INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team DaRT

Dataparallelism for Real-Time

Futurs

THEME COM

Table of contents

1. Team

DaRT is a common project with the University of Science and Technologies of Lille (USTL), via the Laboratory of Fundamental Computer Science of Lille (LIFL, associated to the CNRS as UMR 8022).

Head of project-team
Jean-Luc Dekeyser [Professor, Université des Sciences et Technologies de Lille, HdR]

Vice-head of project-team
Pierre Boulet [Professor, Université des Sciences et Technologies de Lille, HdR]

Administrative assistant
Karine Lewandowski [INRIA]

Faculty members
Cédric Dumoulin [Associate professor, Université des Sciences et Technologies de Lille]
Anne Etien [Associate professor, Polytech Lille, since sept. 2007]
Frédéric Guyomarc'h [Research Associate (secondment INRIA)]
Philippe Marquet [Associate professor, Université des Sciences et Technologies de Lille]
Samy Meftali [Associate professor, Université des Sciences et Technologies de Lille]

Research scientist
Abdoulaye Gamatié [Research scientist, CNRS]

External Collaborator
Smaïl Niar [Associate Professor, Université de Valenciennes et du Hainaut-Cambrésis, HdR]

Post-doctoral fellows
Anne Etien [Post-doctoral fellow, INRIA Futurs, until august 2007]
César Moura [Post-doctoral fellow, INRIA Futurs]
Emmanuel Renaux [Post-doctoral fellow, INRIA Futurs]

Ph.D. students
Adolf Abdallah [French Minister grant]
Yassine Aydi [ENIS Sfax]
Mouna Baklouti [co-advising USTL-ENIS Sfax]
Rabie Ben Atitallah [Interreg project grant]
Hajer Chtioui [co-advising UVHC-ENIS Sfax]
Calin Glitia [INRIA Futurs grant]
Souha Kamoun [French Minister grant]
Sébastien Le Beux [Interreg project grant]
Éric Piel [INRIA Futurs and Regional grant]
Imran Rafiq Quadri [HEC Pakistan grant]
Safouan Taha [co-advising USTL-CEA, CEA grant]
Julien Taillard [VALEO and Regional grant]
Huafeng Yu [INRIA Futurs and Regional grant]

Technical staff
Antoine Honoré [Expert Engineer]
Cyril Joly [Expert Engineer]

Visiting scientists
Abou El Hassan Benyamina [Algerian Government grant]

2. Overall Objectives

2.1. Introduction

For the last few years we have seen the beginning of the “design gap”. This gap is caused by the exponential growth of the integration rate of transistors on chips and the comparatively slower growth of the productivity of the integrated circuits designers. It is now impractical to fill a chip with custom designed logic. One has to reuse existing design parts or fill the chip area with memory (a good example of this evolution is the multi-core processors that include several existing processing cores instead of complexifying a single core). This evolution is clearly attested by the International Technology Roadmap on semiconductors.

In the same time, the computing power requirements of intensive signal processing applications such as video processing, voice recognition, telecommunications, radar or sonar are steadily increasing (several hundreds of Gops for low power embedded systems in a few years). If the design productivity does not increase dramatically, the limiting factor of the growth of the semiconductor industry will not be the physical limitations due to the thinness of the fabrication process but the economy! Indeed we ask to the system design teams to build more complex systems faster, cheaper, bug free and decreasing the power consumption...

We propose in the DaRT project to contribute to the improvement of the productivity of the electronic embedded system design teams. We structure our approach around a few key ideas:

- Promote the use of *parallelism* to help reduce the power consumption while improving the performance.
- *Separate the concerns* in different models to allow reuse of these models and to keep them human readable.
- Propose an environment starting at the highest level of abstraction, namely the *system modeling* level.
- *Automate code production* by the use of (semi)-automatic *model transformations* to build correct by construction code.
- Develop *simulation techniques* at precise abstraction levels (functional, transactional or register transfer levels) to check the design the soonest.
- Promote *strong semantics* in the application model to allow verification, non ambiguous design and automatic code generation.
- Focus on a *limited application domain*, intensive signal processing applications. This restriction allows us to push our developments further without having to deal with the wide variety of applications.

All these ideas are implemented into a prototype co-design environment based on a model driven engineering approach, Gaspard. This open source platform is our test bench and is freely available.

The main technologies we promote are UML 2 [39], MDE [76] and Eclipse EMF [55] for the modeling and model handling; Array-OL [49], [50], [47], [46] and synchronous languages [43] as computation models with strong semantics for verification; SystemC [73] for the simulation; OpenMP for the shared memory parallel execution; VHDL for the synthesis; and Java to code our prototypes.

2.2. Highlights of the year

- The MARTE (Modeling and Analyzis of Real-Time and Embedded systems) UML profile has been standardized by the OMG in July 2007. We have contributed to this standard through our participation to the Promarte submission team where our contributions are the writing of the Repetitive Structure Modeling chapter and a participation to the overall document structure and specially to the allocation modeling, the general component modeling and the hardware resource modeling chapters. The Gaspard2 profile we have been working on these 4 last years has greatly influenced the MARTE standard. We currently participate to the finalization task force that will take the current Beta document to a full standard next year.

- A codesign process for *high-performance system-on-Chip* (HP-SoC) has been developed within the Gaspard2 platform, which adopts the principles of *model-driven engineering* (MDE). It allows one to generate different target code sources from high-level models of systems:
  - synchronous declarative languages for formal validation,
  - OpenMP for intensive numerical simulations,
  - transactional level SystemC code (PVT) for HP-SoC shared memory systems. This executable code car be simulated in a fast way, and permits having interesting performance estimations of criteria as communication over the system
  - and VHDL code that enables to simulate or synthesize a Gaspard2 application. We particularly target FPGAs. Furthermore, an heuristic adapts the parallelism of the application to the characteristics (computing power and IOs) of a given FPGA.

2.3. Towards architecture exploration

The Gaspard2 environment is dedicated to hardware/software co-design of dataflow dominated systems implemented on a single chip. To help the designer, such an environment should include a new approach for the evaluation of alternative software/hardware solutions. Several architectural solutions as well as several application specifications have to be evaluated with regard to their performance and cost. The goal is to select the best solution for the application, the architecture and the association of the former on the latter that satisfies the timing constraints and minimizes the cost.

The research activities we are achieving already generate metrics from SystemC simulations and the refactoring algorithm defined for the transformation of loops to particular multiprocessors are the first steps for exploration. The system exploration phase is implemented as an iterative process directed by the user. To make an automatic exploration system, we also need to be able to transform the multiprocessor description (size, network, memory) and to apply a multi-objective method. The space of solutions is huge and a fast simulation in SystemC at a high abstraction level is a good opportunity to reduce the space in a short delay. After that, a precise simulation at low level in SystemC or even in VHDL can start to refine the solution.

3. Scientific Foundations

3.1. Introduction

ISP Intensive Signal Processing
SoC System-on-Chip

These last few years, our research activities are mainly concerned with data parallel models and compilation techniques. Intensive Signal Processing (ISP) with real time constraints is a particular domain that could benefit from this background. Our project covers the following new trend: a data parallel paradigm for ISP applications. These applications are mostly developed on embedded systems with high performance processing units like DSP or SIMD processors. We focus on multi processor architectures on a single chip (System-on-Chip). To reduce the "time to market", the DaRT project proposes a high level modeling environment for software and hardware design. This level of abstraction already allows the use of verification techniques before any prototyping (as in the Esterel Studio environment from Esterel Technologies [56]). This also permits to automatically produce a mapping and a schedule of the application onto the architecture with code generation (as with the AAA method of SynDEx [77]). The DaRT project contributes to this research field by the three following items:

Co-modeling for HP-SoC design: We define our own metamodels to specify application, architecture, and (software hardware) association. These metamodels present new characteristics as high level data parallel constructions, iterative dependency expression, data flow and control flow mixing, hierarchical and repetitive application and architecture models. All these metamodels are implemented with EMF. Some of them are represented as UML profiles.

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

SoC simulation, verification and synthesis: The data flow philosophy of our metamodel is particularly well suited to a distributed simulation. Thus, we developped a SystemC based simulation environment at two abstraction levels. CABA (Cycle accurate and bit accurate) for accurate performance estimations (power consumption and execution time), and PVT (system level) for fast simulations of complex HP SIMD systems.

To take care of the architecture model and the mapping of the application on it, we use the SystemC platform to simulate at different levels of abstraction the result of the SoC design. This simulation allows to verify the adequacy of the mapping and the schedule (communication delay, load balancing, memory allocation...). We also support IP (Intellectual Property) integration with different levels of specification.

On the other hand, we have been making significant efforts towards the use of formal validation techniques in order to ensure the correctness of designed systems. For that, we particularly consider the synchronous approach.

3.2. Co-modeling for HP-SoC design

Keywords: *MDE*, *Metamodel*, *Model*, *Modeling*, *Transformation*, *UML.*

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

3.2.1. System-on-Chip Design

SoC (System-on-Chip) can be considered as a particular case of embedded systems. SoC design covers a lot of different viewpoints including the application modeling by the aggregation of functional components, the assembly of existing physical components, the verification and the simulation of the modeled system, and the synthesis of a complete end-product integrated into a single chip. As a rule, a SoC includes programmable processors, memory units (data/instructions), interconnection mechanisms and hardware functional units (Digital Signal Processors, application specific circuits). These components can be generated for a particular application; they can also be obtained from IP (Intellectual Property) providers. The ability to re-use software or hardware components is without any doubt a major asset for a codesign system.

The multiplicity of the abstraction levels is appropriate to the modeling approach. The information is used with a different viewpoint for each abstraction level. This information is defined only once in a single model. The links or transformation rules between the abstraction levels permit the re-use of the concepts for a different purpose.

3.2.2. Model-driven engineering

Because of the vast scope of the encountered problems, of the quick evolution of the architectures, we observe a very great diversity regarding programming languages. Ten years ago each new proposed model (for example within the framework of a PhD) led to the implementation of this model in a new language or at least in an extension of a standard language. Thus a variety of dialects were born, without relieving the programmer of the usual constraints of code development. Portability of an application from one language to another (a new one for example) increases the workload of the programmer. This drawback is also true for the development of embedded applications. It is even worse, because the number of abstraction levels has to be added to the diversity of the languages. It is essential to associate a target hardware architecture model to the application specification model, and to introduce also a relationship between them. These two models are practically always different, they are often expressed in two different languages.

From this experience, one can derive some principles for the design of the next generation of environments for embedded application development:

- Design application and architecture with the same language;
- Reuse/perpetuate developed models with actual and upcoming simulation and synthesis tools;
- Provide one single modeling environment for the whole co-design process, possibly supporting a visual specification;
- Benefit from standard formats for exchange and storage.
- Be able to express transformation rules from model to model.

We believe that the Model Driven Engineering [76] can allow us to propose a new method of system design respecting these principles. Indeed, it is based on the common UML modeling language to model all kinds of artifacts.

The Model Driven Engineering (MDE) approach advocates the use of models at different levels of abstraction. A model represents an abstract view of the reality, it is defined by a metamodel specifying the available concepts and the relationship between them. A common MDE development process is to start from a high level of abstraction and to go to a targeted model by flowing through intermediate levels of abstraction. Usually, the high level models contain only domain specific concepts, while technological concepts are introduced smoothly in the intermediate levels. The targeted levels are used for different purposes: code generation, simulation, verification, or as inputs to produce other models, etc.

The clear separation between the high level models and the technological models makes it easy to switch to a new technology while re-using the old designs. Transformations allow to go from one model at a given abstraction level to another model at another level, and to keep the different models synchronized. Related models are described by their metamodels, on which we can define some mapping rules describing how concepts from one metamodel are to be mapped on the concepts of the other metamodel. From these mapping rules the transformations between any models conforming to the metamodels are deduced.

The MDE is based on proved standards: UML 2 [38] for modeling and the MOF (Meta Object Facilities [72]) for metamodel expression. Some profiles, i.e. UML extensions, have been defined in order to express the specificities of a particular domain. In the context of embedded system, the MARTE profile in which we contributed follows the OMG standardisation process. From a technical view point, metamodels and models are realized with the Eclipse ECore modeling language, an implementation of MOF provided by EMF [55].

3.2.3. Models of computation

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

3.2.3.1. Array-OL

The Array-OL language [49], [50], [47], [46] gives a mixed graphical-textual language dedicated to express multidimensional intensive signal processing applications. It relies on the following basic principles:

- All the potential parallelism in an application has to be available in the specification, both *task parallelism* and *data parallelism*.
- Array-OL is a *data dependence expression* language. Only the true data dependencies are expressed in order to express the full parallelism of the application, defining the minimal order of the tasks. Thus any schedule respecting these dependencies will lead to the same result. The language is deterministic.
- It is a *single assignment* formalism. No data element is ever written twice. It can be read several times, though. Array-OL can be considered as a first order functional language.

