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 **1**
2. Overall Objectives **1**
2.1. Overall Objectives 1
3. Scientific Foundations **3**
3.1. Introduction 3
3.2. Co-modeling for SoC design 3
3.2.1. Principles 3
3.2.2. Transformations and Mappings 4
3.2.3. Use of Standards 4
3.2.4. System-on-Chip Design 4
3.2.5. Contributions of the team 5
3.2.5.1. Gaspard2 foundations 6
3.2.5.2. Application Metamodel 7
3.2.5.3. Hardware Architecture MetaModel 9
3.2.5.4. Association Metamodel 9
3.2.5.5. PSM Metamodels 10
3.2.5.6. Transformation Techniques 10
3.3. Optimization Techniques 11
3.3.1. Optimization for parallelism 11
3.3.2. Contributions of the team 11
3.3.2.1. Dataparallel Code Transformations 11
3.3.2.2. Multi-objective Hierarchical Scheduling Heuristics 12
3.4. SoC Simulation 12
3.4.1. Abstraction levels 12
3.4.2. Contribution of the team 13
3.4.2.1. Co-simulation in SystemC 14
3.4.2.2. Multilevel distributed simulation in SystemC 14
3.4.2.3. TLM: Transactional Level Modelling 14
3.4.2.4. Network on Chip (NoC) design and performances estimation 15
4. Application Domains **15**
4.1. Intensive Signal Processing 15
4.2. Automative Safety Embedded Systems 16
5. Software **16**
5.1. ModTransf 16
5.2. Gaspard2 17
6. New Results **19**
6.1. Contributions to meta-models for SoC design 19
6.1.1. Introducing repetition in meta-models for SoC design 19
6.1.2. Meta-computing: a new application domain 20
6.1.3. Towards other metamodels 20
6.1.4. Introducing control in meta-models for SoC design 21
6.1.5. Transformations between meta-models for SoC design 21
6.2. Quantitative aspects and Optimization Techniques 22
6.2.1. Multidimensional code transformations 22
6.2.2. Scheduling Algorithm Coupling 22
6.2.3. Performance Evaluation 22
6.3. Execution and simulation platforms for SoC 23

6.3.1. Operating system support 23
6.3.2. Early supports of FPGA 23
6.3.3. Transaction Level Modelling 23
7. Contracts and Grants with Industry 24
7.1. The PROTES Project: A Carroll Project 24
7.2. Collaboration with Prosilog 24
7.3. Collaboration with CEA List 25
7.4. Collaboration with Valeo - CNRT Futurelec 25
7.5. SoCLib RNRT Platform Project 25
8. Other Grants and Activities 25
8.1. ModEsay Interreg III A Franco/English Cooperation 25
8.2. International initiatives 26
8.2.1. ECSI 26
8.2.2. Center of Embedded Computer Systems, University of California 26
8.2.3. Université de Montreal 26
8.2.4. ENS Tunis 26
8.3. National initiatives 26
8.3.1. CNRS initiatives 26
9. Dissemination 26
9.1. Scientific Community 26
9.2. Teaching 28
10. Bibliography 28

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).

Team Leader

Jean-Luc Dekeyser [Professor, Université des Sciences et Technologies de Lille]

Administrative assistant

Karine Levandowski [INRIA]

Faculty members

Pierre Boulet [Professor, Université des Sciences et Technologies de Lille]
Philippe Marquet [Associate professor, Université des Sciences et Technologies de Lille]
Samy Meftali [Associate professor, Université des Sciences et Technologies de Lille]
Cédric Dumoulin [Associate professor, Université des Sciences et Technologies de Lille, since sept. 2005]

Research scientist

Eric Rutten [CR INRIA, since jan. 2005]

External Collaborator

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

Post-doctoral fellows

Anouar Dziri [Teaching Assistant, Université Lille 2, until aug. 2005]
Luc Charest [Post-doctoral fellow, INRIA Futurs, until nov. 2005]
Abdoulaye Gamatié [Post-doctoral fellow, INRIA Futurs, since sept. 2005]

Ph. D. student

Yassine Aydi [co-advising USTL/ENIS Tunis, since dec. 2005]
Rabie Ben Atitalah [Interreg project grant]
Lossan Bondé [Burkina Fasso grant]
Arnaud Cuccuru [ITEA project grant, until oct. 2005]
Philippe Dumont [Teaching Assistant, Université des Sciences et Technologies de Lille, until aug 2005]
Ali Houdri [CIFRE grant, Thalès, ENSIETA Brest, since dec. 2005]
Ouassila Labbani [CNRS and regional grant]
Sébastien Le Beux [Interreg project grant]
Ashish Meena [Interreg project grant]
Éric Piel [INRIA Futurs and Regional grant]
Mickaël Samyn [French ministry grant, until sept. 2005]
Safouan Taha [co-advising, CEA grant]
Julien Taillard [VALEO and Regional grant, since oct. 2005]
Joël Vennin [CIFRE grant, Prosilog]
Huafeng Yu [INRIA Futurs and Regional grant, since oct. 2005]

