DSMLs and coordinating the use of multiple DSLs for both DSL designers and DSL users.

3.2.1.1. Challenges

Reusability of software artifacts is a central notion that has been thoroughly studied and used by both academics and industrials since the early days of software construction. Essentially, designing reusable artifacts allows the construction of large systems from smaller parts that have been separately developed and validated, thus reducing the development costs by capitalizing on previous engineering efforts. However, it is still hardly possible for language designers to design typical language artifacts (e.g. language constructs, grammars, editors or compilers) in a reusable way. The current state of the practice usually prevents the reusability of language artifacts from one language to another, consequently hindering the emergence of real engineering techniques around software languages. Conversely, concepts and mechanisms that enable artifacts reusability abound in the software engineering community.

Variability in modeling languages occur in the definition of the abstract and concrete syntax as well as in the specification of the language's semantics. The major challenges met when addressing the need for variability are: (i) to set principles for modeling language units that support the modular specification of a modeling language; and (ii) to design mechanisms to assemble these units into a complete language, according to the set of authorized variation points for the modeling language family.

A new generation of complex software-intensive systems (for example smart health support, smart grid, building energy management, and intelligent transportation systems) gives new opportunities for leveraging modeling languages. The development of these systems requires expertise in diverse domains. Consequently, different types of stakeholders (e.g., scientists, engineers and end-users) must work in a coordinated manner on various aspects of the system across multiple development phases. DSMLs can be used to support the work of domain experts who focus on a specific system aspect, but they can also provide the means for coordinating work across teams specializing in different aspects and across development phases. The support and integration of DSMLs leads to what we call **the globalization of modeling languages**, *i.e.* the use of multiple languages for the coordinated development of diverse aspects of a system. One can make an analogy with world globalization in which relationships are established between sovereign countries to regulate interactions (e.g., travel and commerce related interactions) while preserving each country's independent existence.

3.2.1.2. Scientific objectives

We address reuse and variability challenges through the investigation of the time-honored concepts of substitutability, inheritance and components, evaluate their relevance for language designers and provide tools and methods for their inclusion in software language engineering. We will develop novel techniques for the modular construction of language extensions with support to model syntactical variability. From the semantics perspective, we investigate extension mechanisms for the specification of variability in operational semantics, focusing on static introduction and heterogeneous models of computation. The definition of variation points for the three aspects of the language definition provides the foundations for the novel concept Language Unit (LU) as well as suitable mechanisms to compose such units.

We explore the necessary breakthrough in software languages to support modeling and simulation of heterogeneous and open systems. This work relies on the specification of executable domain specific modeling languages (DSMLs) to formalize the various concerns of a software-intensive system, and of models of computation (MoCs) to explicitly model the concurrency, time and communication of such DSMLs. We develop a framework that integrates the necessary foundations and facilities for designing and implementing executable and concurrent domain-specific modeling languages. It also provides unique features to specify composition operators between (possibly heterogeneous) DSMLs. Such specifications are amenable to support the edition, execution, graphical animation and analysis of heterogeneous models. The objective is to provide both a significant improvement to MoCs and DSMLs design and implementation and to the simulation based validation and verification of complex systems.

We see an opportunity for the automatic diversification of programs' computation semantics, for example through the diversification of compilers or virtual machines. The main impact of this artificial diversity is to provide flexible computation and thus ease adaptation to different execution conditions. A combination of static and dynamic analysis could support the identification of what we call *plastic computation zones* in the code. We identify different categories of such zones: (i) areas in the code in which the order of computation can vary (e.g., the order in which a block of sequential statements is executed); (ii) areas that can be removed, keeping the essential functionality [101] (e.g., skip some loop iterations); (iii) areas that can replaced by alternative code (e.g., replace a try-catch by a return statement). Once we know which zones in the code can be randomized, it is necessary to modify the model of computation to leverage the computation plasticity. This consists in introducing variation points in the interpreter to reflect the diversity of models of computation. Then, the choice of a given variation is performed randomly at run time.

3.2.2. Variability Modeling and Engineering

The systematic modeling of variability in software systems has emerged as an effective approach to document and reason about software evolutions and heterogeneity (*cf.* Section 3.1.2). Variability modeling characterizes an “envelope” of possible software variations. The industrial use of variability models and their relation to software artifact models require a complete engineering framework, including composition, decomposition, analysis, configuration and artifact derivation, refactoring, re-engineering, extraction, and testing. This framework can be used both to tame imposed diversity and to manage chosen diversity.

3.2.2.1. Challenges

A fundamental problem is that the **number of variants** can be exponential in the number of options (features). Already with 300 boolean configuration options, approximately 10^{90} configurations exist – more than the estimated count of atoms in the universe. Domains like automotive or operating systems have to manage more than 10000 options (e.g., Linux). Practitioners face the challenge of developing billions of variants. It is easy to forget a necessary constraint, leading to the synthesis of unsafe variants, or to under-approximate the capabilities of the software platform. Scalable modelling techniques are therefore crucial to specify and reason about a very large set of variants.

Model-driven development supports two approaches to deal with the increasing number of concerns in complex systems: multi-view modeling, *i.e.* when modeling each concern separately, and variability modeling. However, there is little support to combine both approaches consistently. Techniques to integrate both approaches will enable the construction of a consistent set of views and variation points in each view.

The design, construction and maintenance of software families have a major impact on **software testing**. Among the existing challenges, we can cite: the selection of test cases for a specific variant; the evolution of test suites with integration of new variants; the combinatorial explosion of the number of software configurations to be tested. Novel model-based techniques for test generation and test management in a software product line context are needed to overcome state-of-the-art limits we already observed in some projects.

3.2.2.2. Scientific objectives

We aim at developing scalable reasoning techniques to **automatically analyze** variability models and their interactions with other views on the software intensive system (requirements, architecture, design, code). These techniques provide two major advancements in the state of the art: (1) an extension of the semantics of variability models in order to enable the definition of attributes (*e.g.*, cost, quality of service, effort) on features and to include these attributes in the reasoning; (2) an assessment of the consistent specification of variability models with respect to system views (since variability is orthogonal to system modeling, it is currently possible to specify the different models in ways that are semantically meaningless). The former aspect of analysis is tackled through constraint solving and finite-domain constraint programming, while the latter aspect is investigated through automatic search-based and learning-based techniques for the exploration of the space of interaction between variability and view models.

We want to develop procedures to **reverse engineer** dependencies and features’ sets from existing software artefacts – be it source code, configuration files, spreadsheets (e.g., product comparison matrices) or requirements. We expect to scale up (e.g., for extracting a very large number of variation points) and guarantee some properties (e.g., soundness of configuration semantics, understandability of ontological semantics). For instance, when building complex software-intensive systems, textual requirements are captured in very large quantities of documents. In this context, adequate models to formalize the organization of requirements documents and automated techniques to support impact analysis (in case of changes in the requirements) have to be developed.

3.2.3. Heterogeneous and dynamic software architectures

Flexible yet dependable systems have to cope with heterogeneous hardware execution platforms ranging from smart sensors to huge computation infrastructures and data centers. Evolutions range from a mere change in the system configuration to a major architectural redesign, for instance to support addition of new features or a change in the platform architecture (new hardware is made available, a running system switches to low bandwidth wireless communication, a computation node battery is running low, etc). In this context, we need to devise formalisms to reason about the impact of an evolution and about the transition from one configuration to another. It must be noted that this axis focuses on the use of models to drive the evolution from design time to run-time. Models will be used to (i) systematically define predictable configurations and variation points through which the system will evolve; (ii) develop behaviors necessary to handle unpredicted evolutions.

3.2.3.1. Challenges

The main challenge is to provide new homogeneous architectural modelling languages and efficient techniques that enable continuous software reconfiguration to react to changes. This work handles the challenges of handling the diversity of runtime infrastructures and managing the cooperation between different stakeholders. More specifically, the research developed in this axis targets the following dimensions of software diversity.

Platform architectural heterogeneity induces a first dimension of imposed diversity (type diversity). Platform reconfigurations driven by changing resources define another dimension of diversity (deployment diversity). To deal with these imposed diversity problems, we will rely on model based runtime support for adaptation, in the spirit of the dynamic distributed component framework developed by the Triskell team. Since the runtime environment composed of distributed, resource constrained hardware nodes cannot afford the overhead of traditional runtime adaptation techniques, we investigate the design of novel solutions relying on models@runtime and on specialized tiny virtual machines to offer resource provisioning and dynamic reconfigurations.

Diversity can also be an asset to optimize software architecture. Architecture models must integrate multiple concerns in order to properly manage the deployment of software components over a physical platform. However, these concerns can contradict each other (*e.g.*, accuracy and energy). In this context, we investigate automatic solutions to explore the set of possible architecture models and to establish valid trade-offs between all concerns in case of changes.

3.2.3.2. Scientific objectives

Automatic synthesis of optimal software architectures. Implementing a service over a distributed platform (*e.g.*, a pervasive system or a cloud platform) consists in deploying multiple software components over distributed computation nodes. We aim at designing search-based solutions to (i) assist the software architect in establishing a good initial architecture (that balances between different factors such as cost of the nodes, latency, fault tolerance) and to automatically update the architecture when the environment or the system itself change. The choice of search-based techniques is motivated by the very large number of possible software deployment architectures that can be investigated and that all provide different trade-offs between qualitative factors. Another essential aspect that is supported by multi-objective search is to explore different architectural solutions that are not necessarily comparable. This is important when the qualitative factors are orthogonal to each other, such as security and usability for example.

Flexible software architecture for testing and data management. As the number of platforms on which software runs increases and different software versions coexist, the demand for testing environments also increases. For example, the number of testing environments to test a software patch or upgrade is the product of the number of running environments the software supports and the number of coexisting versions of the software. Based on our first experiment on the synthesis of cloud environment using architectural models, our objective is to define a set of domain specific languages to catch the requirement and to design cloud environments for testing and data management of future internet systems from data centers to things. These languages will be interpreted to support dynamic synthesis and reconfiguration of a testing environment.

Runtime support for heterogeneous environments. Execution environments must provide a way to account or reserve resources for applications. However, current execution environments such as the Java Virtual Machine do not clearly define a notion of application: each framework has its own definition. For example, in OSGi, an application is a component, in JEE, an application is most of the time associated to a class loader, in the Multi-Tasking Virtual machine, an application is a process. The challenge consists in defining an execution environment that provides direct control over resources (CPU, Memory, Network I/O) independently from the definition of an application. We propose to define abstract resource containers to account and reserve resources on a distributed network of heterogeneous devices.

3.2.4. Diverse implementations for resilience

Open software-intensive systems have to evolve over their lifetime in response to changes in their environment. Yet, most verification techniques assume a closed environment or the ability to predict all changes. Dynamic changes and evolutions thus represent a major challenge for these techniques that aim at assessing the correctness and robustness of the system. On the one hand, DIVERSE will adapt V&V techniques to handle diversity imposed by the requirements and the execution environment, on the other hand we leverage diversity to increase the robustness of software in face of unpredicted situations. More specifically, we address the following V&V challenges.

3.2.4.1. Challenges

One major challenge to build flexible and open yet dependable systems is that current software engineering techniques require architects to foresee all possible situations the system will have to face. However, openness and flexibility also mean unpredictability: unpredictable bugs, attacks, environmental evolutions, etc. Current fault-tolerance [95] and security [68] techniques provide software systems with the capacity of detecting accidental and deliberate faults. However, existing solutions assume that the set of bugs or vulnerabilities in a system does not evolve. This assumption does not hold for open systems, thus it is essential to revisit fault-tolerance and security solutions to account for diverse and unpredictable faults.

Diversity is known to be a major asset for the robustness of large, open, and complex systems (*e.g.*, economical or ecological systems). Following this observation, the software engineering literature provides a rich set of work that choose to implement diversity in software systems in order to improve robustness to attacks or to changes in quality of service. These works range from N-version programming to obfuscation of data structures or control flow, to randomization of instruction sets. An essential remaining challenge is to support the automatic synthesis and evolution of software diversity in open software-intensive systems. There is an opportunity to further enhance these techniques in order to cope with a wider diversity of faults, by multiplying the levels of diversity in the different software layers that are found in software-intensive systems (system, libraries, frameworks, application). This increased diversity must be based on artificial program transformations and code synthesis, which increase the chances of exploring novel solutions, better fitted at one point in time. The biological analogy also indicates that diversity should emerge as a side-effect of evolution, to prevent over-specialization towards one kind of diversity.

3.2.4.2. Scientific objectives

The main objective is to address one of the main limitations of N-version programming for fault-tolerant systems: the manual production and management of software diversity. Through automated injection of artificial diversity we aim at systematically increasing failure diversity and thus increasing the chances of early error detection at run-time. A fundamental assumption for this work is that software-intensive systems can be "good enough" [96], [108].

Proactive program diversification. We aim at establishing novel principles and techniques that favor the emergence of multiple forms of software diversity in software-intensive systems, in conjunction with the software adaptation mechanisms that leverage this diversity. The main expected outcome is a set of meta-design principles that maintain diversity in systems and the experimental demonstration of the effects of software diversity on the adaptive capacities of CASs. Higher levels of diversity in the system provide a pool of software solutions that can eventually be used to adapt to situations unforeseen at design time (bugs, crash, attacks, etc.). Principles of automated software diversification rely on the automated synthesis of variants in a software product line, as well as finer-grained program synthesis combining unsound transformations and genetic programming to explore the space of mutational robustness.

Multi-tier software diversification. We call multi-tier diversification the fact of diversifying several application software components simultaneously. The novelty of our proposal, with respect to the software diversity state of the art, is to diversify the application-level code (for example, diversify the business logics of the application), focusing on the technical layers found in web applications. The diversification of application software code is expected to provide a diversity of failures and vulnerabilities in web server deployment. Web server deployment usually adopts a form of the Reactor architecture pattern, for scalability purposes: multiple copies of the server software stack, called request handlers, are deployed behind a load balancer. This architecture is very favorable for diversification, since by using the multiplicity of request handlers running in a web server we can simultaneously deploy multiple combinations of diverse software components. Then, if one handler is hacked or crashes the others should still be able to process client requests.

EASE Team

3. Research Program

3.1. Collecting pertinent information

In our model, applications adapt their behavior (for instance, the level of automation) to the quality of their perception of the environment. This is important to alleviate the development constraint we usually have on automated system. We "just" have to be sure a given process will always operate at the right automation level given the precision, the completeness or the confidence it has on its own perception. For instance, a car passing through a cross would choose its speed depending on the confidence it has gained during perception data gathering. When it has not enough information or when it could not trust it, it should reduce the automation level, therefore the speed, to only rely on its own sensors. Such adaptation capability shift requirements from the design and deployment (availability, robustness, accuracy, etc.) to the **assessment of the environment perception** we aim to facilitate in this first research axis.

Data characterization. The quality (freshness, accuracy, confidence, reliability, confidentiality, etc.) of the data are of crucial importance to assess the quality of the perception and therefore to ensure proper behavior. The way data is produced, consolidated, and aggregated while flowing to the consumer has an impact on its quality. Moreover part of these quality attributes requires to gather information at several communication layers from various entities. For this purpose, we want to design **lightweight cross-layer interactions** to collect relevant data. As a "frugality" principle should guide our approach, it is not appropriate to build all attributes we can imagine. It is therefore necessary to identify attributes relevant to the application and to have mechanisms to activate/deactivate at run-time the process to collect them.

Data fusion. Raw data should be directly used only to determine low-level abstraction. Further help in abstracting from low-level details can be provided by **data fusion** mechanisms. A good (re)construction of a meaningful information for the application reduces the complexity of the pervasive applications and helps the developers to concentrate on the application logic rather on the management of raw data. Moreover, the reactivity required in pervasive systems and the aggregation of large amounts of data (and its processing) are antagonists. We study **software services that can be deployed closer to the edge of the network**. The exploration of data fusion technics will be guided by different criteria: relevance of abstractions produced for pervasive applications, anonymization of exploited raw data, processing time, etc.

Assessing the correctness of the behavior. To ease the design of new applications and to align the development of new products with the ever faster standard developments, continuous integration could be used in parallel with continuous conformance and interoperability testing. We already participate in the design of new shared platforms that aims at facilitating this providing remote testing tools. Unfortunately, it is not possible to be sure that all potential peers in the surrounding have a conform behavior. Moreover, upon failure or security breach, a piece of equipment could stop to operate properly and lead to global mis-behavior. We want to propose conceptual tools for **testing at runtime devices in the environment**. The result of such conformance or interoperability tests could be stored safely in the environment by authoritative testing entity. Then application could interact with the device with a higher confidence. The confidence level of a device could be part of the quality attribute of the information it contributed to generate. The same set of tools could be used to identify misbehaving device for maintenance purpose or to trigger further testing.

3.2. Building relevant abstraction for new interactions

The pervasive applications are often designed in an ad hoc manner depending on the targeted application area. Ressources (sensors / actuators, connected objets etc.) are often used in silos which complexify the implementation of rich pervasive computing scenarios. In the second research axis, we want to get away from technical aspects identifying **common and reusable system mechanisms** that could be used in various applications.

Tagging the environment. Information relative to environment could be stored by the application itself, but it could be complex to manage for mobile application since it could cross a large number of places with various features. Moreover the developer has to build its own representation of information especially when he wants to share information with other instances of the same application or with other applications. A promising approach is to store and to maintain this information associated to an object or to a place, in the environment itself. The infrastructure should provide services to application developers: add/retrieve information in the environment, share information and control who can access it, add computed properties to object for further usage. We want to study an **extensible model to describe and augment the environment**. Beyond a simple distributed storage, we have in mind a new kind of interaction between pervasive applications and changing environment and between applications themselves.

Taking advantages of the spatial relationships. To understand the world they have to interact with, pervasive applications often have to (re)built a model of it from the exchange they have with others or from their own observations. A part of the programmer's task consists in building a model of the spatial layout of the objects in the surrounding. The term *layout* can be understood in several ways: the co-location of multiple objects in the same vicinity, the physical arrangement of two objects relative to each other, or even the crossing of an object of a physical area to another, etc. Determining remotely these spatial properties (see figure 1 -a) is difficult without exchanging a lot of information. Properties related to the spatial layout are far easier to characterize locally. They could be abstracted from interaction pattern without any complex virtual representation of the environment (see figure 1 -b). We want to be able to rely on this type of spatial layout in a pervasive environment. In the prior years, the members of EASE already worked on **models for processing object interactions** in the physical world to automatically trigger processing. This was the case in particular of the spatial programming principle: physical space is treated as a tuple-space in which objects are automatically synchronized according to their spatial arrangement. We want to follow this approach by considering **richer and more expressive programming models.**

3.3. Acting on the environment

The conceptual tools we aim to study must be *frugal*: they use as less as possible resources, while having the possibility to use much more when it is required. Data needed by an application are not made available for "free"; for example, it costs energy to measure a characteristic of the environment, or to transmit it. So this "design frugality" requires **a fine-grained control** on how data is actually collected from the environment. The third research axis aims at designing solutions that give this control to application developers by **acting on the environment**.

Acting on the data collection. We want to be able to identify which information are really needed during the perception elaboration process. If a piece of data is missing to build a given information with the appropriate quality level, the data collection mechanism should find relevant information in the environment or modify the way it aggregates it. These could lead to a modification of the behavior of the network layer and the path the piece of data uses in the aggregation process.