- The language is *hierarchical* to allow descriptions at different granularity levels and to handle the complexity of the applications. The data dependencies expressed at a level (between arrays) are approximations of the precise dependencies of the sub-levels (between patterns).
- The only data-structure is the multidimensionnal array. Data accesses are done through sub arrays, via a tiling of the arrays.
- The spatial and temporal dimensions are treated equally in the arrays. In particular, time is expanded as a dimension (or several) of the arrays. This is a consequence of single assignment.
- The arrays are toroidal. Indeed, some spatial dimensions may represent some physical tori (think about some hydrophones around a submarine) and some frequency domains obtained by FFTs (Fast Fourier Transformations) are toroidal.

The semantics of Array-OL is that of a first order functional language manipulating multidimensional arrays. It is not a data flow language but can be projected on such a language.

3.2.3.2. Synchronous model

The synchronous approach [43] proposes formal concepts that favor the trusted design of embedded real-time systems. Its basic assumption is that computation and communication are instantaneous (referred to as "synchrony hypothesis"). The execution of a system is seen through the chronology and simultaneity of observed events. This is a main difference from visions where the system execution is rather considered under its chronometric aspect (i.e., duration has a significant role).

There are different synchronous languages, which have strong mathematical foundations. These languages mainly differ from their programming styles. For instance, the Esterel language [44] adopts an imperative style. It is well-suited for the design of control dominated applications. Other languages such as Lustre [48] or Signal [65] rather adopt a declarative style. More specifically, Lustre is functional while Signal is relational. These two languages are well-adapted for dataflow-oriented applications. All these languages are associated with formal tool-sets that have been successfully used in several critical domains (e.g. avionics, automotive, nuclear power plants, etc.).

In the context of the DaRT project, we consider the last family of languages (i.e. declarative languages) to model various refinements of Array-OL descriptions in order to deal with the control aspect as well as the temporal aspect present in target applications. The first aspect is typically addressed by using concepts such as mode automata, which are proposed as an extension mechanism in synchronous declarative languages. The second aspect is studied by considering temporal projections of array dimensions in synchronous languages based on clock notion.

The resulting synchronous models can be then analyzed using the formal techniques and tools provided by the synchronous technology.

3.2.4. Contributions of the team

Our proposal is partially based upon the concepts of the "Y-chart" [57]. The MDE contributes to express the model transformations which correspond to successive refinements between the abstraction levels.

Metamodeling brings a set of tools which will enable us to specify our application and hardware architecture models using UML tools, to reuse functional and physical IPs, to ensure refinements between abstraction levels via mapping rules, to ensure interoperability between the different abstraction levels used in a same codesign, and to ensure the opening to other tools, like verification tools, thought the use of standards.

The application and hardware architecture are described by different metamodels. Some concepts from these two metamodels are similar in order to unify and so simplify their understanding and use. Models for application and hardware architecture may be done separately (maybe by two different people). At this point, it becomes possible to map the application model on the hardware architecture model. A third metamodel, called association metamodel, allows us to express this mapping.

All the previously defined models, application, architecture and association, are platform independent. No component is associated with an execution, simulation or synthesis technology. Such an association targets a given technology (OpenMP, SystemC/PA, VHDL, Lustre, etc.). Once all the components are associated with some technology, the deployment is realized.

This is done by the refinement of the association model to the deployed model first, and then to further abstraction levels (e.g. polyhedron, loop in figure 1).

The diversity of the technologies requires interoperability between abstraction levels and simulation and execution languages. For this purpose we define an interoperability metamodel allowing to model interfaces between different abstraction levels.

Mapping rules between the deployment metamodel, and interoperability and technology metamodels can be defined to automatically specialize the deployment model to the chosen abstraction levels. From each of the resulting models we could automatically produce the execution/simulation code and the interoperability infrastructure.

The simulation results can lead to a refinement of the application, the hardware architecture, the association or the deployment models. We propose a methodology to work with these models. The stages of design could be:

1. Separation of application and hardware architecture modeling.
2. Association with semi-automatic mapping and scheduling.
3. Selection of IP for each element of application/architecture models.
4. Deployment (choice of simulation or execution level and platform for each component).
5. Automatic generation of the various platform specific simulation or execution models.
6. Automatic simulation or execution code generation.
7. Refinement at the highest level given the simulation results.

3.2.4.1. Gaspard2 foundations

The abstract syntax of application and hardware architecture are described by different MOF metamodels. Some concepts from these two meta-models are similar, in order to simplify their understanding and use.

They share a common modelling paradigm, the component oriented approach, to ease reusability. Reusability is one of the key point to face the time to market challenge faced by the designers of embedded systems.

The two meta-models also share some common construction mechanisms, to express repetitive constructs in a compact way. This kind of compact expression makes them more comprehensible for a compiler or an optimisation tool.

To express the mapping of an application model on a hardware architecture model, a third meta-model named association is introduced. This meta-model imports the concepts of the two previously mentioned meta-models.

The definition of the Gaspard2 profile (a detailed specification can be found in [34]) is oriented by the few guiding ideas mentioned in section 2. It offers a high level modeling environment for high performance systems on chip, and includes UML extensions (formally the UML view point of the profile) as well as an abstract syntax expressed in EMOF (Domain view point). The main evolutions according to past versions of the profile are:

- a better structuring of the various packages, in order to define as much common parts as possible for the various aspects of our Y approach
- a better alignment with related OMG standards (SPT, QoS, SysML, MARTE).

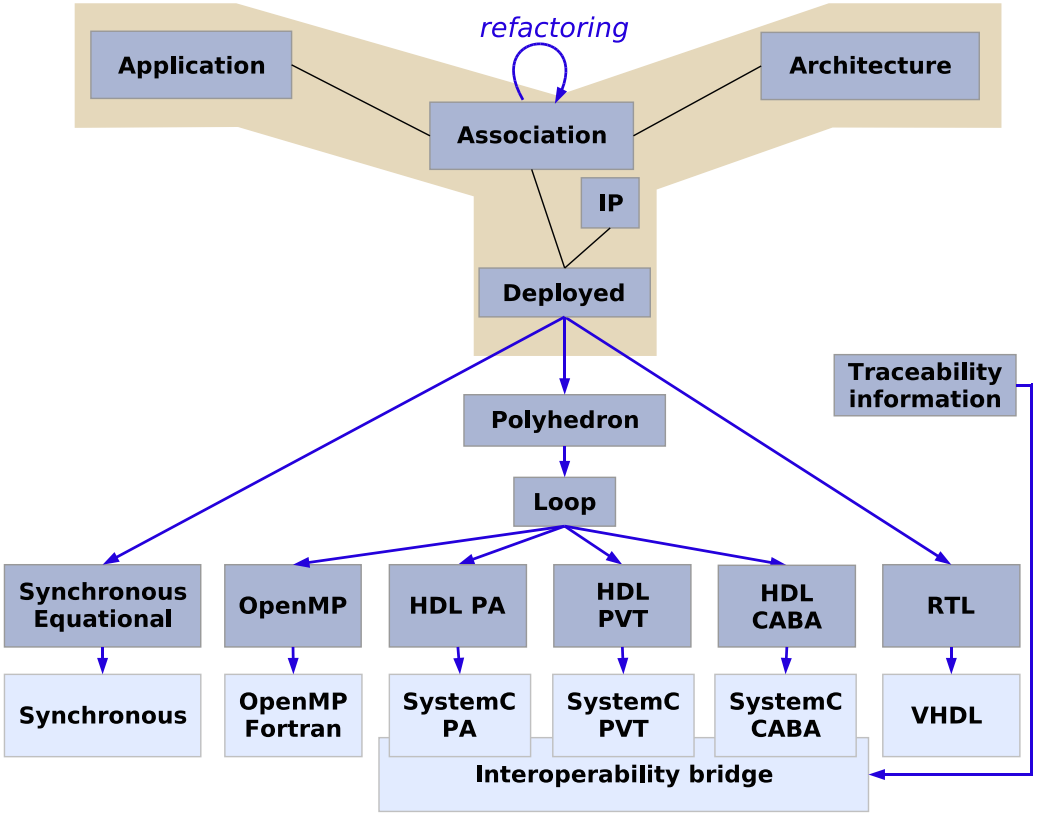

Figure 1. Overview of the metamodels for the "Y" design

3.2.4.2. Application metamodel

The application metamodel focuses on the description of data dependencies between components. Components and dependencies completely describe an algorithm without addition of any parasitic information. Actually any compilation optimization or parallelization technique must respect the data dependencies. This gives many benefits:

- simple description of the algorithm,
- no dependency analysis in the compiler,
- all the parallelism and optimization potential of the algorithm is easily available.

Application components represent some computation and their ports some data input and output capabilities. Data handled in the applications are mainly multidimensional arrays, with one possible infinite dimension representing time.

The application metamodel introduces three kinds of components : Compound, DataParallel, and ElementaryComponents.

A compound component expresses task parallelism by the way of a component graph. The edges of this graph are directed and represent data dependencies.

A data parallel component expresses data parallelism by the way of the parallel repetition of an inner component part on patterns of the input arrays, producing patterns of the output arrays. Some rules must be respected to describe this repetition. In particular, the output patterns must tile exactly the output arrays.

Elementary components are the basic computation units of the application. They have to be defined for each target technology.

Finally, the application metamodel has been extended with a few concepts that allow one to describe reactive control based on the notion of execution mode.

3.2.4.3. Hardware architecture metamodel

The purpose of this meta-model is to satisfy the growing need of embedded system designers to specify the hardware architecture of the system at a high abstraction level. It enables to dimension the resources of the hardware in a precise enough way to be pertinent, while abstracting irrelevant details so that efficient decisions can be taken.

A mechanism similar to the one used in the application meta-model enables to specify repetitive architectures in a compact way. Regular parallel computation units are more and more present in embedded in systems, especially in SoCs. There are two main reasons for that:

1. Time-to-market constraints are becoming so tight that massive reuse of computation units is one of the only ways to get the computation power needed for next generation embedded applications.
2. Using massive parallel repetitions of simple computation units is the only way to get the required computation power density for high performance embedded applications.
3. Parallelism is a good way to reduce power consumption in SoCs. Indeed at equal computing power, a chip able to run several computations simultaneously is clocked at a lower frequency than a chip able to run less computations in a given cycle. As frequency is square in the power consumption equation, this leads to important gains.

The repetitive constructs we propose can be used to model parallel computation units, such as grids, but also complex static or dynamic interconnection networks, or memory banks.

3.2.4.4. Association metamodel

The association metamodel allows to express how the application is projected and scheduled on the architecture. This metamodel imports the application and architecture metamodels in order to associate their components. The association model associates application components with architecture components to express which hardware component executes which functionality. If the hardware component is programmable, the

application components associated with this component are implemented in software, otherwise, they are synthesized as hardware. The dependencies between application components are associated with communication routes. These routes are built as sequences of data paths, components and represent the route of data from one memory to another via processor or DMA initiated data exchanges. The input and output of the functional components are mapped into memories.

As the application and hardware architecture models, the association model takes advantage of a repetitive and hierarchical representation to allow to view the association at different granularities and to factorize its representation.

The association model is the input and the output of the optimization algorithm. Indeed, the optimization can be seen as a refactoring of the association model. Code transformations allow to refactor the application to map it more easily on the target hardware architecture. Once refactored, each application component can be mapped to a corresponding hardware architecture component. The repetitive application components can be distributed to a repetitive hardware architecture component. The association metamodel allows at least the regular mapping strategies of classical data-parallel languages as High Performance Fortran.

3.2.4.5. Deployment metamodel

The deployment metamodel has been identified last year. It allows one to describe the choice of IPs after the association phase in figure 1. It is now defined (see Section 6.1.2 for more details).

3.2.4.6. A metamodel based on polyhedra

In order to simplify and factorize the work on model transformations, a metamodel has been defined which is a common denominator of several of the outputs metamodels. Consequently, this metamodel permits representation of a Gaspard2 model, with no “loss” of information for the model implementations, while being at a lower level of abstraction. The goal of the metamodel is to make the notion of *association* implicit. Instead, the application part of the model is dispatched over the processors (for the tasks) and the memories (for the data). In order to represent the information of which repetition of a task is on which instance of a processor, expressed by the user using the association, the metamodel uses the concepts of polyhedra parametrized by the processor indices. Polyhedron representation has been selected for, at the same time, being able to represent every possible association written by the user and being easily transformed into code.

3.2.4.7. A metamodel based on loops

This metamodel is used after the one based on polyhedra. It is the last common metamodel before targeting implementation platforms. This metamodel is closed to the polyhedron one. It is used to refine model toward code generation. Polyhedra expressions are transformed into loops expressions using CLooG (Chunky Loop Generator) [42]. CLooG is a tool that generates loops for scanning Z-polyhedra. For each polyhedron, CLooG is called by transformation rules and polyhedra information are replaced by LoopStatements in the loop metamodel. Loops, which are parametrized by the processor indices, indicate which repetition of a task is executed by which processor. So each processor has the same code. Then, we can target SoC simulation or high performance computing.

3.2.4.8. A metamodel for procedural language with OpenMP statement

We focused on shared memory computers in order to first simplify the transformations. So we do not have to manage communication between processors. This metamodel is used before code generation for high performance computing. It permits the representation of a code written with procedural language (Fortran and C) with OpenMP statement [74]. So, we have a representation close to the code, which can be used to target different languages. Finally, code generation is a just "pretty printer" for this metamodel.