Technical staff

Stéphane Akhoun [ITEA project grant, until march 2005]
Antoine Honoré [Interreg project grant, since dec. 2005]
Christophe Osuna [ITEA project grant, until june 2005]

2. Overall Objectives

2.1. Overall Objectives

The 2001 International Technology Roadmap for Semiconductors [82] stresses a new problem in the design of electronic systems. Indeed, we face for the first time a design productivity gap, meaning that electronic

system design teams are no longer able to take advantage of all the available transistors on a chip for logic. Because of the superexponential increase of the difficulty of system design, we may well be in a situation in a few years where one could be forced to use more than 90% of a chip area for memory, because of design costs for anything other than memory.

In the same time, the processing 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:

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

All these ideas will be implemented into a prototype design environment based on simulation, Gaspard. This open source platform will be our test bench and will be freely available.

The main technologies we promote are UML 2.0 [56], MDA [52], MOF [53] for the modeling and the automatic model transformations; Array-OL [71], [72], [67], synchronous languages (such as Esterel [64] or Lustre [81]), Kahn process networks [83] as computation models with strong semantics for verification; SystemC [93] for the simulations; VHDL for the synthesis; and Java [63] to code our prototypes.

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 [76]). This also permits to produce automaticaly a mapping and a schedule of the application onto the architecture with code generation (as with the AAA method of SynDEx [99]). The DaRT project contributes to this research field by the three following items:

Co-modeling for 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 UML profiles in respect to the MOF specifications.

Optimization 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: The data flow philosophy of our metamodel is particularly well suited to a distributed simulation. To take care of the architecture model and the mapping of the application on it, we propose to 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 integration with different levels of specification (functional, timed functional, transaction and cycle accurate byte accurate levels).

3.2. Co-modeling for SoC design

Keywords: *MDA, MDA Transformation, MOF, Metamodel, Model, Modeling, 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 MDA (Model Driven Approach) based approach.

3.2.1. Principles

Because of the vast scope of the encountered problems, of the quick evolution of the architectures, we observe a very great diversity as regards the 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 releaving 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 as well 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:

- To refrain from designing programming languages to express the two different models, application and hardware architecture.
- To profit from all the new systems dedicated to simulation or synthesis without having to reformalize these two models.
- To use a single modeling environment possibly supporting a visual specification.
- To benefit from standard formats for exchange and storage.
- To be able to express transformation rules from model to model. Possibly the transformation tools could be generated automatically from this expression.

We believe that the Model Driven Architecture [52], [65] can enable 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 clear separation between the models and the platforms makes it easy to switch to a new technology while re-using the old designs. This may even be done automatically provided the right tools. The MDA is the OMG proposed approach for system development. It primarily focuses on software development, but can be applied to any system development. The MDA is based on models describing the systems to be built. A system description is made of numerous models, each model representing a different level of abstraction. The modeled system can be deployed on one or more platforms via model to model transformations.

3.2.2. Transformations and Mappings

A key point of the MDA is the transformation between models. The 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 we deduce the transformations between any models conforming to the metamodels.

The MDA model to model transformation is in a standardization process at the OMG [91].

3.2.3. Use of Standards

The MDA is based on proven standards: UML for modeling and the MOF for metamodel expression. The new coming UML 2.0 [55] standard is specifically designed to be used with the MDA. It removes some ambiguities found in its predecessors (UML 1.x), allows more precise descriptions and opens the road to automatic exploitation of models. The MOF (Meta Object Facilities [92]) is oriented to the metamodel specifications.

3.2.4. 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 as much the application modeling by the aggregation of functional components, as the assembly of existing physical components, as the verification and the simulation of the modeled system, as 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.

3.2.5. Contributions of the team

Our proposal is partially based upon the concepts of the "Y-chart" [77]. The MDA 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.