Acting on object interactions. Objects in the environment could adapt their behavior in a way that strongly depends on the object itself and that is difficult to generalize. Beyond the specific behaviors of actuators triggered through specialized or standard interfaces, the production of information required by an application could necessitate an adaptation at the object level (eg. calibration, sampling). The environment should then be able to initiate such adaption transparently to the application, which may not know all objects it passes by.

Adapting object behaviors. The radio communication layers become more flexible and able to adapt the way they use energy to what is really required for a given transmission. We already study how beamforming technics could be used to adapt multicast strategy for video services. We want to show how playing with these new parameters of transmissions (eg. beamforming, power, ...) allows to control spatial relationships objects could have. There is a tradeoff to find between the capacity of the medium, the electromagnetic pollution and the reactivity of the environment. We plan to expend our previous on interface selection and more generally on what we call **opportunistic networking**.

KERDATA Project-Team

3. Research Program

3.1. Research axis 1: Convergence of Extreme-Scale Computing and Big Data Infrastructures

The tools and cultures of High Performance Computing and Big Data Analytics have evolved in divergent ways. This is to the detriment of both. However, big computations still generate and are needed to analyze Big Data. As scientific research increasingly depends on both high-speed computing and data analytics, the potential interoperability and scaling convergence of these two eco-systems is crucial to the future. Our objective for the next years is premised on the idea that we must begin to systematically map out and account for the ways in which the major issues associated with Big Data intersect with, impinge upon, and potentially change the plans that are now being laid for achieving Exascale computing.

Collaboration. *This axis is addressed in close collaboration with María Pérez (UPM), Rob Ross (ANL), Toni Cortes (BSC), Bogdan Nicolae (formerly at IBM Research, now at Huawei Research).*

Relevant groups with similar interests are the following ones.

- *The group of Jack Dongarra, Innovative Computing Laboratory at University of Tennessee/Oak Ridge National Laboratory, working on joint tools Exascale Computing and Big Data.*
- *The group of Satoshi Matsuoka, Tokyo Institute of Technology, working on system software for Clouds and HPC.*
- *The group of Franck Cappello at Argonne National Laboratory/NCSA working on on-demand data analytics and storage for extreme-scale simulations and experiments.*

3.1.1. High-performance storage for concurrent Big Data applications

We argue that storage is a plausible pathway to convergence. In this context, we plan to focus on the needs of concurrent Big Data applications that require high-performance storage, as well as transaction support. Although blobs (binary large objects) are an increasingly popular storage model for such applications, state-of-the-art blob storage systems offer no transaction semantics. This demands users to coordinate data access carefully in order to avoid race conditions, inconsistent writes, overwrites and other problems that cause erratic behavior.

We argue there is a gap between existing storage solutions and application requirements, which limits the design of transaction-oriented applications. In this context, one idea on which we plan to focus our efforts is exploring how blob storage systems could provide built-in, multi-blob transactions, while retaining sequential consistency and high throughput under heavy access concurrency.

The early principles of this research direction have already raised interest from our partners at ANL (Rob Ross) and UPM (María Pérez) for potential collaborations. In this direction, the acceptance of our paper on the Týr transactional blob storage system as a Best Student Paper Award Finalist at the SC16 conference [10] is a very encouraging step.

3.1.2. Big Data analytics on Exascale HPC machines

Big Data analytics is another interesting direction that we plan to explore, building on top of these converged storage architectures. Specifically, we will examine the ways in which Exascale infrastructures can be leveraged not only by HPC-centric, but also by scientific, cloud-centric applications. Many of the current state-of-the-art Big Data processing approaches, including Hadoop and Spark [43] are optimized to run on commodity machines. This impacts the mechanisms used to deal with failures and the limited network bandwidth.

A blind adoption of these systems on extreme-scale platforms would result in high overheads. It would therefore prevent users from fully benefiting from the high performance infrastructure. The objective that we set here is to explore design and implementation options for new data analytics systems that can exploit the features of extreme-scale HPC machines: multi-core nodes, multiple memory and storage technologies including a large memory, NVRAM, SSDs, etc.

3.2. Research axis 2: Advanced stream data processing

The recent evolutions in the area of Big Data processing have pointed out some limitations of the initial Map-Reduce model. It is well suited for batch data processing, but less suited for real-time processing of dynamic data streams. New types of data-intensive applications emerge, e.g., for enterprises who need to perform analysis on their stream data in ways that can give fast results (i.e., in real time) at scale (e.g., click-stream analysis and network-monitoring log analysis). Similarly, scientists require fast and accurate data processing techniques in order to analyze their experimental data correctly at scale (e.g., collectively analysis of large data sets distributed in multiple geographically distributed locations).

Our plan is to revisit current data management techniques to cope with the volatile requirements of data-intensive applications on large-scale dynamic clouds in a cost-efficient way.

Collaboration. *This axis is addressed in close collaboration with María Pérez (UPM), Kate Keahey (ANL) and Toni Cortes (BSC).*

Relevant groups with similar interests include the following ones.

- *The AMPLab, UC Berkeley, USA, working on scheduling stream data applications in heterogeneous clouds.*
- *The group of Ewa Deelman, USC Information Sciences Institute, working on resource management for workflows in Clouds.*
- *The XTRA group, Nanyang Technological University, Singapore, working on resource provisioning for workflows in the cloud.*

3.2.1. Stream-oriented, Big Data processing on clouds

The state-of-the-art Hadoop Map-Reduce framework cannot deal with stream data applications, as it requires the data to be initially stored in a distributed file system in order to process them. To better cope with the above-mentioned requirements, several systems have been introduced for stream data processing such as Flink [38], Spark [43], Storm [44], and Google MillWheel [46]. These systems keep computation in memory to decrease latency, and preserve scalability by using data-partitioning or dividing the streams into a set of deterministic batch computations.

However, they are designed to work in dedicated environments and they do not consider the performance variability (i.e., network, I/O, etc.) caused by resource contention in the cloud. This variability may in turn cause high and unpredictable latency when output streams are transmitted to further analysis. Moreover, they overlook the dynamic nature of data streams and the volatility in their computation requirements. Finally, they still address failures in a best-effort manner.

Our objective is to investigate new approaches for reliable, stream Big Data processing on clouds. We will explore new mechanisms that expose resource heterogeneity (observed variability in resource utilization at runtime) when scheduling stream data applications. We will also investigate how to adapt to node failures automatically, and to adapt the failure handling techniques to the characteristics of the running application and to the root cause of failures.

3.2.2. Efficient Edge, Cloud and hybrid Edge/Cloud data processing

Today, we are approaching an important technological milestone: applications are generating huge amounts of data and are demanding low-latency responses to their requests. Mobile computing and Internet of Things (IoT) applications are good illustrations of such scenarios. Using only Cloud computing for such scenarios is challenging. Firstly, Cloud resources are most of the time accessed through Internet, hence, data are sent across high-latency wide area networks, which may degrade the performance of applications. Secondly, it may be impossible to send data to the Cloud due to data regulations, national security laws or simply because an Internet connection is not available. Finally, data transmission costs (e.g., Cloud provider fees, carrier costs) could make a business solution impractical.

Edge computing is a new paradigm which aims to address some of these issues. The key idea is to leverage computing and storage resources at the "edge" of the network, i.e., on processing units located close to the data sources. This allows applications to outsource task execution from the main (Cloud) processing data centers to the edge. The development of Edge computing was accelerated by the recent emergence of stream processing, a new model for handling continuous flows of data in real-time, as opposed to batch processing, which typically processes bounded datasets offline.

However, Edge computing is not a silver bullet. Besides being a new concept not fully established in the community, issues like node volatility, limited processing power, high latency between nodes, fault tolerance and data degradation may impact applications depending on the characteristics of the infrastructure. An important question raised in this context is "how close is physically close enough?". In other words, how much can one improve (or degrade) the performance of an application by performing computation closer to the data sources?

Our objective is to try to answer precisely this question. We are interested in understanding the conditions that enable the usage of Edge or Cloud computing to reduce the time to results and the associated costs. While some state-of- the-art approaches advocate either "100% Cloud" or "100% Edge" solutions, the relative efficiency of a method over the other may vary. Intuitively, it depends on many parameters, including network technology, hardware characteristics, volume of data or computing power, processing framework configuration and application requirements, to cite a few. We plan to study their impact on the overall application performance.

3.3. Research axis 3: I/O management, in situ visualization and analysis on HPC systems at extreme scales

Over the past few years, the increasing amounts of data produced by large-scale simulations have motivated a shift from traditional offline data analysis to in situ analysis and visualization. In situ processing started by coupling a parallel simulation with an analysis or visualization library, to avoid the cost of writing data on storage and reading it back. Going beyond this simple pairwise tight coupling, complex analysis workflows today are graphs with one or more data sources and several interconnected analysis components.

Collaboration. *This axis is worked out in close collaboration with Rob Ross (ANL), Tom Peterka (ANL), Matthieu Dorier (ANL), Toni Cortes (BSC), Bruno Raffin (Inria). Some additional collaborations are in discussion with other members of JLESC, and with CEA and Total.*

Relevant groups with similar interests include the following ones.

- *The group of Manish Parashar at Rutgers University, USA (I/O management for HPC systems, in situ processing).*
- *The group of Scott Klasky at Oak Ridge National Lab, USA (I/O management for HPC systems, in situ processing).*
- *The CNRS IPSL laboratory (Sébastien Denvil, Pôle de modélisation du climat) in Paris, France (in situ data analytics).*

3.3.1. Toward a joint optimized architecture for in situ visualization and advanced processing

From Inria and ANL, four tools at least have emerged to address some challenges of coupling simulations with visualization packages or analysis workflows. Each of them focused on some particular aspect:

Damaris (Inria, [5], [4]) exploits dedicated cores to enable jitter-free I/O and in situ visualization;

Decaf (ANL, [33]) implements a coupling service for workflows;

FlowVR (Inria, [45]) connects workflow components for in situ processing;

Swift (ANL, [48]) focuses on implicitly parallel data flows and was optimized for Big Data processing.

Our plan is to explore how these tools could best leverage their respective strengths in a *joint optimized architecture for in situ visualization and advanced processing* in the HPC area. We published a preliminary study describing the lessons learned from using these tools in production environments with real applications [7]. Such a joint architecture will contribute to address the data volume and velocity challenges raised by data-intensive workflows, including complex data-intensive analytics phases. It may also impact, in a subsequent step, future data analysis pipelines for converged Big Data and HPC architectures.

MYRIADS Project-Team

3. Research Program

3.1. Introduction

In this section, we present our research challenges along four work directions: resource and application management in distributed cloud and fog computing architectures for scaling clouds in Section 3.2 , energy management strategies for greening clouds in Section 3.3 , security and data protection aspects for securing cloud-based information systems and applications in Section 3.4 , and methods for experimenting with clouds in Section 3.5 .

3.2. Scaling fogs and clouds

3.2.1. Resource management in hierarchical clouds

The next generation of utility computing appears to be an evolution from highly centralized clouds towards more decentralized platforms. Today, cloud computing platforms mostly rely on large data centers servicing a multitude of clients from the edge of the Internet. Servicing cloud clients in this manner suggests that locality patterns are ignored: wherever the client issues his/her request from, the request will have to go through the backbone of the Internet provider to the other side of the network where the data center relies. Besides this extra network traffic and this latency overhead that could be avoided, other common centralization drawbacks in this context stand in limitations in terms of security/legal issues and resilience.

At the same time, it appears that network backbones are over-provisioned for most of their usage. This advocates for placing computing resources directly within the backbone network. The general challenge of resource management for such clouds stands in trying to be locality-aware: for the needs of an application, several virtual machines may exchange data. Placing them *close* to each others can significantly improve the performance of the application they compose. More generally, building an overlay network which takes the hierarchical aspects of the platform without being a hierarchical overlay – which comes with load balancing and resilience issues is a challenge by itself.

We expect to integrate the results of these works in the Discovery initiative [41] which aims at revisiting OpenStack to offer a cloud stack able to manage utility computing platforms where computing resources are located in small computing centers in the backbone's PoPs (Point of Presence) and interconnected through the backbone's internal links.

3.2.2. Resource management in fog computing architectures

Fog computing infrastructures are composed of compute, storage and networking resources located at the edge of wide-area networks, in immediate proximity to the end users. Instead of treating the mobile operator's network as a high-latency dumb pipe between the end users and the external service providers, fog platforms aim at deploying cloud functionalities *within* the mobile phone network, inside or close to the mobile access points. Doing so is expected to deliver added value to the content providers and the end users by enabling new types of applications ranging from Internet-of-Things applications to extremely interactive systems (e.g., augmented reality). Simultaneously, it will generate extra revenue streams for the mobile network operators, by allowing them to position themselves as cloud computing operators and to rent their already-deployed infrastructure to content and application providers.

Fog computing platforms have very different geographical distribution compared to traditional clouds. While traditional clouds are composed of many reliable and powerful machines located in a very small number of data centers and interconnected by very high-speed networks, mobile edge cloud are composed of a very large number of points-of-presence with a couple of weak and potentially unreliable servers, interconnected with each other by commodity long-distance networks. This creates new demands for the organization of a scalable mobile edge computing infrastructure, and opens new directions for research.

The main challenges that we plan to address are:

- How should an edge cloud infrastructure be designed such that it remains scalable, fault-tolerant, controllable, energy-efficient, etc.?
- How should applications making use of edge clouds be organized? One promising direction is to explore the extent to which stream-data processing platforms such as Apache Spark and Apache Flink can be adapted to become one of the main application programming paradigms in such environments.

3.2.3. Self-optimizing applications in multi-cloud environments

As the use of cloud computing becomes pervasive, the ability to deploy an application on a multi-cloud infrastructure becomes increasingly important. Potential benefits include avoiding dependence on a single vendor, taking advantage of lower resource prices or resource proximity, and enhancing application availability. Supporting multi-cloud application management involves two tasks. First, it involves selecting an initial multi-cloud application deployment that best satisfies application objectives and optimizes performance and cost. Second, it involves dynamically adapting the application deployment in order to react to changes in execution conditions, application objectives, cloud provider offerings, or resource prices. Handling price changes in particular is becoming increasingly complex. The reason is the growing trend of providers offering sophisticated, dynamic pricing models that allow buying and selling resources of finer granularities for shorter time durations with varying prices.

Although multi-cloud platforms are starting to emerge, these platforms impose a considerable amount of effort on developers and operations engineers, provide no support for dynamic pricing, and lack the responsiveness and scalability necessary for handling highly-distributed, dynamic applications with strict quality requirements. The goal of this work is to develop techniques and mechanisms for automating application management, enabling applications to cope with and take advantage of the dynamic, diverse, multi-cloud environment in which they operate.

The main challenges arising in this context are:

- selecting effective decision-making approaches for application adaptation,
- supporting scalable monitoring and adaptation across multiple clouds,
- performing adaptation actions in a cost-efficient and safe manner.

3.3. Greening clouds

The ICT (Information and Communications Technologies) ecosystem now approaches 5% of world electricity consumption and this ICT energy use will continue grow fast because of the information appetite of Big Data, big networks and big infrastructures as Clouds that unavoidably leads to big power.

3.3.1. Smart grids and clouds

We propose exploiting Smart Grid technologies to come to the rescue of energy-hungry Clouds. Unlike in traditional electrical distribution networks, where power can only be moved and scheduled in very limited ways, Smart Grids dynamically and effectively adapt supply to demand and limit electricity losses (currently 10% of produced energy is lost during transmission and distribution).

For instance, when a user submits a Cloud request (such as a Google search for instance), it is routed to a data center that processes it, computes the answer and sends it back to the user. Google owns several data centers spread across the world and for performance reasons, the center answering the user's request is more likely to be the one closest to the user. However, this data center may be less energy efficient. This request may have consumed less energy, or a different kind of energy (renewable or not), if it had been sent to this further data center. In this case, the response time would have been increased but maybe not noticeably: a different trade-off between quality of service (QoS) and energy-efficiency could have been adopted.

While Clouds come naturally to the rescue of Smart Grids for dealing with this big data issue, little attention has been paid to the benefits that Smart Grids could bring to distributed Clouds. To our knowledge, no previous work has exploited the Smart Grids potential to obtain and control the energy consumption of entire Cloud infrastructures from underlying facilities such as air conditioning equipment (which accounts for 30% to 50% of a data center's electricity bill) to network resources (which are often operated by several actors) and to computing resources (with their heterogeneity and distribution across multiple data centers). We aim at taking advantage of the opportunity brought by the Smart Grids to exploit renewable energy availability and to optimize energy management in distributed Clouds.

3.3.2. Energy cost models

Cloud computing allows users to outsource the computer resources required for their applications instead of using a local installation. It offers on-demand access to the resources through the Internet with a pay-as-you-go pricing model.However, this model hides the electricity cost of running these infrastructures.

The costs of current data centers are mostly driven by their energy consumption (specifically by the air conditioning, computing and networking infrastructure). Yet, current pricing models are usually static and rarely consider the facilities' energy consumption per user. The challenge is to provide a fair and predictable model to attribute the overall energy costs per virtual machine and to increase energy-awareness of users.

Another goal consists in better understanding the energy consumption of computing and networking resources of Clouds in order to provide energy cost models for the entire infrastructure including incentivizing cost models for both Cloud providers and energy suppliers. These models will be based on experimental measurement campaigns on heterogeneous devices. Inferring a cost model from energy measurements is an arduous task since simple models are not convincing, as shown in our previous work. We aim at proposing and validating energy cost models for the heterogeneous Cloud infrastructures in one hand, and the energy distribution grid on the other hand. These models will be integrated into simulation frameworks in order to validate our energy-efficient algorithms at larger scale.

3.3.3. Energy-aware users

In a Cloud moderately loaded, some servers may be turned off when not used for energy saving purpose. Cloud providers can apply resource management strategies to favor idle servers. Some of the existing solutions propose mechanisms to optimize VM scheduling in the Cloud. A common solution is to consolidate the mapping of the VMs in the Cloud by grouping them in a fewer number of servers. The unused servers can then be turned off in order to lower the global electricity consumption.

Indeed, current work focuses on possible levers at the virtual machine suppliers and/or services. However, users are not involved in the choice of using these levers while significant energy savings could be achieved with their help. For example, they might agree to delay slightly the calculation of the response to their applications on the Cloud or accept that it is supported by a remote data center, to save energy or wait for the availability of renewable energy. The VMs are black boxes from the Cloud provider point of view. So, the user is the only one to know the applications running on her VMs.

We plan to explore possible collaborations between virtual machine suppliers, service providers and users of Clouds in order to provide users with ways of participating in the reduction of the Clouds energy consumption. This work will follow two directions: 1) to investigate compromises between power and performance/service quality that cloud providers can offer to their users and to propose them a variety of options adapted to their workload; and 2) to develop mechanisms for each layer of the Cloud software stack to provide users with a quantification of the energy consumed by each of their options as an incentive to become greener.