3.2.4.9. A RTL metamodel

The RTL metamodel proposed in Gaspard2 is used for the generation of VHDL code, which is further synthesized on FPGAs. It is entirely defined this year (see Section 6.2.1 for more details).

3.2.4.10. Synchronous metamodel

The metamodel for synchronous languages serves as a common format for automatic code generation in declarative languages such as Lustre, Lucid Synchrone and Signal. Its implementation has been made complete this year (see Section 6.2.2 for more details).

3.3. Model-based optimization and compilation techniques

Keywords: *Compilation*, *Dataparallelism*, *Heuristics*, *Mapping*, *Optimization*, *Power Consumption*, *Scheduling*.

3.3.1. Optimization for parallelism

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

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

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

3.3.2. Contributions of the team

We focus on two particular subjects in the optimization field: dataparallelism efficient utilization and multi-objective hierarchical heuristics.

3.3.2.1. Dataparallel code transformations

We have studied Array-OL to Array-OL code transformations [47], [78], [52], [51], [53] [58]. Array-OL allows a powerful expression of the data access patterns in such applications and a complete parallelism expression. It is at the heart of our model of application, hardware architecture and association.

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

We have studied the relations of the Array-OL model with other computation models [41] such as Kahn Process Networks [60], [61] and multidimensional synchronous dataflow [69], [68].

We pursue the study of such transformations with three objectives:

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

3.3.2.2. Multi-objective hierarchical scheduling heuristics

When dealing with complex heterogeneous hardware architectures, the scheduling heuristics usually take a task dependence graph as input. Both our application and hardware architecture models are hierarchical and allow repetitive expressions. We propose a Globally Irregular, Locally Regular combination of heuristics to allow to take advantage of both task and data parallelism [66] and have stated evaluation evolutionary heuristics in this context.

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

In embedded systems, minimizing the latency of the application is usually not the good objective function. Indeed, one must reach some real time constraints but it is not useful to run faster than these constraints. It would be more interesting to improve the resource usage to decrease the power consumption or the cost of the hardware architecture. Various techniques exist to reduce power consumption in embedded systems. This research covers:

- The evaluation of the impact of cache management schemas on power consumption [70], [75].
- The study of code compression techniques to reduce the memory requirements of an embedded application [59].
- Clock scaling to choose the slowest speed that satisfies the real-time constraints.

3.3.2.3. Transformation techniques

Model to model transformations are at the heart of the MDE approach. Anyone wishing to use MDE in its projects is sooner or later facing the question: how to perform the model transformations? The standardization process of Query View Transformation [71] was the opportunity for the development of transformation engine as ModTransf. However, since the standard has been published, only few of investigating tools, such as ATL [1] (a transformation dedicated tool) or Kermeta [2] (a generalist tool with facilities to manipulate models) remained compliant with the standard and are powerful enough to fulfil execution needs that we had identified.

In front of the absence of transformation tool that supports external black boxe calls (e.g. native function calls), recursive rule call, rule inheritancy and integration of imperative code, we developped our own tool MoMoTE. MoMoTE is a Java framework defined to enhance model to model transformations. As suggested in the QVT standard, transformations are composed of rules for which the application condition is defined by a query and the model parts to create is specified with traditional model structures.

An other tool MoCode has also been defined, for the model to text transformation (i.e. the code generation). It relies on JET principles. It takes in charge the calls of the right templates as well as the link between the JET and the Ecore worlds.

We also focus on another difficulty encountered in the design of the Gaspard2 supply chain: document transformations. Indeed, transformations are widely encountered in embedded system product chains; each metamodel of the chain involves different and complex concerns; and the exchange on the design of transformation in a co-design context becomes a real need. We, thus, put the next question under the spotlight: how to design and document model transformations?

We proposed TrML, a graphical representation of the transformation to increase documentation, exchange around transformations. This notation fulfills all the following needs:

- graphical representation of model transformations;
- portable notation independent from any existing transformation engines;
- mechanisms to divide transformation into rules;
- focus on what should be transformed, rather than how to implement the transformation;
- black boxes mechanism to enable legacy code call;
- input and output metamodel extension to access complex construct in easier way or to define your own methods or attributes;
- precise definition and use of concepts, thanks to a profile and a metamodel;
- free way to implement the transformation, execution on top of ATL, Kermeta, QVT... are possible;
- standard way to specify the rules, facilitating exchange, comprehension, and documentation.

[1] http://www.eclipse.org/m2m/atl
[2] http://www.kermeta.org

3.3.2.4. Traceability

The Dart project promotes the reuse of IPs for the design of SoC. Such IPs come from different sources with heterogeneous models (different abstraction levels). This approach reduces « time to market », but its application requires new design methodologies.

Gaspard2 proposes a methodology based on Model Driven Engineering which intends the use of many simulation and execution platforms (Java, OpenMP, SystemC, VHDL, etc.) at different levels of abstraction (TLM, RTL, etc.). Models of different platforms and abstraction levels are generated in Gaspard2 by model transformations. The heterogeneity of the targeted platforms leads to an interoperability problem.

We propose a solution based on MDE model transformations techniques to provide a framework to deal with this interoperability problem [45]. This solution is performed in three steps. First, we introduce traceability in model transformations; therefore a trace model is generated along with model transformations. This trace model is then used as an entry model for a transformation which generates an interoperability bridge model. Finally, the code for interoperability is generated from the bridge model of the previous step. In order to automate the process, metamodels for traceability and interoperability bridge have been designed. We also provide the description of the different transformations involved in the process.

Traceability is also an important element for debugging and exploration phases. SystemC simulations generate metrics concerning performance and cost. To be able to transform the application or the architecture specifications, we need to find the original concepts that produce the piece of SystemC code where we observed the performances. Traceability has to facilitate the reserve step toward these specifications.

3.4. HP-SoC simulation, verification and synthesis

Keywords: *SystemC*, *TLM*.

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

SoCs are more and more complex and integrate software parts as well as specific hardware parts (IPs, Intellectual Properties). Generally before obtaining a SoC on silicium, a system is specified at several abstraction levels. Any system design flow consists in refining, more or less automatically, each model to obtain another, starting from a functional model to reach a Register Transfer Level model. One of the biggest design challenges is the development of a strong, low cost and fast simulation tool for system verification and simulation.

The DaRT project is concerned by the simulation at different levels of abstraction of the application/architecture co-model and of the mapping/schedule produced by the optimization phase.

3.4.1. Abstraction levels

Transactional Level modelling (TLM) appeared during the very few last years. It consists in describing systems according to the specifications of the TLM abstraction levels (briefly mentioned in Section 3.4.1). At these levels, communications use function calls (e.g. burst_read(char* buf, int addr, int len);). The major aims of TLM modelling are:

- Enable fast simulations and compact specifications
- Integrate HW and SW models
- Early platform for SW development
- Early system exploration and verification
- IPs reuse

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

The main stadard TLM Levels are :

Communicating Processes (CP): a model, at this level, is similar to an executable specification without any information about the hardware architecture. In fact, the system is composed by functions (processes) exchanging parameters. The communication between modules is point-to-point, and usually modelled using abstract channels.

Communicating Processes with Time (CP + T): it is similar to CP but timing delays are added to processes within the design to reflect the timing constraints of the specification.

Programmer's View (PV): at this level, hardware IPs composing the system's architecture appears in the model. Shared communication links should be modelled at PV level, but both behaviour and communication still untimed.

Programmer's View with Time (PV + T): it is similar to PV but timing delays are added to processes within the design to reflect the timing constraints of the specification and also to process delays of the target architecture. Processors, at this level, should be modelled using Instruction Set Processors (ISS).

Cycle Accurate, Bit Accurate (CABA): the internal structure accurately reflects the registers and the combinatorial logic of the target architecture. The communications are described in details in terms of used protocols and timing. Pins appear as in the physical components, and each module's behaviour corresponds exactly to the behaviour of the physical module, locally at each cycle.

Register Transfer Level (RTL): RTL models are very close to CABA ones, in terms of accuracy. The most important differences with CABA level are:

- In RTL the model is accurate both locally (in each cycle) and globally
- In RTL the description is synthesizable.

The five first levels are commonly called TLM levels, and the two first ones functional levels.

3.4.2. Contribution of the team

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

3.4.2.1. Co-simulation in SystemC

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

The SoC architecture is directly produced from the architecture model. A module in SystemC simulates the behaviour of tasks mapped to a particular processor. Other modules contain the data parallel structures and are able to answer to any read/write requests. The communications between tasks and between tasks and memories are simulated via communication modules in SystemC. These last modules produce interesting results concerning the simultaneous network conflicts and the capacity of this network for this application. The Loop metamodel allows automatic SystemC code generation. The association model is first transformed into a model of this Loop metamodel and this model is then automatically transformed into SystemC code. This development is integrated in the Gaspard2 prototype and uses the MoMoTE tool (see the software section).

3.4.2.2. TLM: Transactional level modelling

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

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

3.4.2.3. Abstraction levels in the simulation platform

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

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

3.4.2.4. Network on Chip (NoC) design and performances estimation

Modern SoCs are very complex and integrate more and more heterogeneous IPs. Due to this complexity, designers need high performance interconnection components. These latter have to be also, as much as possible, flexible to support new applications. This kind of interconnection IPs is unfortunately not available until today. In fact, designers still use buses and simple point-point connections in their designs.

Our contribution in this domain is the proposal of an open Network on Chip library for SoCs design. The NoCs will be mainly an adaptation, for embedded systems, of those proposed for classical multiprocessor architectures. Performances of these networks have been proved, and we believe that such a library will permit the integration of more and more IPs on a chip in a systematic way. This library will be also a support and a completion of existing open SystemC IP libraries as SoCLib. All its components will be OCP compliant [67].

3.4.2.5. Verification

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

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

We recently started a study that aims at integrating test-based approaches in the Gaspard2 framework in order to validate complex systems for which it is often tedious to find adequate abstractions that can be formally verified. We believe that such approaches are perfectly complementary with formal verification techniques.

4. Application Domains

4.1. Intensive signal processing

Keywords: *multimedia*, *telecommunications*.

The DaRT project aims to improve the design of embedded systems with a strong focus on intensive signal processing applications.

This application domain is the most intensive part of signal processing, composed of:

- systematic signal processing;
- intensive data processing.

Many signal and image processing applications follow this organisation: software radio receiver, sonar beam forming, or JPEG 2000 encoder/decoder.

In the framework of the ModEasy project, we also study computation intensive automotive safety embedded systems.

The systematic signal processing is the very first part of a signal processing application. It mainly consists of a chain of filters and regular processing applied on the input signals independently of the signal values. It results in a characterization of the input signals with values of interest.

The intensive data processing is the second part a of a signal processing application. It applies irregular computations on the values issued by the systematic signal processing. Those computations may depend on the signal values.

Below are three application examples from our industrial partners.

Software Radio Receiver This emerging application is structured in a front end systematic signal processing including signal digitalizing, channel selection, and application of filters to eliminate interferences. These first data are decoded in a second and more irregular phase (synchronization, signal demodulation...).

Sonar Beam Forming A classical sonar chain consists in a first and systematic step followed by a more general data processing. The first step provides frequency and location correlations (so called *beam*) from a continuous flow of data delivered by the hydrophones (microphones disposed around a submarine). It is based on signal elementary transformations: FFT (Fast Fourrier Transformation) and discrete integration. The second step analyses a given set of beams and their history to identify temporal correlation and association to signal sources.

JPEG-2000 Encoder/Decoder JPEG-2000 is a new standard format for image compression. The encoder works in a two-steps approach [40]. The first part (from preprocessing to wavelet decomposition) is systematic. The second part of the encoder includes irregular processing (quantification, two coding stages). The decoder works the other way around: a first irregular phase is followed by a systematic phase.

4.2. Transport

Our contribution to the safety in transport applications are twofold. In the context of the ModEasy Interreg project we have studied anti-collision radars for cars and in the context of the I-Trans competitiveness pole we collaborate with the INRETS on the model driven test of the ERTMS european railway signalization standard.

The automotive industry has specific problems, particularly due to increased safety requirements and legal framework. The automobile is a hostile environment: especially in the engine compartment. Some failure modes will be benign, whereas others may be dangerous and cause accidents and endanger human life. The Annex to the IEE Guidance Document on EMC and Functional Safety [ref] enumerates 21 electronic systems that may be present in the modern automobile. Some of these electronic systems have the potential to endanger the safety of vehicle occupants or other road users should an error or a mis-operation occur.

In the ModEasy Interreg project we want to model a cruise control connected to the satellite positioning system, GPS. From a UML specification and using classical verification and model checking techniques, we want to assure the correct behaviour of the system. Using model transformation allows the guarantee of these verifications at the lower levels like SystemC/VHDL.

Collision avoidance radars are now integrated into high end models by car manufacturers. The current devices are however based on the frequency modulation and their maximum range is limited if the emitted power is kept under the recommended values The receiver uses digital correlators which have been implemented via DSP microprocessors. The codes are generated using FPGA devices. In order to achieve greater integration and improve security, we are now seeking to design the major parts as embedded systems based on FPGA and SoC devices. In this context, the use of tools developed in the ModEasy project will improve and facilitate the design of such complex systems. Moreover, as ModEasy is based on metamodels and transformations between metamodels, new algorithms or new FPGAs can rapidly be integrated in the system by the re-use of existing functional blocks.