3.4. Securing clouds

3.4.1. Security monitoring SLO

While the trend for companies to outsource their information system in clouds is confirmed, the problem of securing an information system becomes more difficult. Indeed, in the case of infrastructure clouds, physical resources are shared between companies (also called tenants) but each tenant controls only parts of the shared resources, and, thanks to virtualization, the information system can be dynamically and automatically reconfigured with added or removed resources (for example starting or stopping virtual machines), or even moved between physical resources (for example using virtual machine migration). Partial control of shared resources brings new classes of attacks between tenants, and security monitoring mechanisms to detect such attacks are better placed out of the tenant-controlled virtual information systems, that is under control of the cloud provider. Dynamic and automatic reconfigurations of the information system make it unfeasible for a tenant's security administrator to setup the security monitoring components to detect attacks, and thus an automated self-adaptable security monitoring service is required.

Combining the two previous statements, there is a need for a dependable, automatic security monitoring service provided to tenants by the cloud provider. Our goal is to address the following challenges to design such a security monitoring service:

1. to define relevant Service-Level Objectives (SLOs) of a security monitoring service, that can figure in the Service-Level Agreement (SLA) signed between a cloud provider and a tenant;
2. to design heuristics to automatically configure provider-controlled security monitoring software components and devices so that SLOs are reached, even during automatic reconfigurations of tenants' information systems;
3. to design evaluation methods for tenants to check that SLOs are reached.

Moreover in challenges 2 and 3 the following sub-challenges must be addressed:

- although SLAs are bi-lateral contracts between the provider and each tenant, the implementation of the contracts is based on shared resources, and thus we must study methods to combine the SLOs;
- the designed methods should have a minimal impact on performance.

3.4.2. Data Protection in Cloud-based IoT Services

The Internet of Things is becoming a reality. Individuals have their own swarm of connected devices (e.g. smartphone, wearables, and home connected objects) continually collecting personal data. A novel generation of services is emerging exploiting data streams produced by the devices' sensors. People are deprived of control of their personal data as they don't know precisely what data are collected by service providers operating on Internet (oISP), for which purpose they could be used, for how long they are stored, and to whom they are disclosed. In response to privacy concerns the European Union has introduced, with the Global Data Protection Regulation (GDPR), new rules aimed at enforcing the people's rights to personal data protection. The GDPR also gives strong incentives to oISPs to comply. However, today, oISPs can't make their systems GDPR-compliant since they don't have the required technologies. We argue that a new generation of system is mandatory for enabling oISPs to conform to the GDPR. We plan to to design an open source distributed operating system for native implementation of new GDPR rules and ease the programming of compliant cloud-based IoT services. Among the new rules, transparency, right of erasure, and accountability are the most challenging ones to be implemented in IoT environments but could fundamentally increase people's confidence in oISPs. Deployed on individuals' swarms of devices and oISPs' cloud-hosted servers, it will enforce detailed data protection agreements and accountability of oISPs' data processing activities. Ultimately we will show to what extend the new GDPR rules can be implemented for cloud-based IoT services.

3.5. Experimenting with Clouds

Cloud platforms are challenging to evaluate and study with a sound scientific methodology. As with any distributed platform, it is very difficult to gather a global and precise view of the system state. Experiments are not reproducible by default since these systems are shared between several stakeholder. This is even worsen by the fact that microscopic differences in the experimental conditions can lead to drastic changes since typical Cloud applications continuously adapt their behavior to the system conditions.

3.5.1. Experimentation methodologies for clouds

We propose to combine two complementary experimental approaches: direct execution on testbeds such as Grid'5000, that are eminently believable but rather labor intensive, and simulations (using *e.g.* SimGrid) that are much more light-weighted, but requires are careful assessment. One specificity of the Myriads team is that we are working on these experimental methodologies *per se*, raising the standards of *good experiments* in our community.

We plan to make SimGrid widely usable beyond research laboratories, in order to evaluate industrial systems and to teach the future generations of cloud practitioners. This requires to frame the specific concepts of Cloud systems and platforms in actionable interfaces. The challenge is to make the framework both easy to use for simple studies in educational settings while modular and extensible to suit the specific needs of every advanced industrial-class users.

We aim at leveraging the convergence opportunities between methodologies by further bridging simulation and real testbeds. The predictions obtained from the simulator should be validated against some real-world experiments obtained on the target production platform, or on a similar platform. This (in)validation of the predicted results often improves the understanding of the modeled system. On the other side, it may even happen that the measured discrepancies are due to some mis-configuration of the real platform that would have been undetected without this (in)validation study. In that sense, the simulator constitutes a precious tool for the quality assurance of real testbeds such as Grid'5000.

Scientists need more help to make there Cloud experiments fully reproducible, in the sprit of Open Science exemplified by the HAL Open Archive, actively backed by Inria. Users still need practical solutions to archive, share and compare the whole experimental settings, including the raw data production (particularly in the case of real testbeds) and their statistical analysis. This is a long lasting task to which we plan to collaborate through the research communities gathered around the Grid'5000 and SimGrid scientific instruments.

Finally, since correction and performance can constitute contradictory goals, it is particularly important to study them jointly. To that extend, we want to bridge the performance studies, that constitute our main scientific heritage, to correction studies leveraging formal techniques. SimGrid already includes to exhaustively explore the possible executions. We plan to continue this work to ease the use of the relevant formal methods to the experimenter studying Cloud systems.

3.5.2. Use cases

In system research it is important to work on real-world use cases from which we extract requirements inspiring new research directions and with which we can validate the system services and mechanisms we propose. In the framework of our close collaboration with the Data Science Technology department of the LBNL, we will investigate cloud usage for scientific data management. Next-generation scientific discoveries are at the boundaries of datasets, e.g., across multiple science disciplines, institutions and spatial and temporal scales. Today, data integration processes and methods are largely adhoc or manual. A generalized resource infrastructure that integrates knowledge of the data and the processing tasks being performed by the user in the context of the data and resource lifecycle is needed. Clouds provide an important infrastructure platform that can be leveraged by including knowledge for distributed data integration.

STACK Team

3. Research Program

3.1. Overview

STACK research activities have been organized around four research topics. The two first ones are related to the resource management mechanisms and the programming support that are mandatory to operate and use ICT geo-distributed resources (compute, storage, network). They are transverse to the System/Middleware/Application layers, which generally composed a software stack, and nurture each other (*i.e.*, the resource management mechanisms will leverage abstractions/concepts proposed by the programming support axis and reciprocally). The third and fourth research topics are related to the Energy and Security dimensions (both also crosscutting the three software layers). Although they could have been merged with the two first axes, we identified them as independent research directions due to their critical aspects with respect to the societal challenges they represent. In the following, we detail the actions we plan to do in each research direction.

3.2. Resource Management

The challenge in this axis is to identify, design or revise mechanisms that are mandatory to operate and use a set of massively geo-distributed resources in an efficient manner [15]. This covers considering challenges at the scale of nodes, within one site (*i.e.*, one geographical location) and throughout the whole geo-distributed ICT infrastructure. It is noteworthy that the network community has been investigating similar challenges for the last few years [73]. To benefit from their expertise, in particular on how to deal with intermittent networks, STACK members have recently initiated exchanges and collaborative actions with some network research groups and telcos (see Sections 8.1 and 9.1). We emphasize, however, that we do not deliver contributions related to network equipments/protocols. The scientific and technical achievements we aim to deliver are related to the (distributed) system aspects.

3.2.1. Performance Characterization of Low-Level Building Blocks

Although Cloud Computing has enabled the consolidation of services and applications into a subset of servers, current operating system mechanisms do not provide appropriate abstractions to prevent (or at least control) the performance degradation that occurs when several workloads compete for the same resources [105]. Keeping in mind that server density is going to increase with physical machines composed of more and more cores and that applications will be more and more data intensive, it is mandatory to identify interferences that appear at a low level on each dimension (compute, memory, network, and storage) and propose countermeasures. In particular, previous studies [105], [65] on pros and cons of current technologies – virtual machines (VMs) [79], [87], containers and microservices – which are used to consolidate applications on the same server, should be extended: In addition to evaluating the performance we can expect from each of these technologies on a single node, it is important to investigate interferences that may result from cross-layer and remote communications [106]. We will consider in particular all interactions related to geo-distributed systems mechanisms/services that are mandatory to operate and use geo-distributed ICT infrastructures.

3.2.2. Geo-Distributed System Mechanisms

Although several studies have been highlighting the advantages of geo-distributed ICT infrastructures in various domains (see Section 3.), progress on how to operate and use such infrastructures is marginal. Current solutions [39] [40] are rather close to the initial Cisco Fog Computing proposal that only allows running domain-specific applications on edge resources and centralized Cloud platforms [47] (in other words, these solutions do not allow running stateful workloads in isolated environments such as containers or VMs). More recently, solutions leveraging the idea of federating VIMs (as the aforementioned ETSI MEC proposal [92]) have been proposed. ONAP [99], an industry-driven solution, enables the orchestration and automation of virtual network functions across distinct VIMs. From the academic side, FogBow [48] aims to support federations of Infrastructure-as-a-Service (IaaS) providers. Finally, NIST initiated a collaborative effort with IEEE to advance Federated Cloud platforms through the development of a conceptual architecture and a vocabulary[0]. Although all these projects provide valuable contributions, they face the aforementioned orchestration limitations (*i.e.*, they do not manage decisions taken in each VIM). Moreover, they all have been designed by only considering the developer/user's perspective. They provide abstractions to manage the life cycle of geo-distributed applications, but do not address administrative requirements.

To cope with specifics of Wide-Area networks while delivering most features that made Cloud Computing solutions successful also at the edge, our community should first identify limitations/drawbacks of current resource management system mechanisms with respect to the Fog/Edge requirements and propose revisions when needed [72], [85].

To achieve this aim, STACK members propose to conduct first a series of studies aiming at understanding the software architecture and footprint of major services that are mandatory for operating and using Fog/Edge infrastructures (storage backends, monitoring services, deployment/reconfiguration mechanisms, etc.). Leveraging these studies, we will investigate how these services should be deployed in order to deal with resources constraints, performance variability, and network split brains. We will rely on contributions that have been accomplished in distributed algorithms and self-* approach for the last decade. In the short and medium term, we plan to evaluate the relevance of NewSQL systems [57] to store internal states of distributed system mechanisms in an edge context, and extend our proposals on new storage backends such as key/value stores [56], [98], and burst buffers [107]. We also plan to conduct new investigations on data-stream frameworks for Fog and Edge infrastructures [52]. These initial contributions should enable us to identify general rules to deliver other advanced system mechanisms that will be mandatory at the higher levels in particular for the deployment and reconfiguration manager in charge of orchestrating all resources.

3.2.3. Capacity Planning and Placement Strategies

An objective shared by users and providers of ICT infrastructures is to limit as much as possible the operational costs while providing the expected and requested quality of service (QoS). To optimize this cost while meeting QoS requirements, data and applications have to be placed in the best possible way onto physical resources according to data sources, data types (stream, graphs), application constraints (real-time requirements) and objective functions. Furthermore, the placement of applications must evolve through time to cope with the fluctuations in terms of application resource needs as well as the physical events that occur at the infrastructure level (resource creation/removals, hardware failures, etc.). This placement problem, *a.k.a.* the deployment and reconfiguration challenge as it will be described in Section 3.3 , can be modelized in many different ways, most of the time by multi-dimensional and multi-objective bin-packing problems or by scheduling problems which are known to be difficult to solve. Many studies have been performed, for example, to optimize the placement of virtual machines onto ICT infrastructures [81]. STACK will inherit the knowledge acquired through previous activities in this domain, particularly its use of constraint programming strategies in autonomic managers [77], [76], relying on MAPE (monitor, analyze, plan, and execute) control loops. While constraint programming approaches are known to hardly scale, they enable the composition of various constraints without requiring to change heuristic algorithms each time a new constraint has to be considered [75]. We believe it is a strong advantage to deal with the diversity brought by geo-distributed ICT infrastructures. Moreover, we have shown in previous work that decentralized approaches can tackle the scalability issue while delivering placement decisions good enough and sometimes close to the optimal [91].

Leveraging this expertise, we propose, first, to identify new constraints raised by massively geo-distributed infrastructures (*e.g.*, data locality, energy, security, reliability and the heterogeneity and mobility of the underlying infrastructure). Based on this preliminary study, we will explore new placement strategies not only for computation sandboxes but for data (location, replication, streams, etc.) in order to benefit from the geo-distribution of resources and meet the required QoS.These investigations should lead to collaborations with operational research and optimization groups such as TASC, another research group from IMT Atlantique.

[0] https://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/FederatedCloudPWGFC (Dec 2018).

Second, we will leverage contributions made on the previous axis “Performance Characterization of Low-Level Building Blocks” to determine how the deployment of the different units (software components and data sets) should be executed in order to reduce as much as possible the time to reconfigure the system (*i.e.*, the *Execution* phase in the control loop). In some recent work [87], we have shown that the provisioning of a new virtual machine should be done carefully to mitigate boot penalties. More generally, proposing an efficient action plan for the *Execution* phase will be a major point as Wide-Area-Network specifics may lead to significant delays, in particular when the amount of data to be manipulated is important.

Finally, we will investigate new approaches to decentralize the placement process while considering the geo-distributed context. Among the different challenges to address, we will study how a combination of autonomic managers, at both the infrastructure and application levels [63], could be proposed in a decentralized manner. Our first idea is to geo-distribute a fleet of small control loops over the whole infrastructure. By improving the locality of data collection and weakening combinatorics, these loops would allow the system to address responsiveness and quality expectations.

3.3. Programming Support

We pursue two main research directions relative to new programming support: first, developing new programming models with appropriate support in existing languages (libraries, embedded DSLs, etc.) and, second, providing new means for deployment and reconfiguration in geo-distributed ICT environments, principally supporting the mapping of software onto the infrastructure. For both directions two levels of challenges are considered. On the one hand, the *generic* level refers to efforts on programming support that can be applied to any kind of distributed software, application or system. On this level, contributions could thus be applied to any of the three layers addressed by STACK (*i.e.*, system, middleware or application). On the other hand, the corresponding generic programming means may not be appropriate in practice (*e.g.*, requirements for more dedicated support, performance constraints, etc.), even if they may lead to interesting general properties. For this reason, a *specific* level is also considered. This level could be based on the generic one but addresses specific cases or domains.

3.3.1. Programming Models and Languages Extensions

The current landscape of programming support for cloud applications is fragmented. This fragmentation is based on apparently different needs for various kinds of applications, in particular, web-based, computation-based, focusing on the organization of the computation, and data-based applications, within the last case a quite strong dichotomy between applications considering data as sets or relations, close to traditional database applications and applications considering data as real-time streams. This has led to various programming models, in a loose sense, including for instance microservices, graph processing, dataflows, streams, etc. These programming models have mostly been offered to the application programmer in the guise of frameworks, each offering subtle variants of the programming models with various implementation decisions favoring particular application and infrastructure settings. Whereas most frameworks are dedicated to a given programming model, *e.g.*, basic Pregel [86], Hive [101], Hadoop [102], some of them are more general-purpose through the provision of several programming models, *e.g.*, Flink [51] and Spark [83]. Finally, some dedicated language support has been considered for some models (*e.g.*, the language SPL underlying IBM Streams [78]) as well as core languages and calculi (*e.g.*, [49], [96]).

This situation raises a number of challenges on its own, related to a better structuring of the landscape. It is necessary to better understand the various programming models and their possible relations, with the aim of facilitating, if not their complete integration, at least their composition, at the conceptual level but also with respect to their implementations, as specific languages and frameworks.

Switching to massively geo-distributed infrastructures adds to these challenges by leading to a new range of applications (*e.g.*, smart-* applications) that, by nature, require mixing these various programming models, together with a much more dynamic management of their runtime.

In this context, STACK would like to explore two directions:

- First, we propose to contribute to generic programming models and languages to address composability of different programming models [58]. For example, providing a generic stream data processing model that can operate under both data stream [51] and operation stream [108] modes, thus streams can be processed in micro batches to favour high throughput or record by record to sustain low latency. Software engineering properties such as separation of concerns and composition should help address such challenges [42], [97]. They should also facilitate the software deployment and reconfiguration challenges discussed below.
- Second, we plan to revise relevant programming models, the associated specific languages, and their implementation according to the massive geo-distribution of the underlying infrastructure, the data sources, and application end-users. For example, although SPL is extensible and distributed, it has been designed to run on multi-cores and clusters [78]. It does not provide the level of dynamicity required by geo-distributed applications (*e.g.*, to handle topology changes, loss of connectivity at the edge, etc.). Moreover, as more network data transfers will happen within a massively geo-distributed infrastructure, correctness of data transfers should be guaranteed. This has potential impact from the programming models to their implementations.

3.3.2. Deployment and Reconfiguration Challenges

The second research direction deals with the complexity of deploying distributed software (whatever the layer, application, middleware or system) onto an underlying infrastructure. As both the deployed pieces of software and the infrastructures addressed by STACK are large, massively distributed, heterogeneous and highly dynamic, the deployment process cannot be handled manually by developers or administrators. Furthermore, and as already mentioned in Section 3.2 , the initial deployment of some distributed software will evolve through time because of the dynamicity of both the deployed software and the underlying infrastructures. When considering reconfiguration, which encompasses deployment as a specific case, the problem becomes more difficult for two main reasons: (1) the current state of both the deployed software and the infrastructure has to be taken into account when deciding on a reconfiguration plan, (2) as the software is already running the reconfiguration should minimize disruption time, while avoiding inconsistencies [84], [89]. Many deployment tools have been proposed both in academia and industry [60]. For example, Ansible [0], Chef [0] and Puppet [0] are very well-known generic tools to automate the deployment process through a set of batch instructions organized in groups (*e.g.*, *playbooks* in Ansible). Some tools are specific to a given environment, like Kolla to deploy OpenStack, or the embedded deployment manager within Spark. Few reconfiguration capabilities are available in production tools such as *scaling* and *restart* after a fault [0] [0]. Academia has contributed to generic deployment and reconfiguration models. Most of these contributions are component-based. Component models divide a distributed software as a set of component instances (or modules) and their assembly, where components are connected through well defined interfaces [97]. Thus, modeling the reconfiguration process consists in describing the life cycle of different components and their interactions. Most component-based approaches offer a fixed life cycle, *i.e.*, identical for any component [66]. Two main contributions are able to customize life cycles, Fractal [50], [45] and its evolutions [42], [43], [62], and Aeolus [64]. In Fractal, the *control* part of a component (*e.g.*, its life cycle) is modeled itself as a component assembly that is highly flexible. Aeolus, on the other hand, offers a finer control on both the evolution and the synchronization of the deployment process by modeling each component life cycle with a finite state machine.

A reconfiguration raises at least five questions, all of them are correlated: (1) *why software has to be reconfigured?* (monitoring, modeling and analysis) (2) *what should be reconfigured?* (software modeling and analysis), (3) *how should it be reconfigured?* (software modeling and planning decisions), (4) *where should it*

[0] https://www.ansible.com/
[0] https://www.chef.io/chef/
[0] https://puppet.com/
[0] https://kubernetes.io/
[0] https://jujucharms.com/

be reconfigured? (infrastructure modeling and planning decisions), and (5) *when to reconfigure it?* (scheduling algorithms). STACK will contribute to all aspects of a reconfiguration process as described above. However, according to the expertise of STACK members, we will focus mainly on the three first questions: *why*, *what* and *how*, leaving questions *where* and *when* to collaborations with operational research and optimization teams.