Our other application that has started in late 2006 is the study of a model based test methodology for the ERTMS railway signalization standard. We aim at developping a test methodology that can be used to qualify parts of such a system. We develop general enough metamodels so that they can also be used to test applications developped with Gaspard2. Our approach is based on timing behaviour specifcation with MARTE time annotations and then derivation of test cases based on the analysis of these specifications (by Petri net tools for example).

4.3. High-performance computing

Using the OpenMP/Fortran code generation chain, we have experimented the generated code in a typical operation in the scientific field: the matrix multiplication. We have compared generated code with optimized BLAS library function. Different algorithms have been generated: row-column multiplication, multiplication by block, multiplication by block using optimized BLAS function for the sequential part. Those algorithms have been compared with the sequential BLAS function and the parallel BLAS function. The results [79] show that the way to use Gaspard2 in the High Performance Computing field is to entrust Gaspard2 to manage parallelism and to use optimized function for the sequential part.

5. Software

5.1. MoMoTE

Keywords: *MDA*, *MDE*, *Model Transformation*, *QVT*, *Query View Transformation*.

Participants: Cédric Dumoulin [contact person], Anne Etien, Antoine Honoré, Emmanuel Renaux.

MoMoTE is a Java Framework that allows us to perform models to model transformations. It is composed of an API and an engine. It takes input models that conform to some metamodels and produces output models that conform to other metamodels. A transformation with MoMoTE is composed of rules. A rule describes a subset of the transformation and can call subrules.

The API is provided in an Eclipse plugin. It works with Ecore-based metamodels. Developers that use MoMoTE extend the rule class to make their own set of rules. Each rule must implement a selection on input models. This is done with the EMFT Query API (an Eclipse plugin to query models based on the OMG QVT standard). For each result of the selection, the rule must define elements to be created. Finally the rules are assembled in a hierarchical tree, passed to the transformation engine.

The MoMoTE transformation engine executes rules recursively, creating output element models first and solving references in a second step. Some of the rules are stamped as root, and are executed first. To perform the transformation, users have to provide input models. The engine will create the output models according to the rules.

MoMoTE is used in the Gaspard2 software and distributed with it. It contributes to the transformation chain of Gaspard2. However, it can be used outside Gaspard2 as a standalone transformation engine.

5.2. MoCodE

Keywords: *Code generation*, *MDA*, *MDE*, *language*, *model*.

Participants: Cédric Dumoulin [contact person], Anne Etien, Antoine Honoré.

MoCodE (Models to Code Engine) is an API that enables to perform model to text transformation. It takes as input a set of models conform to several metamodels and a set of Java classes. Then, it produces text files.

MoCodE recursively gets each element of input models and executes a corresponding Java code. The Java code is used to produces text files that can be code written in other languages. The Java code is generated itself from templates, written in JET (a part of the EMF plugin). Developers are in charge of the produced text (i.e. JET templates) and the division of the output files structure (the MoCodE engine configuration). At this time, DaRT uses it to produce OpenMP/FORTRAN, SystemC, LUSTRE and VHDL code from models conform to the Gaspard2 metamodel.

The MoCodE API is an Eclipse plugin based on the Eclipse Modeling Framework (EMF) plugin. This is a part of the Gaspard2 software and processes last operations of its transformation chain. However, it can be used as an autonomous plugin in Eclipse.

5.3. Gaspard2

Keywords: *Eclipse*, *IDE*, *SoC Design*, *Visual Design*.

Participant: Pierre Boulet [contact person].

Gaspard2 is an Integrated Development Environment (IDE) for SoC visual co-modeling. It allows or will allow modeling, simulation, testing and code generation of SoC applications and hardware architectures.

Its purpose is to provide one single environment for all the SoC development processes:

- High level modeling of applications and hardware architectures
- Application and hardware architecture association
- Application refactoring
- Deployment specification
- Model to model transformation (to automatically produce models for several target platforms)
- Code generation
- Simulation
- Reification of any stages of the development

The Gaspard2 tool is based on the Eclipse [54] IDE. A set of plugins provides the different functionalities. Gaspard2 is capable to chain transformations written with the MoMoTE plugin and, at the end, call a code generator written with the MoCodE plugin. Each transformations chain is described in a chain model conform to a chain metamodel. An important part of the core of Gaspard2 is an engine that executes models transformations chains.

Application, hardware architecture, association, deployment and technology models are specified and manipulated by the developer through UML diagrams, and saved by the UML tool in an XMI file format. Gaspard2 manipulates these models through repositories (Java interfaces and implementations) automatically generated thanks to the Ecore specification.

6. New Results

6.1. Specification models

Participants: Rabie Ben Atitallah, Pierre Boulet, Jean-Luc Dekeyser, Cédric Dumoulin, Abdoulaye Gamatié, Frédéric Guyomarc'h, Antoine Honoré, Sébastien Le Beux, Philippe Marquet, César Moura, Éric Piel, Safouan Taha, Julien Taillard, Huafeng Yu.

6.1.1. A MARTE compliant Gaspard2 profile

As part of the recently voted OMG standard MARTE, we have developped the Hardware Resource Model (HRM), which is an open framework for UML-based hardware modeling. HRM adapts the wide range of UML constructs to hardware and provides different modeling views corresponding to logical and physical abstractions [32]. It also covers many detail levels. MARTE handles the heterogeneity of embedded systems by adopting the Y-model that is equivalent to the Gaspard2 one including three models, the application model of the system functionalities, the resource model of the execution platform, which is in turn, composed of three models (generic + software + hardware) and finally the allocation model that maps the application to resources. An appropriate methodology was also developped to apply HRM during the hardware design process [31]. Such a methodology is recommended to get an efficient use of the HRM and benefit from its features. This methodology is incremental and bottom-up. It starts from elementary components and with successive compositions it reaches the whole platform model. The final model is then instantiated to get the real world platform. For cost, time and flexibility reasons, it is a common practice to simulate virtually the modeled hardware and test its adequacy to provide a sufficient execution platform for the embedded application. A code generation from HRM models was developped to interface simulation tools like Simics [80] we have selected for this study. To be conform with this new standard, the Gaspard2 profile was adapted as an extension of the MARTE profile. Now, it uses HRM for hardware modeling, RSM (Repetitive Structure Modeling) for repetitive structures and GCM (Generic Component Modeling) as a base for component modeling. Other concepts, which are part of the Gaspard2 profile but not in MARTE, are kept as is. In the future, some of the Gaspard2 concepts would be suggested for an integration in MARTE.

6.1.2. Deployment

The notion of *Deployment Specification* has been previously introduced in order to generate compilable code from the SoC model. For that, it is necessary for the designer to specify which IP (or function) must be used to implement an elementary component. The Deployment Specification package proposes concepts which (i) allows to describe the relation between a Gaspard2 representation of an Elementary component (a box with ports) to a text-based code (a function with arguments) and (ii) allows to inform the Gaspard compiler and platform of specific behaviours of each component (such as average execution time, power consumption...) in order to generate a high abstraction level simulation in adequacy with the real system.

The Gaspard2 metamodel now offers mechanisms to separately represent IPs from the rest of the model, in a library, and later deploy application and hardware components on them. To demonstrate the usage of those mechanisms and help the first users of Gaspard2, we have built such a library, the GaspardLib. It currently contains a dozen of hardware IPs (such as a MIPS core, a memory, a bus...) and approximately twenty software IPs (such as a DCT, an FFT, a sum of vectors...).

When it was possible, the descriptions of he IPs relay on the notion of abstract implementation of the Gaspard2 metamodel which allows to specify several versions of an IP having the same functionalities but implemented at various levels of abstraction or in different programming languages [19]. This in turn permits to use the IPs defined in GaspardLib independently from the transformation target. For instance one can associate a processor component to a MIPS core, and generating, without modifications in the SoC model, simulations at the TLM PVT level and at the TLM CABA level.

6.1.3. OCL constraints in Gaspard

The Gaspard2 design framework has been extended with an OCL-based verification functionality in order to early check design errors during model definition. This enables to reduce some errors so that transformations can be applied to safe models towards lower abstraction levels.

More precisely, we added OCL constraints to the Gaspard2 profile and developed a plugin called "*Gaspard-Validation*" that allows the user to check the syntactic correctness of described models with respect to their construction rules as stated in Gaspard2. This verification currently applies to the concepts relying on the `Application` metamodel. Future works aim at extending the applicability to all concepts of Gaspard2.

6.1.4. Template

A study has been started to assess the need of defining parameterizable elements in the Gaspard's modeling process, so that Gaspard's elements may be only partially defined in a first moment, leaving some "blanks" to be defined in a later time, but still in the modeling phase (in other words, before instantiating these elements). This would enable the creation of component libraries, increasing reuse.

Discussion focus on the prospective applications of parameterizable elements; on how the "parameterization" concept could be explored using existing tools, namely UML tools (e.g. MagicDraw, Papyrus,etc.); and finally on the benefits and disadvantages of adding it to the Gaspard's modeling process.

A few approaches have been envisaged aimed at putting into practice parameterizable elements. This has been discussed under two different levels:

1. UML: since UML tools are the front-end interface used in the Gaspard2 modeling process, it is important to provide users with ways of expressing parameterizable models.
2. Gaspard2 metamodel: once a parameterizable model has been created using UML tools, the next step is to express the same model in the Gaspard2 metamodel. Consequently, Gaspard2 itself needs to offer support to parameterizable models.

In either case, we scrutinized some points concerning both notation and implementation of parameterizable elements.

6.1.5. Repetitive allocation

The allocation of application functionalities to hardware or software resources is an important point in the SoC design. Such allocation is both is spatial distribution and a temporal schedule of computations to computing resources, of data to memories, of data dependencies to communication resources.

We have introduced a distribution notation of repetitive application components onto repetitive hardware resources that also uses a compact notation to allow specifying High Performance Fortran like distributions such as *block*, *cyclic* or *block-cyclic* distributions. We have included this notation in the MARTE profile [22].

6.1.6. Control features

The Gaspard2 framework is particularly adapted for the design of embedded systems that compute large amounts of data in a regular fashion. However, it does not allow to deal with aspects such as temporal constraints imposed by system environment, or control in computations according to different modes or configurations. So, we proposed an extension of that takes into account control features by considering mode automata and synchronous equations.

The resulting automata-based control holds a semantics that combines sequential automaton transitions and parallel repetitions, through the inter-repetition dependency defined in Gaspard2. Compared to the previous status of this work [15] [64], a complete syntax and formal semantics for this combination are under definition. While such an extension of Gaspard2 with control features increases its expressivity, more complex behaviors will become possible to describe. Validation techniques such as model-checking or discrete controller synthesis can therefore be very useful to check system properties.

An extension of the transformation chain developed in Gaspard2 towards synchronous code in Gaspard2 with control features is to be implemented soon. It could rely on an existing simplified version of a metamodel considered for the integration of control features in VHDL code.

This work is done in collaboration with Eric Rutten from the Pop-Art team-project at INRIA Rhône-Alpes in Grenoble.

6.1.7. A semantics for the Array-OL language

Starting an effort to provide more formal foundations to our work, we have defined a denotational semantics for the Array-OL model of computation. We have also defined a subset of Array-OL, named static Array-OL, that exhibits the good property of allowing a fully static verification of the determinism of the models.

6.2. Model-driven engineering for simulation, compilation, verification, evaluation and synthesis

Participants: Yassine Aydi, Rabie Ben Atitallah, Abou El Hassan Benyamina, Pierre Boulet, Jean-Luc Dekeyser, Cédric Dumoulin, Anne Etien, Abdoulaye Gamatié, Calin Glitia, Frédéric Guyomarc'h, Antoine Honoré, Sébastien Le Beux, Philippe Marquet, Samy Meftali, Smaïl Niar, Éric Piel, Imran Rafiq Quadri, Julien Taillard, Huafeng Yu.

6.2.1. A RTL metamodel

We propose to describe the hardware accelerators dedicated to intensive signal processing at RTL level (Register Transfer Level) i.e. independently from any HDL language (VHDL or Verilog for instance). For this purpose, we developed an RTL metamodel that enables description of hardware accelerator. This metamodel relies on a factorized expression of the parallelism included in hardware accelerators.

The RTL metamodel also enables the description of FPGA according to different views [27]. One view is dedicated to the description of resources contained in an FPGA (storage, computing, etc.); another one focuses on FPGA topology (cell organisation) and the last one defines the FPGA configuration zones.

Some concepts of the RTL metamodel are dedicated to the mapping of a hardware accelerator onto FPGA. These provide implementation characteristics of a hardware accelerator for a given FPGA. Such information will allow a fine topological placement of the hardware accelerator onto the FPGA.

We are currently working on the automatic design space exploration for hardware accelerator. Based on FPGA implementation characteristics of a hardware accelerator, a Gaspard2 application is modified in order to generate (thanks to model to model transformation) another hardware accelerator. The aim of the design space exploration is to enhance the fit between a hardware accelerator and a specific FPGA.

6.2.2. A synchronous equational metamodel

A synchronous metamodel based on the results of last year on the synchronous modeling of Gaspard2 concepts has been proposed [33].

The synchronous data-flow languages, such as Lustre, Signal and Lucid Synchrone, share several common features, which enable their code generation with the help of a unique synchronous metamodel. This metamodel aims not only at enabling models in the three synchronous data-flow languages, but also at bridging the gap between Gaspard and the synchronous technology, which offers a wide family of tools that allows designers to formally validate their models.

6.2.3. Transformation chains for code generation

As illustrated in figure 1, the design approach adopted in the Gaspard2 framework considers model transformations for code generation towards different languages. These languages are intended for various purposes. All implemented transformation chains use MoMoTE and MoCodE. The next paragraphs present each transformation chain.

6.2.3.1. Towards synchronous dataflow languages

The synchronous transformation chain enables the code generation of synchronous dataflow languages, particularly the LUSTRE and SIGNAL language [33], [37]. These transformations start from high-level specifications of *application*. The implemented chain consists of about five thousands lines of code. Another code generator for the synchronous language LUCID SYNCHRONE is still under development.

6.2.3.2. Towards VHDL

A model to model transformation allows to transform a Gaspard application into the RTL metamodel [28]. A specific focus on data dependencies has been set in order to compile both space and time dependencies [26]. Then, the hardware accelerator model conform to the RTL metamodel is transformed into VHDL code via a code generation. The generated code can be easily simulated and synthesised onto FPGA thanks to classical commercial tools (Quartus from Altera for instance). Several experimentations validated this approach. A constraint file is also generated in order to place hardware accelerator design onto FPGA.

6.2.3.3. From deployed models to models conformed to the Loop metamodel

In order to produce the simplified model of a SoC expressed using the Loop metamodel from a deployed model, two successive model transformations have been defined. Following the MDE recommendations, an intermediary metamodel is used between those transformations: the Polyhedron metamodel. Each transformation is a set of transformation rules, each of them working on a very small set of elements. These implementations use MoMoTE.

The first transformation, from a deployed model to the Polyhedron metamodel is composed of 59 rules. They mainly allow us: to express the repetitions as polyhedrons, to separate the application tree following the association specification, to map the data arrays on the memories, and to simplify the deployment specifications. The second transformation, from the Polyhedron metamodel to the Loop metamodel, converts the mapping expressed by polyhedrons into loops expression. Each polyhedron is transformed into loops.

The two following transformations chains start from the Loop metamodel, as an intermediary one between the very high concepts represented in the deployed model and the concepts manipulated in the generated code.

6.2.3.4. Towards OpenMP

OpenMP/Fortran code is generated starting from the Loop metamodel. The generation is made in two steps. The first step generates a model in the OpenMP metamodel from the Loop metamodel. It consists in scheduling the tasks in order to obtain a valid program, determining which variables have to be declared and generating synchronisation barrier when needed. Then from the OpenMP metamodel, OpenMP/Fortran code is generated. A code generator for OpenMP/C is still under development.

6.2.3.5. Towards SystemC/PA

Starting from the Loop metamodel, a transformation generating SystemC code was implemented [20]. It is based on the usage of templates with the MoCodE tool and it generates both the simulation of the hardware components and the application components. Each hardware component is transformed into a SystemC module with its ports linked. For each processor, the part of the application which has to be executed on this processor is generated as a set of *activities* dynamically scheduled and synchronised, following the model of execution defined for the Gaspard2 applications on MPSoCs. Additionally, the framework needed to automatically compile all the simulation code is also generated (as a Makefile). The level of abstraction is the TLM-PA level, allowing the user to see the execution of the program in term of patterns usages, the main data element of a Gaspard2 program, instead of reading and writing bytes. This also permits to speed up the simulation.

6.2.4. Performance evaluation

6.2.4.1. Performance evaluation of MPSoC in SystemC

In our previous work, we have developed an MPSoC platform described at the Timed Programmer View (PVT) level[18]. This platform includes various kinds of component models that have been designed: processors, caches, interconnection network, RAM, DMA controller and a DCT hardware accelerator. At this level, a timing model is defined and plugged in the architectural simulator to approximate the execution time.

Nowadays, early power estimation is increasingly important in MPSoC architectures for a reliable Design Space Exploration (DSE). During this year, we have developed an MPSoC power modeling framework at the PVT level that "allows finding optimal architectural alternatives early in the design flow. These alternatives exhibit a good performance/power trade-off". Using a hybrid power modeling methodology, we developed several power models derived from both physical measurements and analytical expressions. Plugging these power models into the PVT architectural simulator makes it easy to estimate the application's performance and power consumption with high simulation speedup[16]. Experimental results show that our MPSoC environment gives a high simulation speedup factor of up to 18 with a negligible performance and power estimation error margin.

6.2.4.2. New NoC for Intensive Signal Processing applications needs

Intercommunication requirements of future massively parallel SoCs are not satisfied with a single shared bus or with a hierarchical bus due to their poor scalability with the number of processor and their shared bandwidth between all the components.

To overcome this problem, Network on Chip (NoC) has been proposed by academia and industry as an on-chip communication challenge solution for the next generation of multiprocessor system on chip, denoted MPSoC architectures.

Historically, Multistage Interconnection Networks (MIN) are used in multi processors systems. We propose to use MIN as a Network on Chip to connect processors to memory modules on MPSoC.

Many variations of MINs have been introduced. A MIN is defined by, its topology, switching strategy, routing algorithm, scheduling mechanism and fault tolerance. A reconfigurable MIN can take advantage of the regular communication patterns to minimize the contention and to improve the global bandwidth of the system. Simulation in SystemC at different levels of abstraction is a first exploration before synthesis on FPGA.

In this context, we are also developing new coherency maintenance protocols especially dedicated to MPSoC architectures equipped with high performance NoC. In this study, we aim to demonstrate that coherency protocols dedicated to shared memory MPSoC equipped with NoC (such as MIN or crossbar) are more efficient than existing protocols. The under design protocols take into-account the behaviour of the application and automatically adapts the manner by which read/write operations are performed.

6.2.5. Implementing partial dynamic reconfiguration in FPGAs

The domain of Reconfigurable Architectures is very vast and has lead to a spectrum of Reconfigurable Architectures (RAs). These RAs can be classified in general as either Coarse Grain RAs or Fine Grain RAs. The Coarse Grain RAs are basically suited for specific, customized data path applications with advantages of having increased performance and less communication delays. The disadvantage of these RAs is their lack of flexibility to adapt to general applications. Fine Grain RAs (mainly SRAM Based commercial FPGAs) work at the bit level manipulation level and offer greater flexibility in terms of adaptability to applications with a cost of increased reconfiguration time, performance and communication delays.

We basically focus on these Fine Grain RAs, i.e. FPGAs particularly Xilinx's Virtex Series FPGAs which have the capability to carry out Partial Dynamic Run time Reconfiguration (PDR). The objective of the thesis is to extend the work already carried in our Gaspard2 chain and to introduce the notion of dynamic reconfiguration in Gaspard2. For this purpose, we plan to introduce certain aspects of the UML Profile for MARTE (Modeling and Analysis of Real-Time and Embedded Systems) in the Gaspard2 Environment.

We have explored the architecture necessary to carry out the PDR [30]. Addition of Physical properties (such as power consumption, area layout, reconfiguration time) are some of the notions to be included related to PDR in our work. A notion of QoS is also to be implemented associated with a controller responsible to carry out the reconfiguration. This mainly helps to change the context of the application depending on the user needs.

Although, the notion of Control (for the synthesis of VHDL) is present in the existing environment i.e. the static control implemented in the FPGA can choose between different configurations, the only problem is that it is static in nature. For our need, we need to introduce the notion of dynamic control in the Gaspard2 environment which will be integrated with the notion of QoS.

6.2.6. Compilation technique for data-parallelism

The Array-OL transformation toolbox now implemented and integrated in Gaspard2 was tested on real applications. As a result of these tests, we got some important feed-back on the transformations. The major problem is introduced by the presence of more than one intermediary array between two tasks intended for fusion. The solution previously proposed and implemented implied the multiplication of the first task, which introduces a lot of re-calculations. A new partial solution was proposed and developed for the case where the intermediary arrays are represented by the same array but consumed multiple times.

Another direction of research is represented by the introduction of inter-repetition dependencies and their impact on the Array-OL transformations. As a first result was the need to extend the concept of inter-repetition dependence by allowing multiple default-links characterized by tilers (the tilers construction guaranties that just one default-link can be chosen for a repetition instant). An additional stage will be added to the Array-OL transformations, the one that will transform the inter-repetition dependencies according to the repartition of repetitions before and after the transformation.

Due to the complex way we can express pattern consumption/producing in Array-OL with the help of tilers, the task of writing those tilers is not trivial, especially for users not familiar with the semantics of Array-OL. To ease this task we have implemented a tiler editor that can be connected anywhere we need to and provides help by identifying some probable choices for a tiler (separate dimensions for repetition and pattern, patterns parallel with the axes, sliding windows). The next step is to extend this editor with a visual interface more suited for more complicated constructions.

The work about the GILR optimization heuristics has continued with the visit of Abou El Hassan from the university of Oran, Algeria. We have started the investigation of a hierarchical genetic algorithm based heuristic. Preliminary results have been published [21].

6.3. Methods and technology

Participants: Adolf Abdallah, Pierre Boulet, Jean-Luc Dekeyser, Cédric Dumoulin, Anne Etien, Abdoulaye Gamatié, Souha Kamoun, Emmanuel Renaux, Huafeng Yu.

6.3.1. Transformation techniques

The Gaspard2 environment consists of several conception flows composed of models corresponding to various system viewpoints and defined at different levels of abstraction. Transformations are used to go from one model to an other, i.e. from one level of abstraction to another. Developing and maintaining such transformations is a complex task requiring, like other engineering process, tools to document and to ease the development. Actually, transformation world lakes of such tools, and also of a "standard" to document and represent the intent of a transformation.

In the same way that UML help to develop and to document software by representing them graphically, we propose a graphical representation of transformations. We propose to focus on the “what” (what should be transformed) rather than the “how” (how implement the transformation). In other terms, we suggest to extract the intention of the transformation from the implementation details. Since years, modeling activities have been mainly supported by UML-like modeler tools and then, visual notation has become more familiar.

Our proposition relies on a graphical representation of transformation model that seems to be the best abstraction to focus on the purpose of the transformation independently from the implementation details.

We have developed the Transformation Modeling Language TrML. In this modeling language, we gather the core features that we consider mandatory to represent transformations. We provide the corresponding UML profile that makes it portable on any UML tool. TrML is not associated to a transformation engine but is defined in a metamodel, making possible to transform a TrML transformation to a particular engine representation.

Transformations aim to be automatically executed. We thus provide a transformation engine for TrML transformation by building TrML on top of an existing transformation engine, ATL. For this purpose, we adopt the bootstrap mechanism allowing to write a transformation from TrML to the model associated to the ATL engine.

We are currently thinking on building TrML on top of MoMoTE. Two ways can be explored either a code generation to Java or an abstraction of the MoMoTE concepts in a metamodel.

6.3.2. Morphing

Writing transformation rules is a complex task due to navigation constraints. Relaxing these constraints for the transformation time eases the rules conception. It may also be necessary to add features into domain metamodels in order to ease development of tools build around this metamodel, or to ease the transformation rule development. However, these features additions as well as the relaxation of the navigation constraints must be contained in a well defined context in order to authorize exchange of models between tools manipulating several variants of the same domain metamodel. This implies two objectives: modifying a domain metamodel in order to ease the manipulation of the corresponding models and allowing the exchange of models between tools using or constructed upon these adapted metamodels and thus without model transformation.

The exchange of models remains possible as long as the metamodels adaptation is constrained in order not to change the nature of this model but to ease its manipulation. In this context, we have introduced the concept of morph that corresponds to a variant of the original domain metamodel and the concept of gene that specifies these variations. These two notions allow variation without altering the way instances are defined and saved. Each morph provides functionalities that are adapted to the use of this morph. A model, compliant to one morph can be imported by any tools based on morphs of the original domain metamodel. Thus, tools based on morph of a same domain metamodel become interoperable.

6.3.3. Design verification

6.3.3.1. Synchronous approach

The objective of the synchronous modeling of data-intensive computations is to propose, on the one hand, a link between analysis and compilation techniques of data-parallel applications, and on the other hand, analysis and verification techniques of synchronous data-flow and automata-based languages.

With the code generated automatically via the transformation rules implemented in Gaspard2, first examples of validation have been carried out, which include safe array assignment check, deadlock analysis, and functional simulation. The first one helps to find some array assignment problems in the original design, for instance array multiple writing, array partial initialization. The second one involves a compiler-level deadlock analysis, which is not always obvious to detect in the original graphical design. The last one allows the observation the system behavior at the execution period. These validation methods provide Gaspard2 users with a first software verification phase before implementation.

This work is done in collaboration with Eric Rutten from the Pop-Art team-project at INRIA Rhône-Alpes in Grenoble.