First of all, we would like to investigate *why software has to be reconfigured?* Many reasons could be mentioned, such as hardware or software fault tolerance, mobile users, dynamicity of software services, etc. All those reasons are related somehow to the Quality of Service (QoS) or the Service Level Agreement (SLA) between the user and the Cloud provider. We first would like to explore the specificities of QoS and SLAs in the case of massively geo-distributed ICT environments [93]. By being able to formalize this question, analyzing the requirement of a reconfiguration will be facilitated.

Second, we think that four important properties should be enhanced when deploying and reconfiguring models in massively geo-distributed ICT environments. First, as low-latency applications and systems will be subject to deployment and reconfiguration, the performance and the ability to scale are important. Second, as many different kinds of deployments and reconfigurations will concurrently hold within the infrastructure, processes have to be reliable, which is facilitated by a fine-grained control of the process. Finally, as many different software elements will be subject to deployment and reconfiguration, common generic models and engines for deployment and reconfiguration should be designed [12]. For these reasons, we intend to go beyond Aeolus by: first, leveraging the expression of parallelism within the deployment process, which should lead to better performance; second, improving the separation of concerns between the component developer and the reconfiguration developer; third, enhancing the possibility to perform concurrent and decentralized reconfigurations.

Research challenges relative to programming support have been presented above. Many of these challenges are related, in different manners, to the resource management level of STACK or to crosscutting challenges, *i.e.*, energy and security. First, one can notice that any programming model or deployment and reconfiguration implementation should be based on mechanisms related to resource management challenges. For this reason, all challenges addressed within this section are linked with lower level building blocks presented in Section 3.2 . Second, as detailed above, deployment and reconfiguration address at least five questions. The question *what?* is naturally related to programming support. However, questions *why*, *how?*, *where?* and *when?* are also related to Section 3.2 , for example, to monitoring and capacity planning. Moreover, regarding the deployment and reconfiguration challenges, one can note that the same goals recursively happen when deploying the control building blocks themselves (bootstrap issue). This comforts the need to design generic deployment and reconfiguration models and frameworks. These low-level models should then be used as back-ends to higher-level solutions. Finally, as *energy* and *security* are crosscutting themes within the STACK project, many additional energy and security considerations could be added to the above challenges. For example, our deployment and reconfiguration frameworks and solutions could be used to guarantee the deployment of end-to-end security policies or to answer specific energy constraints [74] as detailed in the next section.

3.4. Energy

The overall electrical consumption of DCs grows according to the demand of Utility Computing. Considering that the latter has been continuously increasing since 2008, the energy footprint of Cloud services overall is nowadays critical with about 91 billion kilowatt-hours of electricity [95]. Besides the ecological impact, the energy consumption is a predominant criterion for providers since it determines a large part of the operational cost of their infrastructure. Among the different appraoches that have been investigated to reduce the energy footprint, some studies have been ingestigating the use of renewable energy sources to power microDCs [68]. Workload distribution for geo-distributed DCs is also another promising approach [70], [82], [103]. Our research will extend these results with the ultimate goal of considering the different opportunities to control the energy footprint across the whole stack (hardware and software opportunities, renewable energy, thermal management ...). In particular, we identified several challenges that we will address in this context within the STACK framework:

- First, we propose to evaluate the energy efficiency of low-level building blocks, from the viewpoints of computation (VMs, containers, microkernel, microservices) [61] and data (hard drives, SSD, in-memory storage, distributed file systems). For computations, in the continuity of our previous work [59], [77], we will investigate workload placement policies according to energy (minimizing energy consumption, power capping, thermal load balancing, etc.). Regarding the data dimension, we will investigate, in particular, the trade-offs between energy consumption and data availability, durability and consistency [55], [98]. Our ambition is to propose an adaptive energy-aware data layout and replication scheme to ensure data availability with minimal energy consumption. It is noteworthy that these new activities will also consider our previous work on DCs partially powered by renewable energy (see the SeDuCe project, in Section 6.5), with the ultimate goal of reducing the CO_2 footprint.
- Second, we will complete current studies to understand pros and cons of massively geo-distributed infrastructures from the energy perspective. Addressing the energy challenge is a complex task that involves considering several dimensions such as the energy consumption due to the physical resources (CPU, memory, disk, network), the performance of the applications (from the computation and data viewpoints), and the thermal dissipation caused by air conditioning in each DC. Each of these aspects can be influenced by each level of the software stack (*i.e.*, low-level building blocks, coordination and autonomous loops, and finally application life cycle). In previous projects, we have studied and modeled the consumption of the main components, notably the network, as part of a single microDC. We plan to extend these models to deal with geo-distribution. The objective is to propose models that will enable us to refine our placement algorithms as discussed in the next paragraph. These models should be able to consider the energy consumption induced by all WAN data exchanges, including site-to-site data movements as well as the end users' communications for accessing virtualized resources.
- Third, we expect to implement green-energy-aware balancing strategies, leveraging the aforementioned contributions. Although the infrastructures we envision increase complexity (because WAN aspects should also be taken into account), the geo-distribution of resources brings several opportunities from the energy viewpoint. For instance, it is possible to define several workload/data placement policies according to renewable energy availability. Moreover, a tightly-coupled software stack allows users to benefit from such a widely distributed infrastructure in a transparent way while enabling administrators to balance resources in order to benefit from green energy sources when available. An important difficulty, compared to centralized infrastructures, is related to data sharing between software instances. In particular, we will study issues raised by the distribution and replication of services across several microDCs. In this new context, many challenges must be addressed: where to place the data (Cloud, Edge) in order to mitigate dat a movements? What is the impact in terms of energy consumption, network and response time of these two approaches? How to manage the consistency of replicated data/services? All these aspects must be studied and integrated into our placement algorithms.
- Fourth, we will investigate the energy footprint of the current techniques that address failure and performance variability in large-scale systems. For instance, *stragglers* (*i.e.*, tasks that take a significantly longer time to finish than the normal execution time) are natural results of performance variability, they cause extra resource and energy consumption. Our goal is to understand the energy overhead of these techniques and introduce new handling techniques that take into consideration the energy efficiency of the platform [90].
- Finally, in order to answer specific energy constraints, we want to reify energy aspects at the application level and propose a metric related to the use of energy (Green SLA [41]), for example to describe the maximum allowed CO_2 emissions of a Fog/Edge service. Unlike other approaches [71], [46], [69] that attempt to identify the best trade-off, we want to offer to developers/end-users the opportunity to select the best choice between application performance, correctness and energy footprint. Such a capability will require reifying the energy dimension at the level of big-data and interactive applications. Besides, with the emergence of renewable energy (*e.g.*, solar panels for

microDC), investigating the energy consumption *vs* performance trade-off [74] and the smart usage of green energy for ICT geo-distributed services seems promising. For example, we want to offer the opportunity to developers/end-users to control the scaling of the applications based on this trade-off instead of current approaches that only considered application load. Providing such a capability will also require appropriate software abstractions.

3.5. Security

Because of its large size and complex software structure, geo-distributed applications and infrastructures are particularly exposed to security and privacy issues [94]. They are subject to numerous security vulnerabilities that are frequently exploited by malicious attackers in order to exfiltrate personal, institutional or corporate data. Securing these systems require security and privacy models and corresponding techniques that are applicable at all software layers in order to guard interactions at each level but also between levels. However, very few security models exist for the lower layers of the software stack and no model enables the handling of interactions involving the complete software stack. Any modification to its implementation, deployment status, configuration, etc., may introduce new or trigger existing security and privacy issues. Finally, applications that execute on top of the software stack may introduce security issues or be affected by vulnerabilities of the stack. Overall, security and privacy issues are therefore interdependent with all other activities of the STACK team and constitute an important research topic for the team.

As part of the STACK activities, we consider principally security and privacy issues related to the vertical and horizontal compositions of software components forming the software stack and the distributed applications running on top of it. Modifications to the *vertical composition* of the software stack affect different software levels at once. As an example, side-channel attacks often target virtualized services (*i.e.*, services running within VMs); attackers may exploit insecure hardware caches at the system level to exfiltrate data from computations at the higher level of VM services [88], [104]. Security and privacy issues also affect *horizontal compositions*, that is, compositions of software abstractions on one level: most frequently horizontal compositions are considered on the level of applications/services but they are also relevant on the system level or the middleware level, such as compositions involving encryption and database fragmentation services.

The STACK members aim at addressing two main research issues: enabling full-stack (vertical) security and per-layer (horizontal) security. Both of these challenges are particularly hard in the context of large geo-distributed systems because they are often executed on heterogeneous infrastructures and are part of different administrative domains and governed by heterogeneous security and privacy policies. For these reasons they typically lack centralized control, are frequently subject to high latency and are prone to failures.

Concretely, we will consider two classes of security and privacy issues in this context. First, on a general level, we strive for a method for the programming and reasoning about compositions of security and privacy mechanisms including, but not limited to, encryption, database fragmentation and watermarking techniques. Currently, no such general method exists, compositions have only been devised for specific and limited cases, for example, compositions that support the commutation of specific encryption and watermarking techniques [80], [53]. We provided preliminary results on such compositions [54] and have extended them to biomedical, notably genetic, analyses in the e-health domain [11]. Second, on the level of security and privacy properties, we will focus on isolation properties that can be guaranteed through vertical and horizontal composition techniques. We have proposed first results in this context in form of a compositional notion of distributed side channel attacks that operate on the system and middleware levels [44].

It is noteworthy that the STACK members do not have to be experts on the individual security and privacy mechanisms, such as watermarking and database fragmentation. We are, however, well-versed in their main properties so that we can integrate them into our composition model. We also interact closely with experts in these techniques and the corresponding application domains, notably e-health for instance, in the context of the PrivGen project[0], see Section 9.1 .

[0] Privacy-preserving sharing and processing of genetic data, https://privgen.cominlabs.u-bretagneloire.fr/fr

More generally, we highlight that security issues in distributed systems are very closely related to the other STACK challenges, dimensions and research directions. Guaranteeing security properties across the software stack and throughout software layers in highly volatile and heterogeneous geo-distributed systems is expected to harness and contribute results to the self-management capabilities investigated as part of the team's resource management challenges. Furthermore, security and privacy properties are crosscutting concerns that are intimately related to the challenges of application life cycle management. Similarly, the security issues are also closely related to the team's work on programming support. This includes new means for programming, notably in terms of event and stream programming, but also the deployment and reconfiguration challenges, notably concerning automated deployment. As a crosscutting functionality, the security challenges introduced above must be met in an integrated fashion when designing, constructing, executing and adapting distributed applications as well as managing distributed resources.

WIDE Project-Team

3. Research Program

3.1. Overview

In order to progress in the four fields described above, the WIDE team is developing a research program which aims to **help developers control and master the inherent uncertainties and performance challenges brought by scale and distribution**.

More specifically, our program revolves around four key challenges.

- Challenge 1: Designing Hybrid Scalable Architectures,
- Challenge 2: Constructing Personalizable Privacy-Aware Distributed Systems,
- Challenge 3: Understanding Controllable Network Diffusion Processes,
- Challenge 4: Systemizing Modular Distributed Computability and Efficiency.

These four challenges have in common **the inherent tension between coordination and scalability in large-scale distributed systems**: strong coordination mechanisms can deliver strong guarantees (in terms of consistency, agreement, fault-tolerance, and privacy protection), but are generally extremely costly and inherently non-scalable if applied indiscriminately. By contrast, highly scalable coordination approaches (such as epidemic protocols, eventual consistency, or self-organizing overlays) perform much better when the size of a system increases, but do not, in most cases, provide any strong guarantees in terms of consistency or agreement.

The above four challenges explore these tensions from *four complementary angles*: from an architectural perspective (Challenge 1), from the point of view of a fundamental system-wide guarantee (privacy protection, Challenge 2), looking at one universal scalable mechanism (network diffusion, Challenge 3), and considering the interplay between modularity and computability in large-scale systems (Challenge 4). These four challenges range from practical concerns (Challenges 1 and 2) to more theoretical questions (Challenges 3 and 4), yet present *strong synergies* and *fertile interaction points*. E.g. better understanding network diffusion (Challenge 3) is a key enabler to develop more private decentralized systems (Challenge 2), while the development of a theoretically sound modular computability hierarchy (Challenge 4) has a direct impact on our work on hybrid architectures (Challenge 1).

3.2. Hybrid Scalable Architectures

The rise of planetary-scale distributed systems calls for novel software and system architectures that can support user-facing applications while scaling to large numbers of devices, and leveraging established and emerging technologies. The members of WIDE are particularly well positioned to explore this avenue of research thanks to their experience on de-concentrated architectures combining principles from both decentralized peer-to-peer [44], [54] systems and hybrid infrastructures (i.e. architectures that combines centralized or hierarchical elements, often hosted in well-provisioned data-centers, and a decentralized part, often hosted in a peer-to-peer overlay) [48]. In the short term, we aim to explore two axes in this direction: browser-based communication, and micro services.

3.2.1. Browser-based fog computing

The dramatic increase in the amount of data being produced and processed by connected devices has led to paradigms that seek to decentralize the traditional cloud model. In 2011 Cisco [45] introduced the vision of *fog computing* that combines the cloud with resources located at the edge of the network and in between. More generally, the term *edge computing* has been associated with the idea of adding edge-of-the network storage and computation to traditional cloud infrastructures [40].

A number of efforts in this directions focus on specific hardware, e.g. fog nodes that are responsible for connected IoT devices [46]. However, many of today's applications run within web browsers or mobile phones. In this context, the recent introduction of the WebRTC API, makes it possible for browsers and smartphones to exchange directly between each other, enabling mobile, or browser-based decentralized applications. Maygh [68], for example, uses the WebRTC API to build a decentralized Content Delivery Network that runs solely on web browsers. The fact that the application is hosted completely on a web server and downloaded with enabled websites means that webmasters can adopt the Content Delivery Network (CDN) without requiring users to install any specific software.

For us, the ability of browsers to communicate with each other using the WebRTC paradigm provides a novel playground for new programming models, and for a *browser-based fog architecture* combining both a centralized, cloud-based part, and a decentralized, browser-supported part.

This model offers tremendous potential by making edge-of-the-network resources available through the interconnection of web-browsers, and offers new opportunities for the protection of the personal data of end users. But consistently engineering browser-based components requires novel tools and methodologies.

In particular, WebRTC was primarily designed for exchanging media and data between two browsers in the presence of a coordinating server. Its complex mechanisms for connection establishment make many of the existing peer-to-peer protocols inefficient. To address this challenge, we plan to consider two angles of attack. First, we plan to design novel protocols that take into account the specific requirements set by this new technology. Second, we envisage to investigate variants of the current WebRTC model with cheaper connection-establishment protocols, in order to provide lower delays and bandwidth consumption in large-scale browser-based applications.

We also plan to address the trade-offs associated with hybrid browser-cloud models. For example, when should computation be delegated to browsers and when should it be executed on the cloud in order to maximize the quality of service? Or, how can a decentralized analytics algorithms operating on browser-based data complement or exploit the knowledge built by cloud-based data analytics solutions?

3.2.2. Emergent micro-service deployment and management

Micro-services tend to produce fine-grained applications in which many small services interact in a loosely coupled manner to produce a wide range of services within an organization. Individual services need to evolve independently of each other over time without compromising the availability of the overall application. Lightweight isolation solutions such as containers (Docker, ...), and their associated tooling ecosystem (e.g. Google's Borg [67], Kubernetes [43]) have emerged to facilitate the deployment of large-scale micro-service-based applications, but only provide preliminary solutions for key concerns in these systems, which we would like to investigate and extend.

Most of today's on-line computer systems are now too large to evolve in monolithic, entirely pre-planned ways. This applies to very large data centres, for example, where the placement of virtual machines to reduce heating and power consumption can no longer be treated using top-down exhaustive optimisation approaches beyond a critical size. This is also true of social networking applications, where different mechanisms—e.g. to spread news notifications, or to recommend new contacts—must be adapted to the different sub-communities present in the system.

To cope with the inherent complexity of building complex loosely-coupled distributed systems while fostering and increasing efficiency, maintainability, and scalability, we plan to study how novel programming techniques based on declarative programming, components and epidemic protocols can help design, deploy, and maintain self-adaptive structures (e.g. placement of VM) and mechanisms (e.g. contact recommendations) that are optimized to the local context of very large distributed systems. To fulfill this vision, we plan to explore a three-pronged strategy to raise the level of programming abstraction offered to developers.

- First, we plan to explore the use of high-level domain-specific languages (DSL) to declare how large-scale topologies should be achieved, deployed, and maintained. Our vision is a declarative approach to describe how to combine, deploy and orchestrate micro-services in an abstract manner thus abstracting away developers from the underlying cloud infrastructures, and from the intricacies involved in writing low-level code to build a large-scale distributed application that scales. With this effort, we plan notably to directly support the twin properties of *emergence* (the adaptation “from within”) and *differentiation* (the possibility from parts of the system to diverge while still forming a whole). Our central objective is to search for principled programming constructs to support these two capabilities using a modular and incremental software development approach.

- On a second strand of work, we plan to investigate how unikernels enable smaller footprints, more optimization options, and faster boot times for micro-services. Isolating micro-services into VMs is not the most adequate approach as it requires the use of hypervisors, or virtual machine monitors (VMMs), to virtualize hardware resources. VMMs are well known to be heavyweight with both boot and run time overheads that may have a strong impact on performances. Unikernels seem to offer the right balance between performance and flexibility to address this challenge. One of the key underlying challenges is to compile directly the aforementioned provided DSL to a dedicated and customized machine image, ready to be deployed directly on top of a large set of bare metal servers.
- Depending on the workload it is subjected to, and the state of its execution environment (network, VMs), a large-scale distributed application may present erratic or degraded performance that is hard to anticipate and plan for. There is therefore a strong need to adapt dynamically the way resources are allocated to a running application. We would like to study how the DSL approach we envisage can be extended to enable developers to express orchestration algorithms based on machine learning algorithms.

3.3. Personalizable Privacy-Aware Distributed Systems

On-line services are increasingly moving towards an in-depth analysis of user data, with the objective of providing ever better personalization. But in doing so, personalized on-line services inevitably pose risks to the privacy of users. Eliminating, or even reducing these risks raises important challenges caused by the inherent trade-off between the level of personalization users wish to achieve, and the amount of information they are willing to reveal about themselves (explicitly or through the many implicit sources of digital information such as smart homes, smart cars, and IoT environments).

At a general level, we would like to address these challenges through protocols that can provide access to unprecedented amounts of data coming from sensors, users, and documents published by users, while protecting the privacy of individuals and data sources. To this end, we plan to rely on our experience in the context of distributed systems, recommender systems, and privacy, as well as in our collaborations with experts in neighboring fields such as machine learning, and security. In particular, we aim to explore different privacy-utility tradeoffs that make it possible to provide differentiated levels of privacy guarantees depending on the context associated with data, on the users that provide the data, and on those that access it. Our research targets the general goal of privacy-preserving decentralized learning, with applications in different contexts such as user-oriented applications, and the Internet-of-Things (IoT).