6.3.3.2. Test-based approach

The European Railway Traffic Management System (ERTMS) is a complex railway signaling system featuring hundreds of functional requirements, many of them being safety-critical. To ensure an efficient functional verification of the system, a powerful test specification methodology is needed. We proposed then, a model-based methodology for the test of the ERTMS system.

This methodology consists in capturing the textual requirements by a more formal model and in the automatic generation of test scenarios. As modeling languages, we have used SysML extended with MARTE annotations. Several SysML diagrams were used and adapted to achieve the goal of the test process. The requirement diagram captures the textual specifications and we have chosen the activity diagram as the test model. Test scenarios are specified by sequence diagrams which result from the transformation of activity diagrams [24] [25]. The automation of the test generation process has several benefits especially for the testers in term of time saving and cost decreasing.

Our future work aims at improving the technique and its tool support. We plan to improve the transformation rules to generate several test scenarios. We intend to further investigate the execution of test cases, especially finding a format for the scenarios supported by the ERTMS railway simulator. Indeed the cover of the scenarios can be measured only after their execution, it is thus necessary to be able to simulate the scenarios.

6.4. Experimental validation

Participants: Rabie Ben Atitallah, Pierre Boulet, Jean-Luc Dekeyser, Frédéric Guyomarc'h, Sébastien Le Beux, Philippe Marquet, Samy Meftali, Smaïl Niar, Julien Taillard.

6.4.1. Anti-collision detection radar

In the scope of the European ModEasy project, we had the opportunity to participate to the implementation of a radar anti-collision system for vehicles. In collaboration with the INRETS[3] from Lille (Institut National de Recherche sur les Transports et leur Sécurité) and the IEMN[4] from Valenciennes (Institut d'Electronique de Microélectronique et de Nanotechnologie), we validated the behavior of the correlation algorithm used for obstacle detection during on road real condition tests [23].

6.4.2. MPSoC simulation in SystemC

MultiProcessor Systems-on-Chip (MPSoC) architecture has become a solution for designing embedded systems dedicated to applications that require intensive computations. The most important design challenges in such systems consist in reducing simulation time and estimating performance appropriately. In our work, we focus on the use of Transaction Level Modeling (TLM) in an MPSoC design which corresponds to a set of abstraction levels that simplifies the description of inter-module communication transactions using objects and channels between the communicating modules. Consequently, modeling MPSoC architectures becomes easier and faster than at the CABA (Cycle Accurate Bit Accurate) level.

As our objective is to propose reliable environment for rapid MPSoC design space exploration, the framework has been designed in the context of PVT (Timed Programmer View) level. In the conventional definition of the PVT level, the hardware architecture is specified for both processing and communication parts, as well as some arbitration of the communication infrastructure is applied. In addition for performance estimation, this level is annotated with timing specification.

In this context, we have proposed a new PVT approach, containing three sublevels namely PVT-PA (Pattern Accurate), PVT-TA (Transaction Accurate) and PVT-EA (Event Accurate). These sub-levels are intended to increase the speed of MPSoC simulation. Several component models have been developed using these three sublevels and enhanced with performance and power consumptions estimation tools to provide accurate execution time and power consumption estimates. Simulation results demonstrate the complementarity between the proposed sublevels which provide different simulation speedup/accuracy trade-offs [18].

6.4.3. High-performance computing

The solver part of CARMEL has been modelled with Gaspard2. This work was done in two parts: first, the design of the high-level model, and then the choice of the IP for the building blocks of the solver. Different algorithms and different libraries of IP have been tested to exhibit the most efficient ones. These results have also been validated with the ones produced by the original FORTRAN code of CARMEL.

[3] http://www.inrets.fr/infos/centres/ct_vascq.html
[4] http://www.iemn.univ-lille1.fr/

6.4.4. Massively parallel processing SoC

We are working on the definition and design of mppSoC, a massively parallel system on a chip. The objective of the mppSoC project is to reconsider the interest of massively parallel machines with nowadays design methodologies based on IP reuse and nowadays integration technologies.

MppSoC is a SIMD architecture composed of a grid of processing elements (the PEs) and memories connected by a regular neighbourhood network and a general purpose global network.

Some improvements of the system architecture are possible because of the high degree of integration: The mppSoC PEs share most of their design with the control processor, the integrated network allows to exchange data between PEs, but also between the control processor and the PE memories, and even to connect the external devices to the system.

The first works in the project have already led to the design of a cycle-accurate bit-accurate SystemC simulator of the architecture, and to an implementation prototype on FPGA. A complete tool chain based on an data-parallel language allows to generate binary programs that execute both on the simulator and the hardware implementation. We have been able to integrate 16 processing elements in a single FPGA [29].

We are now considering the redesign of the mppSoC as a set of IPs. Then, from theses IPs, a generic design of a parametrized mppSoC system will be possible. The parametrization of a mppSoC system will defined the number of PEs, the amount of memory, the kind of network required for a specific system that will be design for a given application or class of applications. The whole mppSoC design chain is impacted by this new characteristics.

7. Contracts and Grants with Industry

7.1. The CORTESS Project: A Carroll Project

Partners: CEA, Thales, INRIA (AOSTE, DaRT, ESPRESSO). This project is the follow up of the PROTES Carroll project. Its goals were to finish the standardization of the MARTE profile that was started in the PROTES project. The partners of CORTESS along with the other members of the Promarte (http://www.promarte.org/) submission team have successfully submitted to the OMG a proposal complying to all the requirements of the MARTE request for proposal. This proposal has been standardized in July 2007 and is currently in finalization (http://www.omgmarte.org/).

As part of the CORTESS project, we have also upgraded the Gaspard2 profile to comply with MARTE by reusing as most as possible the modeling constructs provided by MARTE. With the delivery of this profile the CORTESS project is now closed.

7.2. The PEAMS Project: A Carroll Project

Partners: CEA, Thales, INRIA (ALCHEMY, AOSTE, DaRT).

The objective of this project is to evaluate different architecture options in order to set up the Ter@ops project. The associated programming models will be also defined. The needs of the development framework will be defined and different tool options will be proposed. PEAMS will concentrate on data streaming dominated applications.

In this context, the DaRT project proposes his knowledge in the modeling of parallel architecture performance. It will participate to the extension of the SPEAR metamodel in order to enable the modeling of the Ter@ops architecture.

7.3. The OpenEmbedd Project: A RNTL Project

Partners: Airbus, Anyware Technologies, CEA, CS SI, France Telecom, INRIA (AOSTE, DaRT, ESPRESSO), LAAS (CNRS), Thales Aerospace, Thales R&D, Verimag.

The OpenEmbedd project[5] aims is to develop an engineering model driven open-source platform for real time and embedded systems. It deals with (1) UML standard for Real Time and Embedded systems, (2) innovating technology for interoperability, (3) mastering methodology chain, (4) real time models for simulation. The tools are evaluating in practical domains, e.g. the aeronautic sector, automobile sector, and telecom sector. The project will succeed in providing a technological core for Model Driven Engineering, by producing a set of tools dealing with different concerns about real time and embedded systems, and by validating an approach in the representative domains, in both applicative and methodological concerns. Software developed will be open-source. Future platform aims to federate academic partners effort and will guarantee a wide diffusion of the software.

Our OpenEmbedd partners are Airbus, Anyware Technologies, CEA, CS SI, France Telecom, INRIA, LAAS (CNRS), Thales Aerospace, Thales R&D, Verimag. The project has link with three competitiveness poles: Minalogic, System@tic, Aerospace Valley.

The activity of the DaRT Project in the OpenEmbedd RNTL project is to normalize models about real time and embedded systems domains, and more precisely the MARTE profile. The objectives are to participate to elaboration of a graphical editor generated from OpenEmbedd tools, and to work on plugins dedicated to simulation and checking. During the OpenEmbedd meetings that have been held in April and November in Grenoble and Toulouse respectively, DaRT presented the compilation chain from UML towards VHDL and the collapse of the hierarchical and repetitive Gaspard2 architecture model.

7.4. The Ter@ops Project: A System@tic Project

Partners: THALES (TRT, TOSA), Thomson, EADS (EADS Astrium, MBDA), Dassault Aviation, Renault, Valeo, Freescale (SAS), M2000, ARTERIS, Esterel Technologies, VirtualLogix, CEA-LIST, INRIA (Alchemy, Caps, DaRT), IEF, ENSTA, PRiSM, CRI (ARMINES / École des Mines), Laboratoire ETIS, RATP.

The Ter@ops project of the System@tic competitiveness pole aims at developing a hardware platform and the associated development framework for computation intensive applications. This project has started in December 2006.

We work at the level of the framework definition where we study the integration of Gaspard2 in a complete compilation and optimization framework for the Ter@ops platform and the compilation of the control proposed during the thesis of Ouassila Labbani in Gaspard2.

7.5. Collaboration within the competitiveness pole I-Trans

I-Trans is the official industrial cluster, which aims at bringing together major French actors in rail technology and innovative transport systems. The DaRT project strongly participates to this initiative through the collaborations with both concerned academic and industrial actors.

In this direction, we already have many discussions with partners (Inrets-Estas, Lagis, Lamih, Utc/Heudiasyc, Alstom and Certifier). These discussions lead to a few project proposals (Geneve, Klifr) that are under submission for ANR (Agence Nationale de la Recherche) funding. The goal of these proposals concerns on the one hand, the decrease of validation and certification costs in the implementation of the new European railway system ERTMS/ETCS and on the other hand, the ease of interoperability through the mutual recognition of ERTMS components between European member countries. This qualification requires costly and long tests. More specifically, in the context of innovate equipments for railways, the objective consists in developing methods, models and tools dedicated to the generation of scenario tests for the validation of ERTMS components.

The contribution of the DaRT project to the definition of solutions to the above issues is twofold: first, its capabilities in the design of architectures and the association between application and architecture, so as to explore impacts on the test of the deployment on embedded architectures; second, its design experience in the automotive domain (see section 8.1).

[5] http://www.openembedd.org

These aspects are currently studied by a starting PhD student in the DaRT project, conjointly with test problematics. This thesis is co-supervised with the Lagis laboratory.

7.6. Collaboration with CEA List

Partners: CEA List, DaRT

Since last year we have started a point to point collaboration with CEA around a UML profile for co-design. This work is done together with a PhD student (CEA funding). The main contribution consists in defining a metamodel for hardware architecture for the future MARTE standard. The results of this research is also integrated in the Gaspard2 tools at INRIA and in the AccordUML environment at CEA.

This collaboration is complementary with the above partnership between CEA and DaRT in the Cortess project.

7.7. Collaboration with Thales

Partners: ENSIETA, INRIA (DaRT), Thales

In order to increase productivity and thus decrease time to market, we propose to apply Model Driven Engineering (MDE) through the use of process components, which encapsulate the main activities of co-design processes. We consider that activities going from requirements analysis to implementation, whatever the chosen life-cycle, can be capitalized through process components. They are several formalisms to describe a process (BPML, CPR, PSL, SPEM, etc.). We propose a modified version of OMG's SPEM profile (Software Process Engineering Modelling) that allows modelling executable processes in order to implement a full MDE process.

We experiment our approach in a co-design process based on the use of the new MARTE profile and we intend to provide a tool that implements it in order to help engineers. In [62], we explain our approach applied to the development of Radio Frequency Transceiver. During our experimentation, we had to face some problems of metamodel formalization using tools. We have then proposed in [63] a framework to define more formalized metamodels.

The next challenging issue concerns the identification of a way our co-design process can be adapted to the Gaspard2 framework for design exploration.

This work is done in the context of a CIFRE PhD contract co-supervised by Joël Champeau from ENSIETA (Brest) and Jean-Luc Dekeyser.

7.8. Collaboration with Valeo - CNRT Futurelec

Partners: Valeo, INRIA (DaRT), L2EP

The objective of this project consists of fitting Gaspard2 for high performance computing and meta-computing. This work is done with a PhD student (Valeo/region funding). The results of this research are to be considered for integration in the Gaspard tool-set at INRIA and will be tested on finite element code for electric alternator simulator developed by Valeo and L2EP.

8. Other Grants and Activities

8.1. ModEasy Interreg III A Franco-English Cooperation

The ModEasy project[6] develops software tools and techniques in order to facilitate the development of reliable microprocessor-based electronic (embedded) systems using advanced development and verification systems. The defined tools will be evaluated in practical domains such as automotive.

[6] http://www.lifl.fr/modeasy

We have two partners in this project. First, the University of Kent, which works on formal system verification, embedded system development support and hardware integration from research councils and industry. Second, the IEMN (Institut Electronique Microelectronique Nanotechnologies), which has substantial expertise in the safety of land-based transportation systems especially for collision avoidance.

In this context, we have proposed a FPGA prototype that merges a reactive cruise control and an anti-collision radar in a single chip. The results of all partners have been presented during ModEasy Workshop[7] that held in Barcelona in September 2007.

8.2. Co-modeling and model engineering for HPC in electromagnetism software

Partners: IRISA, DaRT, L2EP.

The goal of this project is to use the power of new software engineering tools to design a new version of the electromagnetism solver CARMEL. This project emphasis on the fact that scientist from many different fields (physics, applied mathematics, high performance computing, software engineering) need to collaborate to ensure its success, and deliver a model of CARMEL using Gaspard2 metamodel. Then with the Gaspard2 development environment, we can generate a parallel software to solve the Maxwell equations.