3.3.1. Privacy-preserving decentralized learning

Personalization and recommendation can be seen as a specific case of general machine learning. Production-grade recommenders and personalizers typically centralize and process the available data in one location (a data-center, a cloud service). This is highly problematic, as it endangers the privacy of users, while hampering the analysis of datasets subject to privacy constraints that are held by multiple independent organizations (such as health records). A decentralized approach to machine learning appears as a promising candidate to overcome these weaknesses: if each user or participating organization keeps its data, while only exchanging gradient or model information, privacy leaks seem less likely to occur.

In some cases, decentralized learning may be achieved through relatively simple adaptations of existing centralized models, for instance by defining alternative learning models that may be more easily decentralized. But in all cases, processing growing amounts of information calls for high-performance algorithms and middleware that can handle diverse storage and computation resources, in the presence of dynamic and privacy-sensitive data. To reach this objective, we will therefore leverage our work in distributed and privacy-preserving algorithms and middleware [47], [49], [50] as well as the results of our work on large-scale hybrid architectures in Objective 1.

3.3.2. Personalization in user-oriented applications

As a first application perspective, we plan to design tools that exploit decentralized analytics to enhance user-centric personalized applications. As we observed above, such applications exhibit an inherent trade-off between personalization quality and privacy preservation. The most obvious goal in this direction consists in designing algorithms that can achieve high levels of personalization while protecting sensitive user information. But an equally important one consists in personalizing the trade-off itself by adapting the quality of the personalization provided to a user to his/her willingness to expose information. This, like other desirable behaviors, appears at odds with the way current systems work. For example, a user of a recommender system that does not reveal his/her profile information penalizes other users causing them to receive less accurate recommendations. We would like to mitigate this situation by means of protocols that reward users for sharing information. On the one hand, we plan to take inspiration from protocols for free-riding avoidance in peer-to-peer systems [51], [56]. On the other hand, we will consider blockchains as a tool for tracking and rewarding data contributions. Ultimately, we aim at enabling users to configure the level of privacy and personalization they wish to experience.

3.3.3. Privacy preserving decentralized aggregation

As a second setting we would like to consider target applications running on constrained devices like in the Internet-of-Things (IoT). This setting makes it particularly important to operate on decentralized data in a light-weight privacy-preserving manner, and further highlights the synergy between this objective and Objective 1. For example, we plan to provide data subjects with the possibility to store and manage their data locally on their own devices, without having to rely on third-party managers or aggregators, but possibly storing less private information or results in the cloud. Using this strategy, we intend to design protocols that enable users themselves, or third-party companies to query distributed data in aggregate form, or to run data analytics processes on a distributed set of data repositories, thereby gathering knowledge without violating the privacy of other users. For example, we have started working on the problem of computing an aggregate function over a subset of the data in a distributed setting. This involves two major steps: selection and aggregation. With respect to selection, we envision defining a decentralized data-selection operation that can apply a selection predicate without violating privacy constraints. With respect to aggregation, we will continue our investigation of lightweight protocols that can provide privacy with limited computational complexity [41].

3.4. Network Diffusion Processes

Social, biological, and technological networks can serve as conduits for the spread of ideas, trends, diseases, or viruses. In social networks, rumors, trends and behaviors, or the adoption of new products, spread from person to person. In biological networks, diseases spread through contact between individuals, and mutations spread from an individual to its offsprings. In technological networks, such as the Internet and the power grid, viruses and worms spread from computer to computer, and power failures often lead to cascading failures. The common theme in all the examples above is that the rumor, disease, or failure starts out with a single or a few individual nodes, and propagates through the network, from node to node, to reach a potentially much larger number of nodes.

These types of *network diffusion processes* have long been a topic of study in various disciplines, including sociology, biology, physics, mathematics, and more recently, computer science. A main goal has been to devise mathematical models for these processes, describing how the state of an individual node can change as a function of the state of its neighbors in the network, and then analyse the role of the network structure in the outcome of the process. Based on our previous work, we would like to study to what extent one can affect the outcome of the diffusion process by controlling a small, possibly carefully selected fraction of the network.

For example, we plan to explore how we may increase the spread or speed of diffusion by choosing an appropriate set of seed nodes (a standard goal in viral marketing by word-of-mouth), or achieve the opposite effect either by choosing a small set of nodes to remove (a goal in immunization against diseases), or by seeding a competing diffusion (e.g., to limit the spread of misinformation in a social network).

Our goal is to provide a framework for a systematic and rigorous study of these problems. We will consider several standard diffusion models and extensions of them, including models from mathematical sociology, mathematical epidemiology, and interacting particle systems. We will consider existing and new variants of spread maximization/limitation problems, and will provide (approximation) algorithms or show negative (inapproximability) results. In case of negative results, we will investigate general conditions that make the problem tractable. We will consider both general network topologies and specific network models, and will relate the efficiency of solutions to structural properties of the topology. Finally, we will use these insights to engineer new network diffusion processes for efficient data dissemination.

3.4.1. Spread maximization

Our goal is in particular to study spread maximization in a broader class of diffusion processes than the basic independent cascade (IC) and linear threshold (LT) models of influence [60], [61], [62] that have been studied in this context so far. This includes the *randomized rumor spreading (RS)* model for information dissemination [53], *biased* versions of the *voter model* [57] modelling influence, and the (graph-based) *Moran processes* [64] modelling the spread of mutations. We would like to consider several natural versions of the spread maximization problem, and the relationships between them. For these problems we will use the greedy algorithm and the submodularity-based analytical framework of [60], and will also explore new approaches.

3.4.2. Immunization optimization

Conversely we would also like to explore immunization optimization problems. Existing works on these types of problem assume a *perfect-contagion* model, i.e., once a node gets infected, it deterministically infects all its non-immunized neighbors. We plan to consider various diffusion processes, including the standard *susceptible–infected* (SI), *susceptible–infected–recovered* (SIR) and *susceptible–infected–susceptible* (SIS) epidemic models, and explore the extent to which results and techniques for the perfect-contagion model carry over to these probabilistic models. We will also investigate whether techniques for spread maximization could be applied to immunization problems.

Some immunization problems are known to be hard to approximate in general graphs, even for the perfect-contagion model, e.g., the fixed-budget version of the fire-fighter problem cannot be approximated to any $n^{1-\epsilon}$ factor [42]. This strand of work will consider restricted graph families, such as trees or graphs of small treewidth, for such problems. In addition, for some immunization problems, there is a large gap between the best known approximation algorithm and the best known inaproximability result, and we would like to make progress in reducing these gaps.

3.5. Systemizing Modular Distributed Computability and Efficiency

The applications and services envisaged in Objectives 1 and 2 will lead to increasingly complex and multifaceted systems. Constructing these novel hybrid and decentralized systems will naturally push our need to understand distributed computing beyond the current state of the art. These trends therefore demand research efforts in establishing sound theoretical foundations to allow everyday developers to master the design, properties and implementation of these systems. We plan to investigate these foundations along two directions: first by studying novel approaches to some fundamental problems of *mutual exclusion and distributed coordination*, and second by exploring how we can build a *comprehensive and modular framework* capturing the foundations of *distributed computation*.

3.5.1. Randomized algorithm for mutual exclusion and coordination

To exploit the power of massive distributed applications and systems (such as those envisaged in Objectives 1 and 2) or multiple processors, algorithms must cope with the scale and asynchrony of these systems, and their inherent instability, e.g., due to node, link, or processor failures. Our goal is to explore the power and limits of randomized algorithms for large-scale networks of distributed systems, and for shared memory multi-processor systems, in effect providing fundamental building blocks to the work envisioned in Objectives 1 and 2.

For shared memory systems, randomized algorithms have notably proved extremely useful to deal with asynchrony and failures. Sometimes probabilistic algorithms provide the only solution to a problem; sometimes they are more efficient; sometimes they are simply easier to implement. We plan to devise efficient algorithms for some of the fundamental problems of shared memory computing, such as mutual exclusion, renaming, and consensus.

In particular, looking at the problem of *mutual exclusion*, it is desirable that mutual exclusion algorithms be *abortable*. This means that a process that is trying to lock the resource can abort its attempt in case it has to wait too long. Abortability is difficult to achieve for mutual exclusion algorithms. We will try to extend our algorithms for the *cache-coherent* (CC) and the *distributed shared memory* (DSM) model in order to make them abortable, while maintaining expected constant *Remote Memory References* (RMRs) complexity, under optimistic system assumptions. In order to achieve this, the algorithm will use strong synchronization primitives, called compare-and-swap objects. As part of our collaboration with the University of Calgary, we will work on implementing those objects from registers in such a way that they also allow aborts. Our goal is to build on existing non-abortable implementations [55]. We plan then later to use these objects as building blocks in our mutual exclusion algorithm, in order to make them work even if the system does not readily provide such primitives.

We have also started working on blockchains, as these represent a new and interesting trade-off between probabilistic guarantees, scalability, and system dynamics, while revisiting some of the fundamental questions and limitations of consensus in fault-prone asynchronous systems.

3.5.2. Modular theory of distributed computing

Practitioners and engineers have proposed a number of reusable frameworks and services to implement specific distributed services (from Remote Procedure Calls with Java RMI or SOAP-RPC, to JGroups for group communication, and Apache Zookeeper for state machine replication). In spite of the high conceptual and practical interest of such frameworks, many of these efforts lack a sound grounding in distributed computation theory (with the notable exceptions of JGroups and Zookeeper), and often provide punctual and partial solutions for a narrow range of services. We argue that this is because we still lack a generic framework that unifies the large body of fundamental knowledge on distributed computation that has been acquired over the last 40 years.

To overcome this gap we would like to develop a systematic model of distributed computation that organizes the functionalities of a distributed computing system into reusable modular constructs assembled via well-defined mechanisms that maintain sound theoretical guarantees on the resulting system. This research vision arises from the strong belief that distributed computing is now mature enough to resolve the tension between the social needs for distributed computing systems, and the lack of a fundamentally sound and systematic way to realize these systems.

To progress on this vision, we plan in the near future to investigate, from a distributed software point of view, the impact due to failures and asynchrony on the layered architecture of distributed computing systems. A first step in this direction will address the notions of *message adversaries* (introduced a long time ago in [66]) and *process adversaries* (investigated in several papers, e.g. [65], [52], [58], [59], [63]). The aim of these notions is to consider failures, not as "bad events", but as part of the normal behavior of a system. As an example, when considering round-based algorithms, a message adversary is a daemon which, at every round, is allowed to suppress some messages. The aim is then, given a problem P, to find the strongest adversary under which P

can be solved ("strongest" means here that giving more power to the adversary makes the problem impossible to solve). This work will allow us to progress in terms of general *layered* theory of distributed computing, and allow us to better *map* distributed computing models and their relations, in the steps of noticeable early efforts in this direction [65], [39].

HYBRID Project-Team

3. Research Program

3.1. Research Program

The scientific objective of Hybrid team is to improve 3D interaction of one or multiple users with virtual environments, by making full use of physical engagement of the body, and by incorporating the mental states by means of brain-computer interfaces. We intend to improve each component of this framework individually, but we also want to improve the subsequent combinations of these components.

The "hybrid" 3D interaction loop between one or multiple users and a virtual environment is depicted in Figure 1 . Different kinds of 3D interaction situations are distinguished (red arrows, bottom): 1) body-based interaction, 2) mind-based interaction, 3) hybrid and/or 4) collaborative interaction (with at least two users). In each case, three scientific challenges arise which correspond to the three successive steps of the 3D interaction loop (blue squares, top): 1) the 3D interaction technique, 2) the modeling and simulation of the 3D scenario, and 3) the design of appropriate sensory feedback.

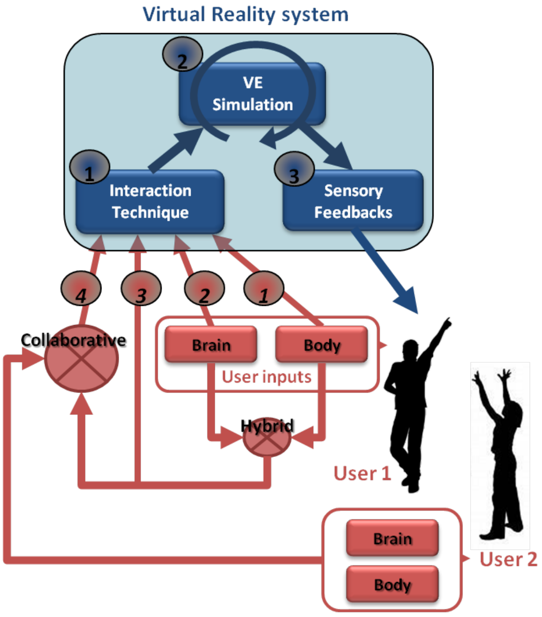

Figure 1. 3D hybrid interaction loop between one or multiple users and a virtual reality system. Top (in blue) three steps of 3D interaction with a virtual environment: (1-blue) interaction technique, (2-blue) simulation of the virtual environment, (3-blue) sensory feedbacks. Bottom (in red) different cases of interaction: (1-red) body-based, (2-red) mind-based, (3-red) hybrid, and (4-red) collaborative 3D interaction.

The 3D interaction loop involves various possible inputs from the user(s) and different kinds of output (or sensory feedback) from the simulated environment. Each user can involve his/her body and mind by means of corporal and/or brain-computer interfaces. A hybrid 3D interaction technique (1) mixes mental and motor inputs and translates them into a command for the virtual environment. The real-time simulation (2) of the virtual environment is taking into account these commands to change and update the state of the virtual world and virtual objects. The state changes are sent back to the user and perceived by means of different sensory feedbacks (e.g., visual, haptic and/or auditory) (3). The sensory feedbacks are closing the 3D interaction loop. Other users can also interact with the virtual environment using the same procedure, and can eventually “collaborate” by means of “collaborative interactive techniques” (4).

This description is stressing three major challenges which correspond to three mandatory steps when designing 3D interaction with virtual environments:

- **3D interaction techniques:** This first step consists in translating the actions or intentions of the user (inputs) into an explicit command for the virtual environment. In virtual reality, the classical tasks that require such kinds of user command were early categorized in four [50]: navigating the virtual world, selecting a virtual object, manipulating it, or controlling the application (entering text, activating options, etc). The addition of a third dimension, the use of stereoscopic rendering and the use of advanced VR interfaces make however inappropriate many techniques that proved efficient in 2D, and make it necessary to design specific interaction techniques and adapted tools. This challenge is here renewed by the various kinds of 3D interaction which are targeted. In our case, we consider various cases, with motor and/or cerebral inputs, and potentially multiple users.
- **Modeling and simulation of complex 3D scenarios:** This second step corresponds to the update of the state of the virtual environment, in real-time, in response to all the potential commands or actions sent by the user. The complexity of the data and phenomena involved in 3D scenarios is constantly increasing. It corresponds for instance to the multiple states of the entities present in the simulation (rigid, articulated, deformable, fluids, which can constitute both the user’s virtual body and the different manipulated objects), and the multiple physical phenomena implied by natural human interactions (squeezing, breaking, melting, etc). The challenge consists here in modeling and simulating these complex 3D scenarios and meeting, at the same time, two strong constraints of virtual reality systems: performance (real-time and interactivity) and genericity (e.g., multi-resolution, multi-modal, multi-platform, etc).
- **Immersive sensory feedbacks:** This third step corresponds to the display of the multiple sensory feedbacks (output) coming from the various VR interfaces. These feedbacks enable the user to perceive the changes occurring in the virtual environment. They are closing the 3D interaction loop, making the user immersed, and potentially generating a subsequent feeling of presence. Among the various VR interfaces which have been developed so far we can stress two kinds of sensory feedback: visual feedback (3D stereoscopic images using projection-based systems such as CAVE systems or Head Mounted Displays); and haptic feedback (related to the sense of touch and to tactile or force-feedback devices). The Hybrid team has a strong expertize in haptic feedback, and in the design of haptic and “pseudo-haptic” rendering [51]. Note that a major trend in the community, which is strongly supported by the Hybrid team, relates to a “perception-based” approach, which aims at designing sensory feedbacks which are well in line with human perceptual capacities.

These three scientific challenges are addressed differently according to the context and the user inputs involved. We propose to consider three different contexts, which correspond to the three different research axes of the Hybrid research team, namely: 1) body-based interaction (motor input only), 2) mind-based interaction (cerebral input only), and then 3) hybrid and collaborative interaction (i.e., the mixing of body and brain inputs from one or multiple users).

3.2. Research Axes

The scientific activity of Hybrid team follows three main axes of research:

- **Body-based interaction in virtual reality.** Our first research axis concerns the design of immersive and effective "body-based" 3D interactions, i.e., relying on a physical engagement of the user’s body. This trend is probably the most popular one in VR research at the moment. Most VR setups make use of tracking systems which measure specific positions or actions of the user in order to interact with a virtual environment. However, in recent years, novel options have emerged for measuring “full-body” movements or other, even less conventional, inputs (e.g. body equilibrium). In this first research axis we are thus concerned by the emergence of new kinds of “body-based interaction” with virtual environments. This implies the design of novel 3D user interfaces and novel 3D interactive techniques, novel simulation models and techniques, and novel sensory feedbacks for body-based interaction with virtual worlds. It involves real-time physical simulation of complex interactive phenomena, and the design of corresponding haptic and pseudo-haptic feedback.

- **Mind-based interaction in virtual reality.** Our second research axis concerns the design of immersive and effective “mind-based” 3D interactions in Virtual Reality. Mind-based interaction with virtual environments is making use of Brain-Computer Interface technology. This technology corresponds to the direct use of brain signals to send “mental commands” to an automated system such as a robot, a prosthesis, or a virtual environment. BCI is a rapidly growing area of research and several impressive prototypes are already available. However, the emergence of such a novel user input is also calling for novel and dedicated 3D user interfaces. This implies to study the extension of the mental vocabulary available for 3D interaction with VE, then the design of specific 3D interaction techniques "driven by the mind" and, last, the design of immersive sensory feedbacks that could help improving the learning of brain control in VR.
- **Hybrid and collaborative 3D interaction.** Our third research axis intends to study the combination of motor and mental inputs in VR, for one or multiple users. This concerns the design of mixed systems, with potentially collaborative scenarios involving multiple users, and thus, multiple bodies and multiple brains sharing the same VE. This research axis therefore involves two interdependent topics: 1) collaborative virtual environments, and 2) hybrid interaction. It should end up with collaborative virtual environments with multiple users, and shared systems with body and mind inputs.

LACODAM Project-Team

3. Research Program

3.1. Introduction

The three research axes of the LACODAM project-team are the following. First, we briefly introduce these axes, as well as their interplay:

- The first research axis is dedicated to the design of *novel pattern mining methods*. Pattern mining is one of the most important approaches to discover novel knowledge in data, and one of our strongest areas of expertise. The work on this axis will serve as foundations for work on the other two axes. Thus, this axis will have the strongest impact on our goals overall.
- The second axis tackles another aspect of knowledge discovery in data: the *interaction between the user and the system* in order to co-discover novel knowledge. Our team has plenty of experience collaborating with domain experts, and is therefore aware of the need to improve such interaction.
- The third axis concerns *decision support*. With the help of methods from the two previous axes, our goal here is to design systems that can either assist humans with making decisions, or make relevant decisions in situations where extremely fast reaction is required.