8.3. STIC INRIA - Tunisia program

In the relation with the co-advising of a PhD student on the topic of "Modeling and high-level design of communication architectures for systems on chip : network on chip with dynamical reconfiguration", a collaboration with the team of Pr. Mohamed Abid at CES-ENIS is being built up. This collaboration is supported by the STIC Inria-Tunisia program, which aims at promoting the design of metamodels, transformation tools and techniques for the implementation of reconfigurable systems-on-chip. The resulting co-design environment will be validated on embedded systems dedicated to security in automobile, and more specifically in the design of cruise control systems integrating anti-collision radars.

Several successful student exchanges have been realized in 2006 between DaRT and CES-ENIS. Many other exchanges are already programmed for 2007.

8.4. International initiatives

8.4.1. ECSI

The European Electronic Chips & Systems design Initiative Missions are to identify, develop and promote efficient methods for electronic system design, with particular regards to the needs of the System-on-Chip and to provide ECSI members with a competitive advantage in this domain for the benefit of the European industry. The list of participants is on:
http://www.ecsi.org.

Our team became an ECSI member in 2004. In this context we organized the ECSI conference in Lille: FDL'04. Pierre Boulet is a member of the executive committee of ECSI and secretary of ECSI.

8.4.2. University of California - Irvine

We started collaboration with the Center of Embedded Computer Systems (CECS) of the University of California at Irvine around networks on-chip design. Our objective consists mainly in modelling high performance networks as multistage Delta networks as cycle accurate, SystemC, reusable IPs [67]. We plan to continue this in this topic our collaboration and move in the future to FPGA implementations of these networks. In fact, aspects of dynamic reconfiguration of FPGAs make them very adapted to 'intelligent' NoC implementation.

[7] http://www.lifl.fr/modeasy/workshop.html

8.4.3. University of Montreal

The collaboration with University of Montreal laboratory continued this year. The interested laboratory is the LASSO (Laboratoire d'Analyse et de Synthèse des Systèmes Ordinés[8]), from DIRO department (Département d'Informatique et de Recherche Opérationnelle[9]). Student Frédéric Bastien came in the team during two months, and worked on the VHDL description of a massively parallel machine on FPGA.

8.4.4. University of Oran

A collaboration has started with the university of Oran, Algeria. Abou El Hassan Benyamina has been invited for one year (mid 2006 to mid 2007) to initiate the collaboration. He is working with Pierre Boulet on scheduling and mapping algorithms for SoC.

8.5. National initiatives

We are members of the ASR[10] and SoC-SiP[11] GDRs (research groups from the CNRS).

9. Dissemination

9.1. Scientific Community

Pierre Boulet was in the steering committee and the program committee of FDL[12] since 2005. He is the UML topic program chair of FDL'07 and FDL'08. He is secretary and member of the executive committee of ECSI. He runs the UML for SoC design mailing list (umlsoc@univ-lille1.fr) with 136 subscribers from all over the world. He is a member of the INRIA evaluation committee since September 2005. He was in several PhD thesis committees in 2007. He will be vice-director of the LIFL starting January 1st, 2008.

Jean-Luc Dekeyser has been member of different program committees: ECMDA-FA 2007, recosoc07, Soc'07, GTTSE'07; invited speaker in FETCH'07 Grenoble, referee for some journals and conferences in MDE and SoC design. He is project leader of the ModEasy interreg. He was in about ten PhD and HdR thesis committees. As director of the Ph.D. program at LIFL, Inria and Doctoral School SPI, he was involved in the belgium/france relationship concerning PhD program. He was visiting ENIS Sfax tunisa in january and november 2007.

Cedric Dumoulin was in the program committee of IDM 07 (French workshop on Model Driven Engineering) held in Toulouse[13], and in the program committee of the special issue on MDE of the French review Revue Objet.

Philippe Marquet is in the program committee of the International Conference on Real-Time and Network systems (RTNS). He also served as referee for international journals and conferences in the field of real-time systems and hardware/software design.

The team organized the ModEasy workshop[14] that held in Barcelona in conjunction with the ECSI FDL conference. Rabie Ben Atitalah was program chair, Sébastien Le Beux was chairman.

9.2. Teaching

As the DaRT team is mostly composed of professors and associate professors, we have a very large teaching activity. The more directly related to the research themes of the team are the master-level courses "System-on-Chip design" (Pierre Boulet, Jean-Luc Dekeyser, Samy Meftali, Abdoulaye Gamatié) and "Introduction to real-time operating systems" (Philippe Marquet), "Simulation of Systems and Architectures" (Philippe Marquet and Samy Meftali) and "distributed systems" (Pierre Boulet).

[8] http://www.iro.umontreal.ca/~lablasso/lablasso/
[9] http://www.iro.umontreal.ca
[10] http://asr.cnrs.fr/
[11] http://www.lirmm.fr/soc_sip/
[12] http://www.ecsi.org/fdl
[13] http://www.planetmde.org/idm07
[14] http://www.lifl.fr/modeasy/workshop.html

The following internships were advised in the team:

- Gilles Atigossou, M1 Computer Science, Université des Sciences et Technologies de Lille,
- Louis Bigo, 3rd year Polytech'Lille
- Yasmina Boutahar, 3rd year Polytech'Lille
- Asma Charfi, M1 Computer Science, ENIS Université de Sfax (Tunisia)
- Lorenzo Chevalet, M1 Computer Science, Université des Sciences et Technologies de Lille
- Léo Davesne, 3rd year Polytech'Lille
- Flori Glitia, M1 Computer Science, Université des Sciences et Technologies de Lille
- Daniel Morosain, M1 Automatic Control and Computers, University POLITEHNICA of Bucharest (Romania)
- Bilel Neji, M1 Computer Science, ENIS Université de Sfax (Tunisia)
- Dorothée Perret, 3rd year Polytech'Lille
- Imen Tayari, M2 Computer Science, Université des Sciences et Technologies de Lille
- Gaetan Vansteelandt, M1 Computer Science, Université des Sciences et Technologies de Lille
- Mathias Vantieghem, 2nd year Génie Mathématique et Modélisation, Polytech'Clermont-Ferrand
- Nicolas Wojcik, M1 Computer Science, Université des Sciences et Technologies de Lille

10. Bibliography

Major publications by the team in recent years

[1] A. C. ALJUNDI, J.-L. DEKEYSER, M. T. KECHADI, I. D. SCHERSON. *A universal performance factor for multi-criteria evaluation of multistage interconnection networks.*, in "Future Generation Comp. Syst.", vol. 22, n^{o} 7, 2006, p. 794-804.

[2] R. BEN ATITALLAH, É. PIEL, S. NIAR, P. MARQUET, J.-L. DEKEYSER. *Multilevel MPSoC simulation using an MDE approach*, in "IEEE International SoC Conference (SoCC 2007), Hsinchu, Taiwan", September 2007.

[3] P. BOULET, C. DUMOULIN, A. HONORÉ. *Model Driven Engineering for System-on-Chip Design*, in "From MDD concepts to experiments and illustrations", S. GÉRARD, J.-P. BABEAU, J. CHAMPEAU (editors), ISBN:1-905209-59-0, ISTE, September 2006, http://iste.co.uk/index.php?f=a&ACTION=View&id=147.

[4] P. BOULET, P. MARQUET, É. PIEL, J. TAILLARD. *Repetitive Allocation Modeling with MARTE*, in "Forum on specification and design languages (FDL'07), Barcelona, Spain", Invited paper, September 2007.

[5] A. CUCCURU, J.-L. DEKEYSER, P. MARQUET, P. BOULET. *Towards UML 2 extensions for compact modeling of regular complex topologies*, in "MODELS/UML 2005, ACM/IEEE 8th international conference on model driven engineering languages and systems, Montego Bay, Jamaica", October 2005.

[6] A. GAMATIÉ, T. GAUTIER, P. LE GUERNIC, J.-P. TALPIN. *Polychronous Design of Embedded Real-Time Applications*, in "ACM Transactions on Software Engineering and Methodology (TOSEM)", vol. 16, n^{o} 2, April 2007.

[7] O. LABBANI, J.-L. DEKEYSER, É. RUTTEN. *Safe Design Methodology for an Intelligent Cruise Control System with GPS*, in "VTC'2006 Fall: 64th IEEE Vehicular Technology Conference, Montréal, Québec, Canada", (Also to appear as an IEEE ITSS Newsletter in December 2006), September 2006.

[8] S. LE BEUX, P. MARQUET, J.-L. DEKEYSER. *A Design Flow to Map Parallel Applications onto FPGAs*, in "17th IEEE International Conference on Field Programmable Logic and Applications, FPL, Amsterdam, Netherlands", August 2007.

[9] P. MARQUET, S. DUQUENNOY, S. LE BEUX, S. MEFTALI, J.-L. DEKEYSER. *Massively Parallel Processing on a Chip*, in "ACM Int'l Conf. on Computing Frontiers, Ischia, Italy", May 2007.

[10] S. MEFTALI, J. VENNIN, J.-L. DEKEYSER. *Automatic Generation of Geographically Distributed System Simulation Models for IP/SoC Design*, in "The 46th IEEE International Symposium on Circuits and Systems, Cairo, Egypt", December 2003.

[11] H. SBEYTI, S. NIAR, L. EECKHOUT. *Pattern-Driven Prefetching for Multimedia Applications on Embedded Processors*, in "Journal of System Architecture", vol. 52, n^{o} 4, April 2006, p. 199-212.

Year Publications

Doctoral dissertations and Habilitation theses

[12] S. LE BEUX. *Un flot de conception pour applications de traitement du signal systématique implémentées sur FPGA à base d'Ingénierie Dirigée par les Modèles*, Thèse de doctorat (PhD Thesis), Laboratoire d'informatique fondamentale de Lille, Université des sciences et technologies de Lille, France, December 2007.

[13] É. PIEL. *Ordonnancement de systèmes parallèles temps-réel, De la modélisation à la mise en œuvre par l'ingénierie dirigée par les modèles*, Thèse de doctorat (PhD Thesis), Laboratoire d'informatique fondamentale de Lille, Université des sciences et technologies de Lille, France, December 2007.

Articles in refereed journals and book chapters

[14] A. GAMATIÉ, T. GAUTIER, P. LE GUERNIC, J.-P. TALPIN. *Polychronous Design of Embedded Real-Time Applications*, in "ACM Transactions on Software Engineering and Methodology (TOSEM)", vol. 16, n^{o} 2, April 2007.

[15] O. LABBANI, JEAN-LUC. DEKEYSER, P. BOULET, É. RUTTEN. *UML2 profile for modelling controlled data parallel applications*, in "Advances in Design and Specification Languages for Embedded Systems", S. A. HUSS (editor), ISBN: 978-1-4020-6147-9, Springer Verlag, September 2007.

Publications in Conferences and Workshops

[16] R. ATITALLAH, S. NIAR, J.-L. DEKEYSER. *MPSoC Power Estimation Framework at Transaction Level Modeling*, in "19th International Conference on Microelectronics (ICM 2007), Cairo, Egypt", December 2007.

[17] Y. AYDI, S. MEFTALI, M. ABID, J.-L. DEKEYSER. *Dynamicity Analysis of Delta MINs for MPSoC Architectures*, in "Conférence internationale des Sciences et Techniques de l'Automatique (STA'07)", November 2007.

[18] R. BEN ATITALLAH, S. NIAR, S. MEFTALI, J.-L. DEKEYSER. *An MPSoC performance estimation framework using transaction level modeling*, in "IEEE RTCSA'2007, Daegu, Korea", 2007.

[19] R. BEN ATITALLAH, É. PIEL, S. NIAR, P. MARQUET, J.-L. DEKEYSER. *Multilevel MPSoC simulation using an MDE approach*, in "IEEE International SoC Conference (SoCC 2007), Hsinchu, Taiwan", September 2007.

[20] R. BEN ATITALLAH, É. PIEL, J. TAILLARD, S. NIAR, J.-L. DEKEYSER. *From High Level MPSoC description to SystemC Code Generation*, in "International ModEasy'07 Workshop in conjunction with Forum on specification and Design Languages (FDL'07), Barcelona, Spain", September 2007.

[21] A. E. H. BENYAMINA, P. BOULET. *Multi-objective Mapping for NoC Architectures*, in "1st International Conference on Digital Communications and Computer Applications, Jordan", March 2007, p. 132-139.

[22] P. BOULET, P. MARQUET, É. PIEL, J. TAILLARD. *Repetitive Allocation Modeling with MARTE*, in "Forum on specification and design languages (FDL'07), Barcelona, Spain", Invited paper, September 2007.

[23] J.-L. DEKEYSER, S. LE BEUX, P. MARQUET. *Une approche modèle pour la conception conjointe de systèmes embarqués hautes performances dédiés au transport*, in "Workshop International : Logistique & Transport (LT'2007)", November 2007.

[24] S. KAMOUN, P. BOULET. *Model-Based Testing of the ERTMS System with SysML and MARTE*, in "MoDeVVa'07 (integrating V&V in MDE), Nashville, USA", October 2007.