The following figure sums up the detailed work presented in the next few pages: we show the three research axes of the team (X-axis) on the left and our main applications areas (Y-axis) below. In the middle there are colored squares that represent the precise research topics of the team aligned with their axis and main application area. These research topics will be described in this section. Lines represent projects that can link several topics, and that are also connected to their main application area.

3.2. Pattern mining algorithms

Twenty years of research in pattern mining have resulted in efficient approaches to handle the algorithmic complexity of the problem. Existing algorithms are now able to efficiently extract patterns with complex structures (ex: sequences, graphs, co-variations) from large datasets. However, when dealing with large, real-world datasets, these methods still output a huge set of patterns, which is impractical for human analysis. This problem is called *pattern explosion*. The ongoing challenge of pattern mining research is to extract fewer but more meaningful patterns. The LACODAM team is committed to solve the pattern explosion problem by pursuing the following four research topics:

1. the design of dedicated algorithms for mining temporal patterns
2. the design of flexible pattern mining approaches
3. the automatic selection of interesting data mining results
4. the design of parallel pattern algorithms to ensure scalability

The originality of our contributions relies on the exploration of knowledge-based approaches whose principle is to incorporate dedicated domain knowledge (aka application background knowledge) deep into the mining process. While most data mining approaches are based on agnostic approaches designed to cope with pattern explosion, we propose to develop data mining techniques that rely on knowledge-based artificial intelligence techniques. This entails the use of structured knowledge representations, as well as reasoning methods, in combination with mining.

The first topic concerns classical pattern mining in conjunction with expert knowledge in order to define new pattern types (and related algorithms) that can solve applicative issues. In particular, we investigate how to handle temporality in pattern representations which turns out to be important in many real world applications (in particular for decision support) and deserves particular attention.

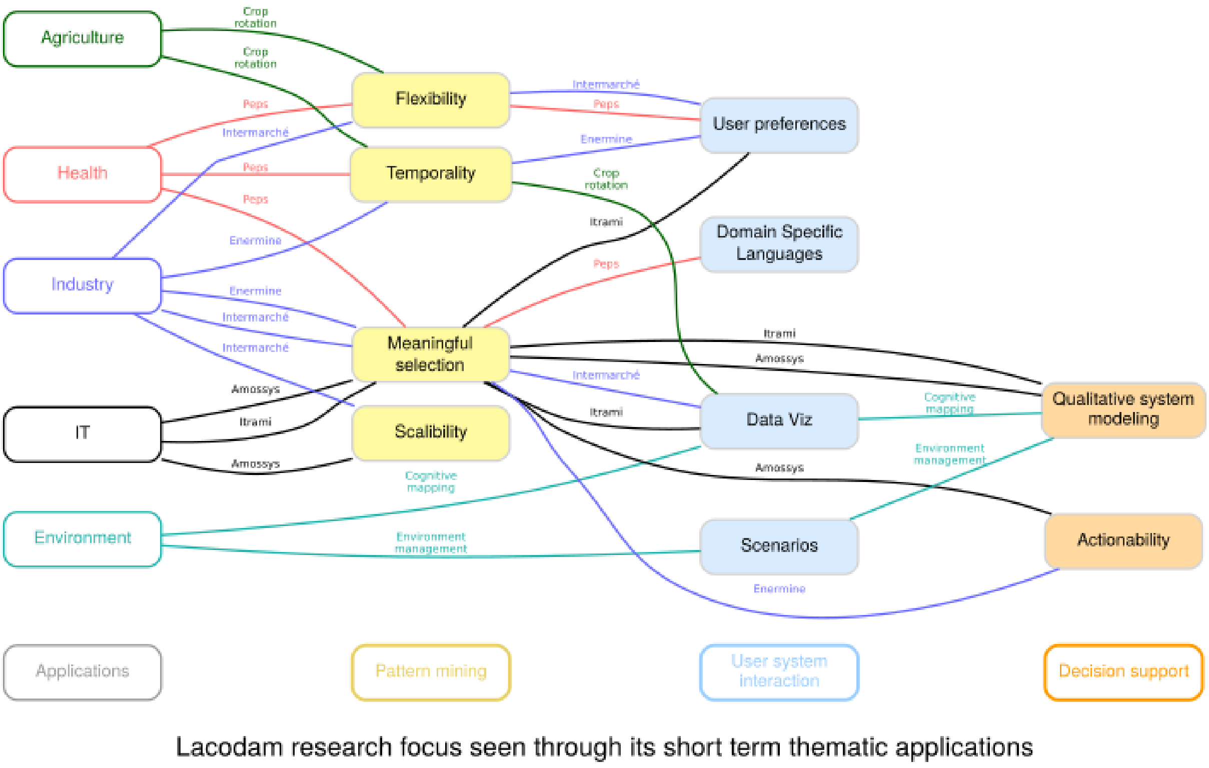

Figure 1. Lacodam research topics organized by axis and application

The next two topics aim at proposing alternative pattern mining methods to let the user incorporate, on her own, knowledge that will help define her pattern domain of interest. Flexible pattern mining approaches enable analysts to easily incorporate extra knowledge, for example domain related constraints, in order to extract only the most relevant patterns. On the other hand, the selection of interesting data mining results aims at devising strategies to filter out the results that are useless to the data analyst. Besides the challenge of algorithmic efficiency, we are interested in formalizing the foundations of interestingness, according to background knowledge modeled with logical knowledge representation paradigms.

Last but not least, pattern mining algorithms are compute-intensive. It is thus important to exploit all the available computing power. Parallelism is for a foreseeable future one of the main ways to speed up computations, and we have a strong competence on the design of parallel pattern mining algorithms. We will exploit this competence in order to guarantee that our approaches scale up to the data provided by our partners.

3.3. User/system interaction

As we pointed out before, there is a strong need to present relevant patterns to the user. This can be done by using more specific constraints, background knowledge and/or tailor-made optimization functions. Due to the difficulty of determining these elements beforehand, one of the most promising solutions is that the system and the user co-construct the definition of relevance, i.e., to have a human in the loop. This requires to have means to present intermediate results to the user, and to get user feedback in order to guide the search space exploration process in the right direction. This is an important research axis for LACODAM, which will be tackled in several complementary ways:

- *Domain Specific Languages:* One way to interact with the user is to propose a Domain Specific Language (DSL) tailored to the domain at hand and to the analysis tasks. The challenge is to propose a DSL allowing the users to easily express the required processing workflows, to deploy those workflows for mining on large volumes of data and to offer as much automation as possible.
- *What if / What for scenarios:* We also investigate the use of scenarios to query results from data mining processes, as well as other complex processes such as complex system simulations or model predictions. Such scenarios are answers to questions of the type “what if [situation]?” or “what [should be done] for [expected outcome]?”.
- *User preferences:* In exploratory analysis, users often do not have a precise idea of what they want, and are not able to formulate such queries. Hence, in LACODAM we investigate simple ways for users to express their interests and preferences, either during the mining process – to guide the search space exploration –, or afterwards during the filtering and interpretation of the most relevant results.
- *Data visualization:* Most of the research directions presented in this document require users to examine patterns at some point. The output of most pattern mining algorithms is usually a (long) list of patterns. While this presentation can be sufficient for some applications, often it does not provide a complete understanding, especially for non-experts in pattern mining. A transversal research topic that we want to explore in LACODAM is to propose data visualization techniques that are adequate for understanding output results. Numerous (failed) experiments have shown that data mining and data visualization are fields, which require distinct skills, thus researchers in one field usually do not make significant advances in the other field (this is detailed in [Keim 2010]). Thus, our strategy is to establish collaborations with prominent data visualization teams for this line of research, with a long term goal to recruit a specialist in data visualization if the opportunity arises.

3.4. Decision support

Patterns have proved to be quite useful for decision-aid. Predictive sequential patterns, to give an example, have a direct application in diagnosis. Itemsets and contrast patterns can be used for interpretable machine learning (ML). In regards to diagnosis, LACODAM inherits, from the former DREAM team, a strong background in decision support systems with internationally recognized expertise in this field. This AI subfield is concerned with determining whether a system is operating normally or not, and the cause of faulty behaviors. The studied system can be an agro- or eco-system, a software system (e.g., a ML classifier), a living being, etc. In relation to interpretable machine learning (ML), this subfield is concerned with the conception of models whose answers are understandable by users. This can be achieved by inducing inherently white-box models from data such as rule-based classifiers/regressors, or by mining rules and explanations from black-box models. The latter setting is quite common due to the high accuracy of black-box models compared to natively interpretable models. Pattern mining is a powerful tool to mine explanations from black-box systems. Those explanations can be used to diagnose biases in systems, either to debug and improve the model, or to generate trust in the verdicts of intelligent software agents.

The increasing volumes of data coming from a range of different systems (ex: sensor data from agro-environmental systems, log data from software systems and ML models, biological data coming from health monitoring systems) can help human and software agents make better decisions. Hence, LACODAM builds upon the idea that decision support systems (an interest bequeathed from DREAM) should take advantage of the available data. This third and last research axis is thus a meeting point for all members of the team, as it requires the integration of AI techniques for traditional decision support systems with results from data mining techniques.

Three main research sub-axes are investigated in LACODAM:

- *Diagnosis-based approaches.* We are exploring how to integrate knowledge found from pattern mining approaches, possibly with the help of interactive methods, into the qualitative models. The goal of such work is to automate as much as possible the construction of prediction models, which can require a lot of human effort.
- *Actionable patterns and rules.* In many settings of “exploratory data mining”, the actual interestingness of a pattern is hard to assess, as it may be subjective. However, for some applications there are well defined measures of interestingness and applicability for patterns. Patterns and rules that can lead to actual actions –that are relevant to the user– are called "actionable patterns" and are of vital importance to industrial settings.
- *Mining explanations from ML systems.* Interpretable ML and AI is a current trend for technical, ethical, and legal reasons [28]. In this regard, pattern mining can be used to spot regularities that arise when a complex black-box model yields a particular verdict. For instance, one may want to know the conditions under which the control module of a self-driving car decided to stop without apparent reason, or which factors caused a ML-based credit assessor to reject a loan request. Patterns and conditions are the building blocks for the generation of human-readable explanations for such black-box systems.

3.5. Long-term goals

The following perspectives are at the convergence of the three aforementioned research axes and can be seen as ideal towards our goals:

- *Automating data science workflow discovery.* The current methods for knowledge extraction and construction of decision support systems require a lot of human effort. Our three research axes aim at alleviating this effort, by devising methods that are more generic and by improving the interaction between the user and the system. An ideal solution would be that the user could forget completely about the existence of pattern mining or decision support methods. Instead the user would only loosely specify her problem, while the system constructs various data science / decision support workflows, possibly further refined via interactions.

  We consider that this is a second order AI task, where AI techniques such as planning are used to explore the workflow search space, the workflow itself being composed of data mining and/or decision support components. This is a strategic evolution for data science endeavors, were the demand far exceeds the available human skilled manpower.

- *Logic argumentation based on epistemic interest.* Having increasingly automated approaches will require better and better ways to handle the interactions with the user. Our second long term goal is to explore the use of logic argumentation, i.e., the formalisation of human strategies for reasoning and arguing, in the interaction between users and data analysis tools. Alongside visualization and interactive data mining tools, logic argumentation can be a way for users to query both the results and the way they are obtained. Such querying can also help the expert to reformulate her query in an interactive analysis setting.

  This research direction aims at exploiting principles of interactive data analysis in the context of epistemic interestingness measures. Logic argumentation can be a natural tool for interactions between the user and the system: display of possibly exhaustive list of arguments, relationships between arguments (e.g., reinforcement, compatibility or conflict), possible solutions for argument conflicts, etc.

  The first step is to define a formal argumentation framework for explaining data mining results. This implies to continue theoretical work on the foundations of argumentation in order to identify the most adapted framework (either existing or a new one to be defined). Logic argumentation may be implemented and deeply explored in ASP, allowing us to build on our expertise in this logic language.

- *Collaborative feedback and knowledge management.* We are convinced that improving the data science process, and possibly automating it, will rely on high-quality feedback from communities on the web. Consider for example what has been achieved by collaborative platforms such as StackOverflow: it has become the reference site for any programming question.

  Data science is a more complex problem than programming, as in order to get help from the community, the user has to share her data and workflow, or at least some parts of them. This raises obvious privacy issues that may prevent this idea to succeed. As our research on automating the production of data science workflows should enable more people to have access to data science results, we are interested in the design of collaborative platforms to exchange expert advices over data, workflows and analysis results. This aims at exploiting human feedback to improve the automation of data science system via machine learning methods.

LINKMEDIA Project-Team

3. Research Program

3.1. Scientific background

LINKMEDIA is a multidisciplinary research team, with multimedia data as the main object of study. We are guided by the data and their specificity—semantically interpretable, heterogeneous and multimodal, available in large amounts, unstructured and disconnected—, as well as by the related problems and applications.

With multimedia data at the center, orienting our choices of methods and algorithms and serving as a basis for experimental validation, the team is directly contributing to the following scientific fields:

- multimedia: content-based analysis; multimodal processing and fusion; multimedia applications;
- computer vision: compact description of images; object and event detection;
- natural language processing: topic segmentation; information extraction;
- information retrieval: high-dimensional indexing; approximate k-nn search; efficient set comparison.

LINKMEDIA also takes advantage of advances in the following fields, adapting recent developments to the multimedia area:

- signal processing: image processing; compression;
- machine learning: deep architectures; structured learning; adversarial learning;
- security: data encryption; differential privacy;
- data mining: time series mining and alignment; pattern discovery; knowledge extraction.

3.2. Workplan

Research activities in LINKMEDIA are organized along three major lines of research which build upon the scientific domains already mentioned.

3.2.1. Unsupervised motif discovery

As an alternative to supervised learning techniques, unsupervised approaches have emerged recently in multimedia with the goal of discovering directly patterns and events of interest from the data, in a totally unsupervised manner. In the absence of prior knowledge on what we are interested in, meaningfulness can be judged based on one of three main criteria: unexpectedness, saliency and recurrence. This last case posits that repeating patterns, known as motifs, are potentially meaningful, leading to recent work on the unsupervised discovery of motifs in multimedia data [50], [48], [49].

LINKMEDIA seeks to *develop unsupervised motif discovery approaches which are both accurate and scalable*. In particular, we consider the discovery of repeating objects in image collections and the discovery of repeated sequences in video and audio streams. Research activities are organized along the following lines:

- developing the scientific basis for scalable motif discovery: sparse histogram representations; efficient co-occurrence counting; geometry and time aware indexing schemes;
- designing and evaluating accurate and scalable motif discovery algorithms applied to a variety of multimedia content: exploiting efficient geometry or time aware matching functions; fast approximate dynamic time warping; symbolic representations of multimedia data, in conjunction with existing symbolic data mining approaches;
- developing methodology for the interpretation, exploitation and evaluation of motif discovery algorithms in various use-cases: image classification; video stream monitoring; transcript-free natural language processing (NLP) for spoken document.

3.2.2. Description and structuring

Content-based analysis has received a lot of attention from the early days of multimedia, with an extensive use of supervised machine learning for all modalities [51], [45]. Progress in large scale entity and event recognition in multimedia content has made available general purpose approaches able to learn from very large data sets and performing fairly decently in a large number of cases. Current solutions are however limited to simple, homogeneous, information and can hardly handle structured information such as hierarchical descriptions, tree-structured or nested concepts.

LINKMEDIA aims at *expanding techniques for multimedia content modeling, event detection and structure analysis*. The main transverse research lines that LINKMEDIA will develop are as follows:

- context-aware content description targeting (homogeneous) collections of multimedia data: latent variable discovery; deep feature learning; motif discovery;
- secure description to enable privacy and security aware multimedia content processing: leveraging encryption and obfuscation; exploring adversarial machine learning in a multimedia context; privacy-oriented image processing;
- multilevel modeling with a focus on probabilistic modeling of structured multimodal data: multiple kernels; structured machine learning; conditional random fields.

3.2.3. Linking and collection data model

Creating explicit links between media content items has been considered on different occasions, with the goal of seeking and discovering information by browsing, as opposed to information retrieval via ranked lists of relevant documents. Content-based link creation has been initially addressed in the hypertext community for well-structured texts [44] and was recently extended to multimedia content [52], [47], [46]. The problem of organizing collections with links remains mainly unsolved for large heterogeneous collections of unstructured documents, with many issues deserving attention: linking at a fine semantic grain; selecting relevant links; characterizing links; evaluating links; etc.

LINKMEDIA targets pioneering research on media linking by *developing scientific ground, methodology and technology for content-based media linking* directed to applications exploiting rich linked content such as navigation or recommendation. Contributions are concentrated along the following lines:

- algorithmic of linked media for content-based link authoring in multimedia collections: time-aware graph construction; multimodal hypergraphs; large scale k-nn graphs;
- link interpretation and characterization to provide links semantics for interpretability: text alignment; entity linking; intention vs. extension;
- linked media usage and evaluation: information retrieval; summarization; data models for navigation; link prediction.

MIMETIC Project-Team

3. Research Program

3.1. Biomechanics and Motion Control

Human motion control is a highly complex phenomenon that involves several layered systems, as shown in Figure 3 . Each layer of this controller is responsible for dealing with perceptual stimuli in order to decide the actions that should be applied to the human body and his environment. Due to the intrinsic complexity of the information (internal representation of the body and mental state, external representation of the environment) used to perform this task, it is almost impossible to model all the possible states of the system. Even for simple problems, there generally exists an infinity of solutions. For example, from the biomechanical point of view, there are much more actuators (i.e. muscles) than degrees of freedom leading to an infinity of muscle activation patterns for a unique joint rotation. From the reactive point of view there exists an infinity of paths to avoid a given obstacle in navigation tasks. At each layer, the key problem is to understand how people select one solution among these infinite state spaces. Several scientific domains have addressed this problem with specific points of view, such as physiology, biomechanics, neurosciences and psychology.

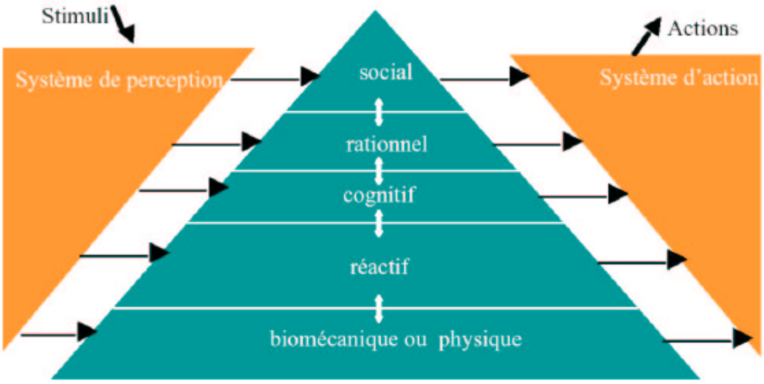

Figure 3. Layers of the motion control natural system in humans.

In biomechanics and physiology, researchers have proposed hypotheses based on accurate joint modeling (to identify the real anatomical rotational axes), energy minimization, force and torques minimization, comfort maximization (i.e. avoiding joint limits), and physiological limitations in muscle force production. All these constraints have been used in optimal controllers to simulate natural motions. The main problem is thus to define how these constraints are composed altogether such as searching the weights used to linearly combine these criteria in order to generate a natural motion. Musculoskeletal models are stereotyped examples for which there exists an infinity of muscle activation patterns, especially when dealing with antagonist muscles. An unresolved problem is to define how to use the above criteria to retrieve the actual activation patterns, while optimization approaches still leads to unrealistic ones. It is still an open problem that will require multidisciplinary skills including computer simulation, constraint solving, biomechanics, optimal control, physiology and neuroscience.

In neuroscience, researchers have proposed other theories, such as coordination patterns between joints driven by simplifications of the variables used to control the motion. The key idea is to assume that instead of controlling all the degrees of freedom, people control higher level variables which correspond to combinations of joint angles. In walking, data reduction techniques such as Principal Component Analysis have shown that lower-limb joint angles are generally projected on a unique plane whose angle in the state space is associated with energy expenditure. Although knowledge exists for specific motions, such as locomotion or grasping, this type of approach is still difficult to generalize. The key problem is that many variables are coupled and it is very difficult to objectively study the behavior of a unique variable in various motor tasks. Computer simulation is a promising method to evaluate such type of assumptions as it enables to accurately control all the variables and to check if it leads to natural movements.

Neuroscience also addresses the problem of coupling perception and action by providing control laws based on visual cues (or any other senses), such as determining how the optical flow is used to control direction in navigation tasks, while dealing with collision avoidance or interception. Coupling of the control variables is enhanced in this case as the state of the body is enriched by the large amount of external information that the subject can use. Virtual environments inhabited with autonomous characters whose behavior is driven by motion control assumptions is a promising approach to solve this problem. For example, an interesting problem in this field is navigation in an environment inhabited with other people. Typically, avoiding static obstacles together with other people displacing into the environment is a combinatory problem that strongly relies on the coupling between perception and action.

One of the main objectives of MimeTIC is to enhance knowledge on human motion control by developing innovative experiments based on computer simulation and immersive environments. To this end, designing experimental protocols is a key point and some of the researchers in MimeTIC have developed this skill in biomechanics and perception-action coupling. Associating these researchers to experts in virtual human simulation, computational geometry and constraints solving enable us to contribute to enhance fundamental knowledge in human motion control.

3.2. Experiments in Virtual Reality

Understanding interactions between humans is challenging because it addresses many complex phenomena including perception, decision-making, cognition and social behaviors. Moreover, all these phenomena are difficult to isolate in real situations, and it is therefore highly complex to understand their individual influence on these human interactions. It is then necessary to find an alternative solution that can standardize the experiments and that allows the modification of only one parameter at a time. Video was first used since the displayed experiment is perfectly repeatable and cut-offs (stop the video at a specific time before its end) allow having temporal information. Nevertheless, the absence of adapted viewpoint and stereoscopic vision does not provide depth information that are very meaningful. Moreover, during video recording session, the real human is acting in front of a camera and not of an opponent. The interaction is then not a real interaction between humans.

Virtual Reality (VR) systems allow full standardization of the experimental situations and the complete control of the virtual environment. It is then possible to modify only one parameter at a time and to observe its influence on the perception of the immersed subject. VR can then be used to understand what information is picked up to make a decision. Moreover, cut-offs can also be used to obtain temporal information about when information is picked up. When the subject can moreover react as in a real situation, his movement (captured in real time) provides information about his reactions to the modified parameter. Not only is the perception studied, but the complete perception-action loop. Perception and action are indeed coupled and influence each other as suggested by Gibson in 1979.

Finally, VR allows the validation of virtual human models. Some models are indeed based on the interaction between the virtual character and the other humans, such as a walking model. In that case, there are two ways to validate it. First, they can be compared to real data (e.g. real trajectories of pedestrians). But such data are not always available and are difficult to get. The alternative solution is then to use VR. The validation of the realism of the model is then done by immersing a real subject in a virtual environment in which a virtual character is controlled by the model. Its evaluation is then deduced from how the immersed subject reacts when interacting with the model and how realistic he feels the virtual character is.

3.3. Computational Geometry

Computational geometry is a branch of computer science devoted to the study of algorithms which can be stated in terms of geometry. It aims at studying algorithms for combinatorial, topological and metric problems concerning sets of points in Euclidian spaces. Combinatorial computational geometry focuses on three main problem classes: static problems, geometric query problems and dynamic problems.

In static problems, some inputs are given and the corresponding outputs need to be constructed or found. Such problems include linear programming, Delaunay triangulations, and Euclidian shortest paths for instance. In geometric query problems, commonly known as geometric search problems, the input consists of two parts: the search space part and the query part, which varies over the problem instances. The search space typically needs to be preprocessed, in a way that multiple queries can be answered efficiently. Some typical problems are range searching, point location in a portioned space, or nearest neighbor queries. In dynamic problems, the goal is to find an efficient algorithm for finding a solution repeatedly after each incremental modification of the input data (addition, deletion or motion of input geometric elements). Algorithms for problems of this type typically involve dynamic data structures. Both of previous problem types can be converted into a dynamic problem, for instance, maintaining a Delaunay triangulation between moving points.

In this context, distance geometry relies solely on distances, instead of points and lines, as in classical geometry. Various applications lead to the definition of problems that can be formulated as a distance geometry, including sensor network localization, robot coordination, the identification of molecular conformations, or as in the context of MimeTIC relations between objects in virtual scenes (e.g., distances between body segments, agents, or cameras). In recent years, scientific research has been oriented to the assumptions allowing for discretizing the search space of a given distance geometry problem. The discretization (which is exact in some situations) allows to conceive ad-hoc and efficient algorithms, and for enumerating the entire solution set of a given instance.

The MimeTIC team works on problems such as crowd simulation, spatial analysis, path and motion planning in static and dynamic environments, camera planning with visibility constraints for instance. The core of those problems, by nature, relies on problems and techniques belonging to computational geometry. Proposed models pay attention to algorithms complexity to be compatible with performance constraints imposed by interactive applications.

PANAMA Project-Team

3. Research Program

3.1. Axis 1: Sparse Models and Representations

3.1.1. Efficient Sparse Models and Dictionary Design for Large-scale Data

Sparse models are at the core of many research domains where the large amount and high-dimensionality of digital data requires concise data descriptions for efficient information processing. Recent breakthroughs have demonstrated the ability of these models to provide concise descriptions of complex data collections, together with algorithms of provable performance and bounded complexity.

A crucial prerequisite for the success of today's methods is the knowledge of a "dictionary" characterizing how to concisely describe the data of interest. Choosing a dictionary is currently something of an "art", relying on expert knowledge and heuristics.

Pre-chosen dictionaries such as wavelets, curvelets or Gabor dictionaries, are based upon stylized signal models and benefit from fast transform algorithms, but they fail to fully describe the content of natural signals and their variability. They do not address the huge diversity underlying modern data much beyond time series and images: data defined on graphs (social networks, internet routing, brain connectivity), vector valued data (diffusion tensor imaging of the brain), multichannel or multi-stream data (audiovisual streams, surveillance networks, multimodal biomedical monitoring).

The alternative to a pre-chosen dictionary is a trained dictionary learned from signal instances. While such representations exhibit good performance on small-scale problems, they are currently limited to low-dimensional signal processing due to the necessary training data, memory requirements and computational complexity. Whether designed or learned from a training corpus, dictionary-based sparse models and the associated methodology fail to scale up to the volume and resolution of modern digital data, for they intrinsically involve difficult linear inverse problems. To overcome this bottleneck, a new generation of efficient sparse models is needed, beyond dictionaries, encompassing the ability to provide sparse and structured data representations as well as computational efficiency. For example, while dictionaries describe low-dimensional signal models in terms of their "synthesis" using few elementary building blocks called atoms, in "analysis" alternatives the low-dimensional structure of the signal is rather "carved out" by a set of equations satisfied by the signal. Linear as well as nonlinear models can be envisioned.

3.1.2. Compressive Learning

A flagship emerging application of sparsity is the paradigm of compressive sensing, which exploits sparse models at the analog and digital levels for the acquisition, compression and transmission of data using limited resources (fewer/less expensive sensors, limited energy consumption and transmission bandwidth, etc.). Besides sparsity, a key pillar of compressive sensing is the use of random low-dimensional projections. Through compressive sensing, random projections have shown their potential to allow drastic dimension reduction with controlled information loss, provided that the projected signal vector admits a sparse representation in some transformed domain. A related scientific domain, where sparsity has been recognized as a key enabling factor, is Machine Learning, where the overall goal is to design statistically founded principles and efficient algorithms in order to infer general properties of large data collections through the observation of a limited number of representative examples. Marrying sparsity and random low-dimensional projections with machine learning shall allow the development of techniques able to efficiently capture and process the information content of large data collections. The expected outcome is a dramatic increase of the impact of sparse models in machine learning, as well as an integrated framework from the signal level (signals and their acquisition) to the semantic level (information and its manipulation), and applications to data sizes and volumes of collections that cannot be handled by current technologies.

3.2. Axis 2: Robust Acoustic Scene Analysis

3.2.1. Compressive Acquisition and Processing of Acoustic Scenes

Acoustic imaging and scene analysis involve acquiring the information content from acoustic fields with a limited number of acoustic sensors. A full 3D+t field at CD quality and Nyquist spatial sampling represents roughly 10^6 microphones/m^3. Dealing with such high-dimensional data requires to drastically reduce the data flow by positioning appropriate sensors, and selecting from all spatial locations the few spots where acoustic sources are active. The main goal is to develop a theoretical and practical understanding of the conditions under which compressive acoustic sensing is both feasible and robust to inaccurate modeling, noisy measures, and partially failing or uncalibrated sensing devices, in various acoustic sensing scenarii. This requires the development of adequate algorithmic tools, numerical simulations, and experimental data in simple settings where hardware prototypes can be implemented.

3.2.2. Robust Audio Source Separation

Audio signal separation consists in extracting the individual sound of different instruments or speakers that were mixed on a recording. It is now successfully addressed in the academic setting of linear instantaneous mixtures. Yet, real-life recordings, generally associated to reverberant environments, remain an unsolved difficult challenge, especially with many sources and few audio channels. Much of the difficulty comes from the combination of (i) complex source characteristics, (ii) sophisticated underlying mixing model and (iii) adverse recording environments. Moreover, as opposed to the "academic" blind source separation task, most applicative contexts and new interaction paradigms offer a variety of situations in which prior knowledge and adequate interfaces enable the design and the use of informed and/or manually assisted source separation methods.

The former METISS team has developed a generic and flexible probabilistic audio source separation framework that has the ability to combine various acoustic models such as spatial and spectral source models. Building on this existing framework, a first objective of PANAMA is to instantiate and validate specific instances of this framework targeted to real-world industrial applications, such as 5.1 movie re-mastering, interactive music soloist control and outdoor speech enhancement. Extensions of the framework are needed to achieve real-time online processing, and advanced constraints or probabilistic priors for the sources at hand need to be designed, while paying attention to computational scalability issues.

In parallel to these efforts, expected progress in sparse modeling for inverse problems shall bring new approaches to source separation and modeling, as well as to source localization, which is often an important first step in a source separation workflow.

3.2.3. Robust Audio Source Localization

Audio source localization consists in estimating the position of one or several sound sources given the signals received by a microphone array. Knowing the geometry of an audio scene is often a pre-requisite to perform higher-level tasks such as speaker identification and tracking, speech enhancement and recognition or audio source separation. It can be decomposed into two sub-tasks : (i) compute spatial auditory features from raw audio input and (ii) map these features to the desired spatial information. Robustly addressing both these aspects with a limited number of microphones, in the presence of noise, reverberation, multiple and possibly moving sources remains a key challenge in audio signal processing. The first aspect will be tackled by both advanced statistical and acoustical modeling of spatial auditory features. The second one will be addressed by two complementary approaches. *Physics-driven* approaches cast sound source localization as an inverse problem given the known physics of sound propagation within the considered system. *Data-driven* approaches aim at learning the desired feature-to-source-position mapping using real-world or synthetic training datasets adapted to the problem at hand. Combining these approaches should allow a widening of the notion of source localization, considering problems such as the identification of the directivity or diffuseness of the source as well as some of the boundary conditions of the room. A general perspective is to investigate the relations between the physical structure of the source and the particular structures that can be discovered or enforced in the representations and models used for characterization, localization and separation.

3.3. Axis 3: Large-scale Audio Content Processing and Self-organization

3.3.1. Motif Discovery in Audio Data

Facing the ever-growing quantity of multimedia content, the topic of motif discovery and mining has become an emerging trend in multimedia data processing with the ultimate goal of developing weakly supervised paradigms for content-based analysis and indexing. In this context, speech, audio and music content, offers a particularly relevant information stream from which meaningful information can be extracted to create some form of "audio icons" (key-sounds, jingles, recurrent locutions, musical choruses, etc ...) without resorting to comprehensive inventories of expected patterns.

This challenge raises several fundamental questions that will be among our core preoccupations over the next few years. The first question is the deployment of motif discovery on a large scale, a task that requires extending audio motif discovery approaches to incorporate efficient time series pattern matching methods (fingerprinting, similarity search indexing algorithms, stochastic modeling, etc.). The second question is that of the use and interpretation of the motifs discovered. Linking motif discovery and symbolic learning techniques, exploiting motif discovery in machine learning are key research directions to enable the interpretation of recurring motifs.

On the application side, several use cases can be envisioned which will benefit from motif discovery deployed on a large scale. For example, in spoken content, word-like repeating fragments can be used for several spoken document-processing tasks such as language-independent topic segmentation or summarization. Recurring motifs can also be used for audio summarization of audio content. More fundamentally, motif discovery paves the way for a shift from supervised learning approaches for content description to unsupervised paradigms where concepts emerge from the data.

3.3.2. Structure Modeling and Inference in Audio and Musical Contents

Structuring information is a key step for the efficient description and learning of all types of contents, and in particular audio and musical contents. Indeed, structure modeling and inference can be understood as the task of detecting dependencies (and thus establishing relationships) between different fragments, parts or sections of information content.

A stake of structure modeling is to enable more robust descriptions of the properties of the content and better model generalization abilities that can be inferred from a particular content, for instance via cache models, trigger models or more general graphical models designed to render the information gained from structural inference. Moreover, the structure itself can become a robust descriptor of the content, which is likely to be more resistant than surface information to a number of operations such as transmission, transduction, copyright infringement or illegal use.

In this context, information theory concepts need to be investigated to provide criteria and paradigms for detecting and modeling structural properties of audio contents, covering potentially a wide range of application domains in speech content mining, music modeling or audio scene monitoring.

RAINBOW Project-Team

3. Research Program

3.1. Main Vision

The vision of Rainbow (and foreseen applications) calls for several general scientific challenges: (i) high-level of autonomy for complex robots in complex (unstructured) environments, (ii) forward interfaces for letting an operator giving high-level commands to the robot, (iii) backward interfaces for informing the operator about the robot 'status', (iv) user studies for assessing the best interfacing, which will clearly depend on the particular task/situation. Within Rainbow we plan to tackle these challenges at different levels of depth:

- the **methodological and algorithmic side** of the sought human-robot interaction will be the **main focus** of Rainbow. Here, by also exploiting the previous Lagadic expertise, we will be interest in advancing the state-of-the-art in sensor-based online planning, control and manipulation for mobile/fixed robots. For instance, while classically most control approaches (especially those sensor-based) have been essentially *reactive*, we believe that less myopic strategies based on online/reactive trajectory optimization will be needed for the future Rainbow activities. The core ideas of Model-Predictive Control approaches (also known as Receding Horizon) or, in general, numerical optimal control methods will play a role in the Rainbow activities, for allowing the robots to reason/plan over some future time window and better cope with constraints. We will also consider extending classical sensor-based motion control/manipulation techniques to more realistic scenarios, such as deformable/flexible objects ("**Advanced Sensor-based Control**" axis). Finally, it will also be important to spend research efforts into the field of *Optimal Sensing*, in the sense of generating (again) trajectories that can optimize the state estimation problem in presence of scarce sensory inputs and/or non-negligible measurement and process noises, especially true for the case of mobile robots ("**Optimal and Uncertainty-Aware Sensing**" axis). We also aim at addressing the case of coordination between a single human user and multiple robots where, clearly, as explained the autonomy part plays even a more crucial role (no human can control multiple robots at once, thus a high degree of autonomy will be required by the robot group for executing the human commands);
- the **interfacing side** will also be a focus of the Rainbow activities. As explained above, we will be interested in both the *forward* (human → robot) and *backward* (robot → human) interfaces. The forward interface will be mainly addressed from the *algorithmic* point of view, i.e., how to map the few degrees of freedom available to a human operator (usually in the order of 3–4) into complex commands for the controlled robot(s). This mapping will typically be mediated by an "AutoPilot" onboard the robot(s) for autonomously assessing if the commands are feasible and, if not, how to least modify them ("**Advanced Sensor-based Control**" axis).
  The backward interface will, instead, mainly consist of a visual/haptic feedback for the operator. Here, we aim at exploiting our expertise in using force cues for informing an operator about the status of the remote robot(s). However, the sole use of classical *grounded* force feedback devices (e.g., the typical force-feedback joysticks) will not be enough due to the different kinds of information that will have to be provided to the operator. In this context, the recent interest in the use of *wearable* haptic interfaces is very interesting and will be investigated in depth (these include, e.g., devices able to provide vibro-tactile information to the fingertips, wrist, or other parts of the body). The main challenges in these activities will be the mechanical conception (and construction) of suitable wearable interfaces for the tasks at hand, and in the generation of force cues for the operator: the force cues will be a (complex) function of the robot state, therefore motivating research in algorithms for mapping the robot state into a few variables (the force cues) ("**Haptics for Robotics Applications**" axis);

- the **evaluation side** that will assess the proposed interfaces with some user studies, or acceptability studies by human subjects. Although this activity **will not** be a main focus of Rainbow (complex user studies are beyond the scope of our core expertise), we will nevertheless devote some efforts into having some reasonable level of user evaluations by applying standard statistical analysis based on psychophysical procedures (e.g., randomized tests and Anova statistical analysis). This will be particularly true for the activities involving the use of smart wheelchairs, which are intended to be used by human users *and* operate inside human crowds. Therefore, we will be interested in gaining some level of understanding of how semi-autonomous robots (a wheelchair in this example) can predict the human intention, and how humans can react to a semi-autonomous mobile robot.