[25] S. KAMOUN, P. BOULET. *Une approche modèle pour la génération de scénarios de tests : Application au système ERTMS/ETCS*, in "Workshop International : Logistique and Transport 2007, Sousse, Tunisie", November 2007.

[26] S. LE BEUX, P. MARQUET, J.-L. DEKEYSER. *A Design Flow to Map Parallel Applications onto FPGAs*, in "17th IEEE International Conference on Field Programmable Logic and Applications, FPL, Amsterdam, Netherlands", August 2007.

[27] S. LE BEUX, P. MARQUET, J.-L. DEKEYSER. *Multiple Abstraction Views of FPGA to Map Parallel Application*, in "3rd International Workshop on Reconfigurable Communication Centric System-on-Chips, ReCoSoC, Montpellier, France", June 2007.

[28] S. LE BEUX, P. MARQUET, A. HONORÉ, J.-L. DEKEYSER. *A Model Driven Engineering Design Flow to Generate VHDL*, in "International ModEasy'07 Workshop, Barcelona, Spain", September 2007, http://www2.lifl.fr/modeasy/workshop.html.

[29] P. MARQUET, S. DUQUENNOY, S. LE BEUX, S. MEFTALI, J.-L. DEKEYSER. *Massively Parallel Processing on a Chip*, in "ACM Int'l Conf. on Computing Frontiers, Ischia, Italy", May 2007.

[30] I. R. QUADRI, S. MEFTALI, J.-L. DEKEYSER. *An MDE Approach for Implementing Partial Dynamic Reconfiguration in FPGAs*, in "16th International Workshop on IP Based System-on-chip, IP'07, Grenoble, France", December 2007.

[31] S. TAHA, A. RADERMACHER, J.-L. DEKEYSER, S. GERARD. *MARTE: UML-based Hardware Design from Modeling to Simulation*, in "Forum on specification and Design Languages - FDL'07, Barcelona - Spain", September 2007.

[32] S. TAHA, A. RADERMACHER, S. GÉRARD, J.-L. DEKEYSER. *An Open Framework for Detailed Hardware Modeling*, in "International Symposium on Industrial Embedded Systems (SIES '07)", July 2007.

[33] H. YU, A. GAMATIÉ, E. RUTTEN, JEAN-LUC. DEKEYSER. *Model Transformations from a Data Parallel Formalism towards Synchronous Languages*, in "Forum on specification and Design Languages - FDL'07, Barcelona - Spain", September 2007.

Internal Reports

[34] R. BEN ATITALLAH, P. BOULET, A. CUCCURU, J.-L. DEKEYSER, A. HONORÉ, O. LABBANI, S. LE BEUX, P. MARQUET, É. PIEL, J. TAILLARD, H. YU. *Gaspard2 UML profile documentation*, Technical Report, n^{o} 0342, INRIA, September 2007, http://hal.inria.fr/inria-00171137/en.

[35] P. BOULET. *Array-OL Revisited, Multidimensional Intensive Signal Processing Specification*, Research Report, n^{o} RR-6113, INRIA, feb 2007, http://hal.inria.fr/inria-00128840/en.

[36] A. ETIEN, C. DUMOULIN, E. RENAUX. *Towards a Unified Notation to Represent Model Transformation*, Research Report, n^{o} RR-6187, INRIA, May 2007, https://hal.inria.fr/inria-00145204.

[37] H. YU, A. GAMATIÉ, E. RUTTEN, JEAN-LUC. DEKEYSER. *Model Transformations from a Data Parallel Formalism towards Synchronous Languages*, Rapport de Recherche, n^{o} 6291, INRIA, September 2007, http://hal.inria.fr/inria-00172302.

References in notes

[38] OBJECT MANAGEMENT GROUP, INC. (editor). *U2 Partners' (UML 2.0): Superstructure, 2nd revised submission*, January 2003, http://www.omg.org/cgi-bin/doc?ptc/03-01-02.

[39] OBJECT MANAGEMENT GROUP, INC. (editor). *(UML 2.0): Superstructure Draft Adopted Specification*, July 2003, http://www.omg.org/cgi-bin/doc?ptc/03-07-06.

[40] M. D. ADAMS. *The JPEG-2000 Still Image Compression Standard*, Technical report, n^{o} N2412, ISO/IEC JTC 1/SC 29/WG 1, September 2001.

[41] A. AMAR, P. BOULET, P. DUMONT. *Projection of the Array-OL Specification Language onto the Kahn Process Network Computation Model*, in "International Symposium on Parallel Architectures, Algorithms, and Networks, Las Vegas, Nevada, USA", December 2005, http://sigact.acm.org/ispan05/.

[42] C. BASTOUL. *Code Generation in the Polyhedral Model Is Easier Than You Think*, in "PACT'13 IEEE International Conference on Parallel Architecture and Compilation Techniques, Juan-les-Pins", September 2004, p. 7–16.

[43] A. BENVENISTE, P. CASPI, S. EDWARDS, N. HALBWACHS, P. LE GUERNIC, R. DE SIMONE. *The Synchronous Languages Twelve Years Later*, in "Proceedings of the IEEE", vol. 91, n^{o} 1, January 2003, p. 64-83.

[44] G. BERRY. *The Foundations of Esterel*, in "Proof, Language and Interaction: Essays in Honour of Robin Milner", MIT Press, 1998, http://www-sop.inria.fr/meije/esterel/doc/main-papers.html.

[45] L. BONDÉ. *Transformations de Modèles et Interopérabilité dans la Conception de Systèmes Hétérogènes sur Puce à base d'IP*, Doctorat en Informatique, Université des Sciences et Technologies de Lille, December 2006.

[46] P. BOULET. *Array-OL Revisited, Multidimensional Intensive Signal Processing Specification*, Research Report, n^{o} RR-6113, INRIA, February 2007, http://hal.inria.fr/inria-00128840/en.

[47] P. BOULET, J.-L. DEKEYSER, J.-L. LEVAIRE, P. MARQUET, J. SOULA, A. DEMEURE. *Visual Data-parallel Programming for Signal Processing Applications*, in "9th Euromicro Workshop on Parallel and Distributed Processing, PDP 2001, Mantova, Italy", February 2001, p. 105–112.

[48] P. CASPI, D. PILAUD, N. HALBWACHS, J.A. PLAICE. *Lustre: a declarative language for real-time programming*, in "Proceedings of the 14th ACM SIGACT-SIGPLAN symposium on Principles of programming languages (POPL'87)", ACM Press, 1987, p. 178-188.

[49] A. DEMEURE, A. LAFAGE, E. BOUTILLON, D. ROZZONELLI, J.-C. DUFOURD, J.-L. MARRO. *Array-OL : Proposition d'un Formalisme Tableau pour le Traitement de Signal Multi-Dimensionnel*, in "Gretsi, Juan-Les-Pins, France", September 1995.

[50] A. DEMEURE, Y. DEL GALLO. *An Array Approach for Signal Processing Design*, in "Sophia-Antipolis conference on Micro-Electronics (SAME 98), France", October 1998.

[51] P. DUMONT, P. BOULET. *Transformations de code Array-OL : implémentation de la fusion de deux tÃ¢ches*, Technical report, Laboratoire d'Informatique fondamentale de Lille et Thales Communications, October 2003.

[52] P. DUMONT. *Étude des Transformations d'un Code Array-OL dans Gaspard*, Research Report, n^{o} 02-11, Laboratoire d'informatique fondamentale de Lille, Université des sciences et technologies de Lille, France, September 2002, http://www.lifl.fr/LIFL1/publications/2002-11.ps.

[53] P. DUMONT. *Spécification multidimensionnelle pour le traitement du signal systématique*, (In French), Thèse de doctorat (PhD Thesis), Laboratoire d'informatique fondamentale de Lille, Université des sciences et technologies de Lille, 2005.

[54] ECLIPSE CONSORTIUM. *The Eclipse Project*, 2003, http://www.eclipse.org.

[55] ECLIPSE CONSORTIUM. *EMF Eclipse Modeling Framework*, 2007, http://www.eclipse.org/emf.

[56] ESTEREL TECHNOLOGIES. *SoC Design, Validation and Verification*, 2002, http://www.esterel-technologies.com/v3/?id=29453.

[57] D. D. GAJSKI, R. KUHN. *Guest Editor Introduction: New VLSI-Tools*, in "IEEE Computer", vol. 16, n^{o} 12, December 1983, p. 11-14.

[58] C. GLITIA. *Code transformations for systematic signal processing and memory size optimizations*, Master recherche, Université des sciences et technologies de Lille, 2006.

[59] N. KADRI, S. NIAR, A. BABA-ALI. *Impact of Code Compression on the Power Consumption in Embedded Systems*, in "international conference on Embedded Systems and Applications ESA'03", June 2003.

[60] G. KAHN. *The Semantics of a Simple Language for Parallel Programming*, in "Information Processing 74: Proceedings of the IFIP Congress 74", J. L. ROSENFELD (editor), North-Holland, IFIP, August 1974, p. 471–475.

[61] G. KAHN, D. B. MACQUEEN. *Coroutines and networks of parallel processes*, in "Information Processing 77: Proceedings of the IFIP Congress 77", B. GILCHRIST (editor), North-Holland, 1977, p. 993–998.

[62] A. KOUDRI, D. AULAGNIER, J. CHAMPEAU. *FPGA design based on UML/MDA Approach: Application to an RF Tranceiver development*, in "4th workshop on tUML for SoC Design", June 2007.

[63] A. KOUDRI, D. AULAGNIER, J. CHAMPEAU. *Une sémantique opérationnelle pour une meilleure métamodélisation*, in "Atelier sur la Sémantique des Modèles", march 2007.

[64] O. LABBANI, É. RUTTEN, J.-L. DEKEYSER. *Safe Design Methodology for an Intelligent Cruise Control System with GPS*, December 2006, IEEE Intelligent Transportation Systems Society Newsletter, vol. 8, nr. 4, pp. 16-23.

[65] P. LE GUERNIC, J.-P. TALPIN, J.-C. LE LANN. *Polychrony for System Design*, in "Journal for Circuits, Systems and Computers", vol. 12, n^{o} 3, April 2003, p. 261–304.

[66] A. MEENA, P. BOULET. *Model Driven Scheduling Framework for Multiprocessor SoC Design*, in "Workshop on Scheduling for Parallel Computing, SPC 2005, Poznan, Poland", September 2005, http://www.cs.put.poznan.pl/mdrozdowski/spc-ppam05/.

[67] S. MEFTALI, J.-L. DEKEYSER, ISAAC D. SCHERSON. *Scalable Multistage Networks for Multiprocessor System-on-Chip Design*, in "International Symposium on Parallel Architectures, Algorithms, and Networks, Las Vegas, Nevada, USA", December 2005, http://sigact.acm.org/ispan05/.

[68] P. K. MURTHY, E. A. LEE. *Multidimensional Synchronous Dataflow*, in "IEEE Transactions on Signal Processing", July 2002.

[69] P. K. MURTHY. *Scheduling Techniques for Synchronous and Multidimensional Synchronous Dataflow*, Ph. D. Thesis, University of California, Berkeley, CA, 1996.

[70] S. NIAR, L. EECKHOUT, K. DEBOSSCHERE. *Comparing multiported cache schemes*, in "PDPTA-2003", June 2003.

[71] OBJECT MANAGEMENT GROUP, INC.. *MOF 2.0 Query / Views / Transformations RFP*, OMG paper, 2003.

[72] OBJECT MANAGEMENT GROUP, INC.. *Meta Object Facility (MOF) Core Specification, Version 2.0*, January 2006, http://www.omg.org/docs/formal/06-01-01.pdf.

[73] OPEN SYSTEMC INITIATIVE. *SystemC*, 2002, http://www.systemc.org/.

[74] OPENMP ARCHITECTURE REVIEW BOARD. *OpenMP Application Programme Interface*, May 2005, http://www.openmp.org/drupal/mp-documents/spec25.pdf.

[75] H. SBEYTI, S. NIAR, L. EECKHOUT. *Adaptive Prefetching for Multimedia Applications in Embedded Systems*, in "DATE'04, Paris, France", EDA IEEE, February 2004.

[76] D. SCHMIDT. *Model-Driven Engineering*, in "IEEE Computer", vol. 39, n^{o} 2, February 2006, p. 41-47.

[77] Y. SOREL. *Massively Parallel Computing Systems with Real Time Constraints - The "Algorithm Architecture Adequation" Methodology*, in "Proceedings of the 1st International Conference on Massively Parallel Computing Systems, Los Alamitos, CA, USA", IEEE Computer Society Press, May 1994, p. 44–54.

[78] J. SOULA. *Principe de Compilation d'un Langage de Traitement de Signal*, (In French), Thèse de doctorat (PhD Thesis), Laboratoire d'informatique fondamentale de Lille, Université des sciences et technologies de Lille, December 2001.

[79] J. TAILLARD, F. GUYOMARC'H, J.-L. DEKEYSER. *A Graphical Framework for High Performance Computing using an MDE Approach*, in "16th Euromicro International Conference on Parallel, Distributed and network-based Processing, Toulouse, France", to appear, February 2008.

[80] VIRTUTECH. *Simics Platform*, 2007, http://www.virtutech.com.