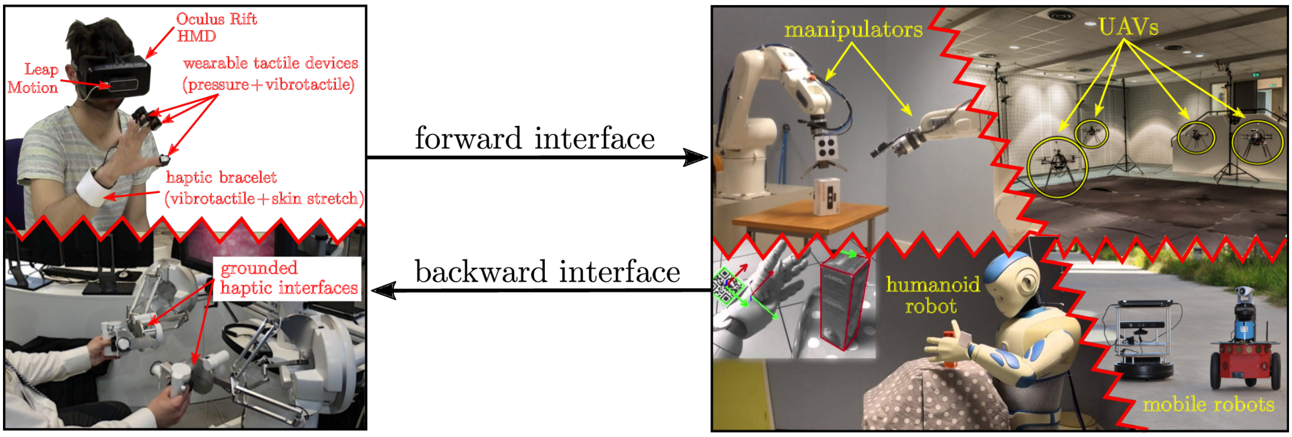

Figure 1. An illustration of the prototypical activities foreseen in Rainbow in which a human operator is in partial (and high-level) control of single/multiple complex robots performing semi-autonomous tasks

Figure 1 depicts in an illustrative way the *prototypical* activities foreseen in Rainbow. On the righthand side, complex robots (dual manipulators, humanoid, single/multiple mobile robots) need to perform some task with high degree of autonomy. On the lefthand side, a human operator gives some high-level commands and receives a visual/haptic feedback aimed at informing her/him at best of the robot status. Again, the main challenges that Rainbow will tackle to address these issues are (in order of relevance): (i) methods and algorithms, mostly based on first-principle modeling and, when possible, on numerical methods for online/reactive trajectory generation, for enabling the robots with high autonomy; (ii) design and implementation of visual/haptic cues for interfacing the human operator with the robots, with a special attention to novel combinations of grounded/ungrounded (wearable) haptic devices; (iii) user and acceptability studies.

3.2. Main Components

Hereafter, a summary description of the four axes of research in Rainbow.

3.2.1. Optimal and Uncertainty-Aware Sensing

Future robots will need to have a large degree of autonomy for, e.g., interpreting the sensory data for accurate estimation of the robot and world state (which can possibly include the human users), and for devising motion plans able to take into account many constraints (actuation, sensor limitations, environment), including also the state estimation accuracy (i.e., how well the robot/environment state can be reconstructed from the sensed data). In this context, we will be particularly interested in (i) devising trajectory optimization strategies able to maximize some norm of the information gain gathered along the trajectory (and with the available sensors). This can be seen as an instance of Active Sensing, with the main focus on *online/reactive* trajectory optimization strategies able to take into account several requirements/constraints (sensing/actuation limitations, noise characteristics). We will also be interested in the coupling between optimal sensing and concurrent execution of additional tasks (e.g., navigation, manipulation). (ii) Formal methods for guaranteeing the accuracy of localization/state estimation in mobile robotics, mainly exploiting tools from interval analysis. The interest in these methods is their ability to provide possibly conservative but guaranteed accuracy bounds on the best accuracy one can obtain with the given robot/sensor pair, and can thus be used for planning purposes of for system design (choice of the best sensor suite for a given robot/task). (iii) Localization/tracking of objects with poor/unknown or deformable shape, which will be of paramount importance for allowing robots to estimate the state of "complex objects" (e.g., human tissues in medical robotics, elastic materials in manipulation) for controlling its pose/interaction with the objects of interest.

3.2.2. Advanced Sensor-based Control

One of the main competences of the previous Lagadic team has been, generally speaking, the topic of *sensor-based control*, i.e., how to exploit (typically onboard) sensors for controlling the motion of fixed/ground robots. The main emphasis has been in devising ways to directly couple the robot motion with the sensor outputs in order to invert this mapping for driving the robots towards a configuration specified as a desired sensor reading (thus, directly in sensor space). This general idea has been applied to very different contexts: mainly standard vision (from which the Visual Servoing keyword), but also audio, ultrasound imaging, and RGB-D.

Use of sensors for controlling the robot motion will also clearly be a central topic of the Rainbow team too, since the use of (especially onboard) sensing is a main characteristics of any future robotics application (which should typically operate in unstructured environments, and thus mainly rely on its own ability to sense the world). We then naturally aim at making the best out of the previous Lagadic experience in sensor-based control for proposing new advanced ways of exploiting sensed data for, roughly speaking, controlling the motion of a robot. In this respect, we plan to work on the following topics: (i) "direct/dense methods" which try to directly exploit the raw sensory data in computing the control law for positioning/navigation tasks. The advantages of these methods is the need for little data pre-processing which can minimize feature extraction errors and, in general, improve the overall robustness/accuracy (since all the available data is used by the motion controller); (ii) sensor-based interaction with objects of unknown/deformable shapes, for gaining the ability to manipulate, e.g., flexible objects from the acquired sensed data (e.g., controlling online a needle being inserted in a flexible tissue); (iii) sensor-based model predictive control, by developing *online/reactive* trajectory optimization methods able to plan feasible trajectories for robots subjects to sensing/actuation constraints with the possibility of (onboard) sensing for continuously replanning (over some future time horizon) the optimal trajectory. These methods will play an important role when dealing with complex robots affected by complex sensing/actuation constraints, for which pure reactive strategies (as in most of the previous Lagadic works) are not effective. Furthermore, the coupling with the aforementioned optimal sensing will also be considered; (iv) multi-robot decentralised estimation and control, with the aim of devising again sensor-based strategies for groups of multiple robots needing to maintain a formation or perform navigation/manipulation tasks. Here, the challenges come from the need of devising "simple" decentralized and scalable control strategies under the presence of complex sensing constraints (e.g., when using onboard cameras, limited fov, occlusions). Also, the need of locally estimating global quantities (e.g., common frame of reference, global property of the formation such as connectivity or rigidity) will also be a line of active research.

3.2.3. Haptics for Robotics Applications

In the envisaged *shared* cooperation between human users and robots, the typical sensory channel (besides vision) exploited to inform the human users is most often the force/kinesthetic one (in general, the sense of touch and of applied forces to the human hand or limbs). Therefore, a part of our activities will be devoted to study and advance the use of *haptic* cueing algorithms and interfaces for providing a feedback to the users during the execution of some shared task. We will consider: (i) multi-modal haptic cueing for general teleoperation applications, by studying how to convey information through the kinesthetic and cutaneous channels. Indeed, most haptic-enabled applications typically only involve kinesthetic cues, e.g., the forces/torques that can be felt by grasping a force-feedback joystick/device. These cues are very informative about, e.g., preferred/forbidden motion directions, but are also inherently limited in their resolution since the kinesthetic channel can easily become overloaded (when too much information is compressed in a single cue). In recent years, the arise of novel cutaneous devices able to, e.g., provide vibro-tactile feedback on the fingertips or skin, has proven to be a viable solution to *complement* the classical kinesthetic channel. We will then study how to combine these two sensory modalities for different prototypical application scenarios, e.g., 6-dof teleoperation of manipulator arms, virtual fixtures approaches, and remote manipulation of (possibly deformable) objects; (ii) in the particular context of medical robotics, we plan to address the problem of providing haptic cues for typical medical robotics tasks, such as semi-autonomous needle insertion and robot surgery by exploring the use of kinesthetic feedback for rendering the mechanical properties of the tissues, and vibrotactile feedback for providing with guiding information about pre-planned paths (with the aim of increasing the usability/acceptability of this technology in the medical domain); (iii) finally, in the context of multi-robot control we would like to explore how to use the haptic channel for providing information about the status of *multiple* robots executing a navigation or manipulation task. In this case, the problem is (even more) how to map (or compress) information about many robots into a few haptic cues. We plan to use specialized devices, such as actuated exoskeleton gloves able to provide cues to each fingertip of a human hand, or to resort to "compression" methods inspired by the *hand postural synergies* for providing coordinated cues representative of a few (but complex) motions of the multi-robot group, e.g., coordinated motions (translations/expansions/rotations) or collective grasping/transporting.

3.2.4. Shared Control of Complex Robotics Systems

This final and main research axis will exploit the **methods, algorithms and technologies** developed in the previous axes for realizing applications involving complex semi-autonomous robots operating in complex environments together with human users. The *leitmotiv* is to realize advanced *shared control* paradigms, which essentially aim at blending robot autonomy and user's intervention in an optimal way for exploiting the best of both worlds (robot accuracy/sensing/mobility/strength and human's cognitive capabilities). A common theme will be the issue of where to "draw the line" between robot autonomy and human intervention: obviously, there is no general answer, and any design choice will depend on the particular task at hand and/or on the technological/algorithmic possibilities of the robotic system under consideration.

A *prototypical* envisaged application, exploiting and combining the previous three research axes, is as follows: a complex robot (e.g., a two-arm system, a humanoid robot, a multi-UAV group) needs to operate in an environment exploiting its onboard sensors (in general, vision as the main exteroceptive one) and deal with many constraints (limited actuation, limited sensing, complex kinematics/dynamics, obstacle avoidance, interaction with difficult-to-model entities such as surrounding people, and so on). The robot must then possess a quite large autonomy for interpreting and exploiting the sensed data in order to estimate its own state and the environment one ("**Optimal and Uncertainty-Aware Sensing**" axis), and for planning its motion in order to fulfil the task (e.g., navigation, manipulation) by coping with all the robot/environment constraints. Therefore, advanced control methods able to exploit the sensory data at its most, and able to cope *online* with constraints in an optimal way (by, e.g., continuously replanning and predicting over a future time horizon) will be needed ("**Advanced Sensor-based Control**" axis), with a possible (and interesting) coupling with the sensing part for optimizing, at the same time, the state estimation process. Finally, a human operator will typically be in charge of providing high-level commands (e.g., where to go, what to look at, what to grasp and where) that will then be autonomously executed by the robot, with possible local modifications because of the various (local) constraints. At the same time, the operator will also receive *online* visual-force cues informative of, in general, how well her/his commands are executed and if the robot would prefer or suggest other plans (because of the local constraints that are not of the operator's concern). This information will have to be visually and haptically rendered with an optimal combination of cues that will depend on the particular application ("**Haptics for Robotics Applications**" axis).

SIROCCO Project-Team

3. Research Program

3.1. Introduction

The research activities on analysis, compression and communication of visual data mostly rely on tools and formalisms from the areas of statistical image modelling, of signal processing, of machine learning, of coding and information theory. Some of the proposed research axes are also based on scientific foundations of computer vision (e.g. multi-view modelling and coding). We have limited this section to some tools which are central to the proposed research axes, but the design of complete compression and communication solutions obviously rely on a large number of other results in the areas of motion analysis, transform design, entropy code design, etc which cannot be all described here.

3.2. Data Dimensionality Reduction

Manifolds, graph-based transforms, compressive sensing

Dimensionality reduction encompasses a variety of methods for low-dimensional data embedding, such as sparse and low rank models, random low-dimensional projections in a compressive sensing framework, and sparsifying transforms including graph-based transforms. These methods are the cornerstones of many visual data processing tasks (compression, inverse problems).

Sparse representations, *compressive sensing*, and *dictionary learning* have been shown to be powerful tools for efficient processing of visual data. The objective of *sparse representations* is to find a sparse approximation of a given input data. In theory, given $A \in \mathbb{R}^{m \times n}$, and $\mathbf{b} \in \mathbb{R}^m$ with $m << n$ and A is of full row rank, one seeks the solution of $\min\{\|\mathbf{x}\|_0 \ : \ A\mathbf{x} = \mathbf{b}\}$, where $\|\mathbf{x}\|_0$ denotes the L_0 norm of x, i.e. the number of non-zero components in z. A is known as the dictionary, its columns a_j are the atoms, they are assumed to be normalized in Euclidean norm. There exist many solutions x to $Ax = b$. The problem is to find the sparsest solution x, i.e. the one having the fewest non zero components. In practice, one actually seeks an approximate and thus even sparser solution which satisfies $\min\{\|\mathbf{x}\|_0 \ : \ \|A\mathbf{x} - \mathbf{b}\|_p \leq \rho\}$, for some $\rho \geq 0$, characterizing an admissible reconstruction error.

The recent theory of *compressive sensing*, in the context of discrete signals, can be seen as an effective dimensionality reduction technique. The idea behind compressive sensing is that a signal can be accurately recovered from a small number of linear measurements, at a rate much smaller than what is commonly prescribed by the Shannon-Nyquist theorem, provided that it is sparse or compressible in a known basis. Compressed sensing has emerged as a powerful framework for signal acquisition and sensor design, with a number of open issues such as learning the basis in which the signal is sparse, with the help of dictionary learning methods, or the design and optimization of the sensing matrix. The problem is in particular investigated in the context of light fields acquisition, aiming at novel camera design with the goal of offering a good trade-off between spatial and angular resolution.

While most image and video processing methods have been developed for cartesian sampling grids, new imaging modalities (e.g. point clouds, light fields) call for representations on irregular supports that can be well represented by *graphs*. Reducing the dimensionality of such signals require designing novel transforms yielding compact signal representation. One example of transform is the Graph Fourier transform whose basis functions are given by the eigenvectors of the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is a diagonal degree matrix whose i^{th} diagonal element is equal to the sum of the weights of all edges incident to the node i. The eigenvectors of the Laplacian of the graph, also called laplacian eigenbases, are analogous to the Fourier bases in the Euclidean domain and allow representing the signal residing on the graph as a linear combination of eigenfunctions akin to Fourier Analysis. This transform is particular efficient for compacting smooth signals on the graph. The problems which therefore need to be addressed are (i) to define graph structures on which the corresponding signals are smooth for different imaging modalities and (ii) the design of transforms compacting well the signal energy with a tractable computational complexity.

3.3. Deep neural networks

Autoencoders, Neural Networks, Recurrent Neural Networks

From dictionary learning which we have investigated a lot in the past, our activity is now evolving towards deep learning techniques which we are considering for dimensionality reduction. We address the problem of unsupervised learning of transforms and prediction operators that would be optimal in terms of energy compaction, considering autoencoders and neural network architectures.

An autoencoder is a neural network with an encoder g_e, parametrized by θ, that computes a representation Y from the data X, and a decoder g_d, parametrized by ϕ, that gives a reconstruction $\widehat{X}$ of X (see Figure below). Autoencoders can be used for dimensionality reduction, compression, denoising. When it is used for compression, the representation need to be quantized, leading to a quantized representation $\widehat{Y} = Q(Y)$ (see Figure below). If an autoencoder has fully-connected layers, the architecture, and the number of parameters to be learned, depends on the image size. Hence one autoencoder has to be trained per image size, which poses problems in terms of genericity.

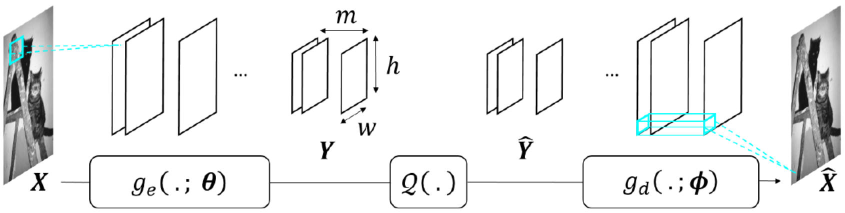
m
h
w
...
...
$\boldsymbol{Y}$
$\widehat{\boldsymbol{Y}}$
$\boldsymbol{X}$
$g_e(.;\boldsymbol{\theta})$
$Q(.)$
$g_d(.;\boldsymbol{\phi})$
$\widehat{\boldsymbol{X}}$

Figure 1. Illustration of an autoencoder.

To avoid this limitation, architectures without fully-connected layer and comprising instead convolutional layers and non-linear operators, forming convolutional neural networks (CNN) may be preferrable. The obtained representation is thus a set of so-called feature maps.

The other problems that we address with the help of neural networks are scene geometry and scene flow estimation, view synthesis, prediction and interpolation with various imaging modalities. The problems are posed either as supervised or unsupervised learning tasks. Our scope of investigation includes autoencoders, convolutional networks, variational autoencoders and generative adversarial networks (GAN) but also recurrent networks and in particular Long Short Term Memory (LSTM) networks. Recurrent neural networks attempting to model time or sequence dependent behaviour, by feeding back the output of a neural network layer at time t to the input of the same network layer at time t+1, have been shown to be interesting tools for temporal frame prediction. LSTMs are particular cases of recurrent networks made of cells composed of three types of neural layers called gates.

Deep neural networks have also been shown to be very promising for solving inverse problems (e.g. super-resolution, sparse recovery in a compressive sensing framework, inpainting) in image processing. Variational autoencoders, generative adversarial networks (GAN), learn, from a set of examples, the latent space or the manifold in which the images, that we search to recover, reside. The inverse problems can be re-formulated using a regularization in the latent space learned by the network. For the needs of the regularization, the learned latent space may need to verify certain properties such as preserving distances or neighborhood of the input space, or in terms of statistical modelling. GANs, trained to produce images that are plausible, are also useful tools for learning texture models, expressed via the filters of the network, that can be used for solving problems like inpainting or view synthesis.

3.4. Coding theory

OPTA limit (Optimum Performance Theoretically Attainable), Rate allocation, Rate-Distortion optimization, lossy coding, joint source-channel coding multiple description coding, channel modelization, oversampled frame expansions, error correcting codes.

Source coding and channel coding theory [0] is central to our compression and communication activities, in particular to the design of entropy codes and of error correcting codes. Another field in coding theory which has emerged in the context of sensor networks is Distributed Source Coding (DSC). It refers to the compression of correlated signals captured by different sensors which do not communicate between themselves. All the signals captured are compressed independently and transmitted to a central base station which has the capability to decode them jointly. DSC finds its foundation in the seminal Slepian-Wolf [0] (SW) and Wyner-Ziv [0] (WZ) theorems. Let us consider two binary correlated sources X and Y. If the two coders communicate, it is well known from Shannon's theory that the minimum lossless rate for X and Y is given by the joint entropy $H(X,Y)$. Slepian and Wolf have established in 1973 that this lossless compression rate bound can be approached with a vanishing error probability for long sequences, even if the two sources are coded separately, provided that they are decoded jointly and that their correlation is known to both the encoder and the decoder.

In 1976, Wyner and Ziv considered the problem of coding of two correlated sources X and Y, with respect to a fidelity criterion. They have established the rate-distortion function $R*_{X|Y}(D)$ for the case where the side information Y is perfectly known to the decoder only. For a given target distortion D, $R*_{X|Y}(D)$ in general verifies $R_{X|Y}(D) \leq R*_{X|Y}(D) \leq R_X(D)$, where $R_{X|Y}(D)$ is the rate required to encode X if Y is available to both the encoder and the decoder, and R_X is the minimal rate for encoding X without SI. These results give achievable rate bounds, however the design of codes and practical solutions for compression and communication applications remain a widely open issue.

[0] T. M. Cover and J. A. Thomas, Elements of Information Theory, Second Edition, July 2006.

[0] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources." IEEE Transactions on Information Theory, 19(4), pp. 471-480, July 1973.

[0] A. Wyner and J. Ziv, "The rate-distortion function for source coding ith side information at the decoder." IEEE Transactions on Information Theory, pp. 1-10, January 1976.