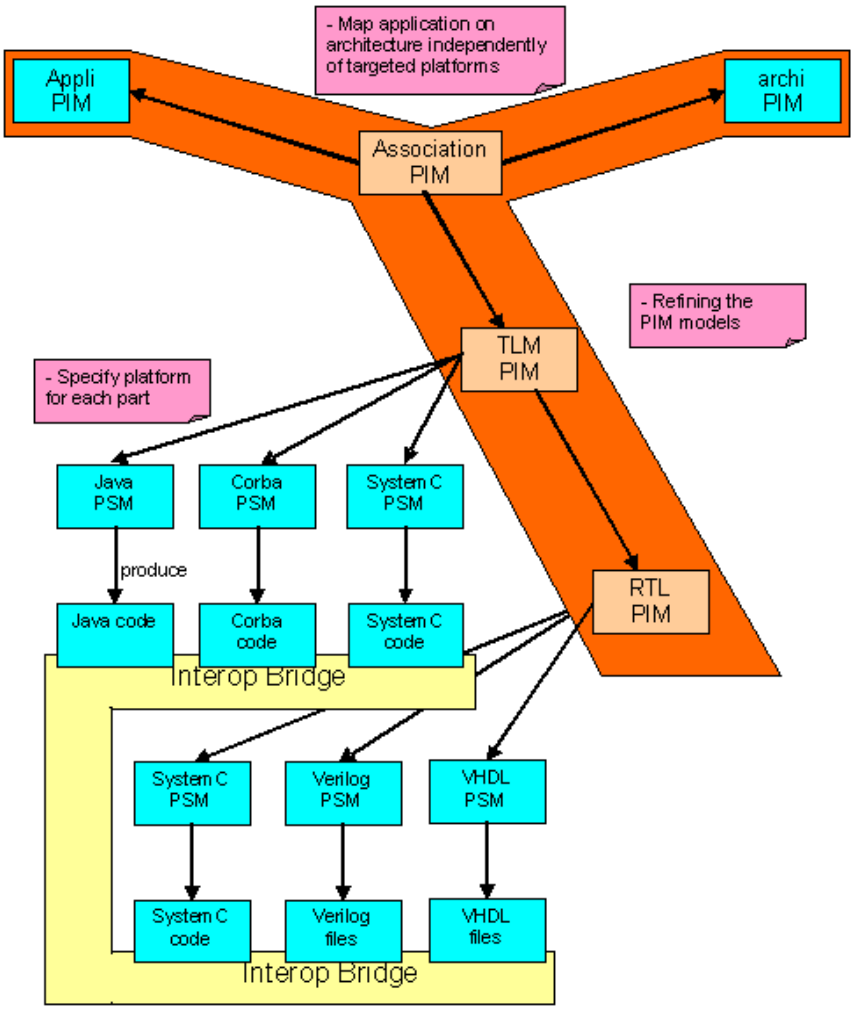

Figure 1. Overview of the metamodels for the "Y" design

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. For this purpose we introduce a third metamodel, named association metamodel, to express associations between the functional components and the hardware components. This metamodel imports the two previously presented metamodels.

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 (Java, SystemC RTL, SystemC TLM, VHDL, etc). Once all the components are associated with some technology, the deployment is realized. This is done by the refinement of the PIM association model to the PIM TLM model first (Transaction Level Model), and to the PIM RTL model second (Register Transfer Level).

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 technologies.

Mapping rules between the deployment metamodel, and interoperability and technology metamodels can be defined to automatically specialize the deployment model to the chosen technologies. 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. Separate application and hardware architecture modeling.
2. Association with semi-automatic mapping and scheduling.
3. Deployment (choice of simulation or execution level and platform for each component).
4. Automatic generation of the various platform specific simulation or execution models.
5. Automatic simulation or execution code generation.
6. Refinement at the PIM level given the simulation results.

3.2.5.1. Gaspard2 foundations

The abstract syntax of application and hardware architecture are described by different MOF meta-models. 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 that the conception of embedded systems implies.

The two meta-models also share 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 an 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 Gaspard profile (a detailed specification can be found in [15]) is oriented by the few guiding ideas mentionned 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 MOF (Domain view point). The main evolutions according to past versions of the profile are:

- a better structuration 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)[1].

The architecture of the Gaspard profile is now structured around five packages (fig.2):

- **component package**: Extensively reuse mechanisms defined in UML 2 [57](composition, assembling), with some minor but interesting extensions (capability to parameterize components/IPs, which is a common practice for IP providers). The main goal of this package is to ease and encourage reuse of components. As illustrated by the `import` dependencies, application and hardware architecture packages share a common definition of the Component concept
- **factorization package** [30]: Introduces mechanisms enabling to express in a compact way a repetition of structural elements and the regular topology of the links between them. As illustrated by the `use` dependencies, component (and as a consequence application and hardware architecture) and association packages use these common mechanisms. This proposal is partially inspired by the Array-OL language [71]
- **hardware architecture package** [22]: Enables to dimension an hardware architecture and the resources that compose it, and to describe the topology of their interconnections. The structure of this proposal is inspired by the SPT profile [54].
- **application package**: Defines a simple but powerful design pattern for modelling of application only via data dependences expression.
- **association package** [27]: Offers mechanisms to take into accound two aspects of association, i.e characterization (put into action via the QoS profile [58]) and allocation (via the Allocation concept of SysML [59]). The concepts introduced enable to express mapping directives for allocation of data and tasks of an application onto an hardware architecture, which is itself oriented by the characterizations.

3.2.5.2. Application Metamodel

The application metamodel focuses on the description of data dependences 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 dependences. This gives many benefits:

- simple description of the algorithm,
- no dependence 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 multidimentional arrays, with one possible infinite dimension representing time.

The application meta-model 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 dependences.

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.

[1] In [15], we also show why UML for SoC [60] is not relevant for our goals

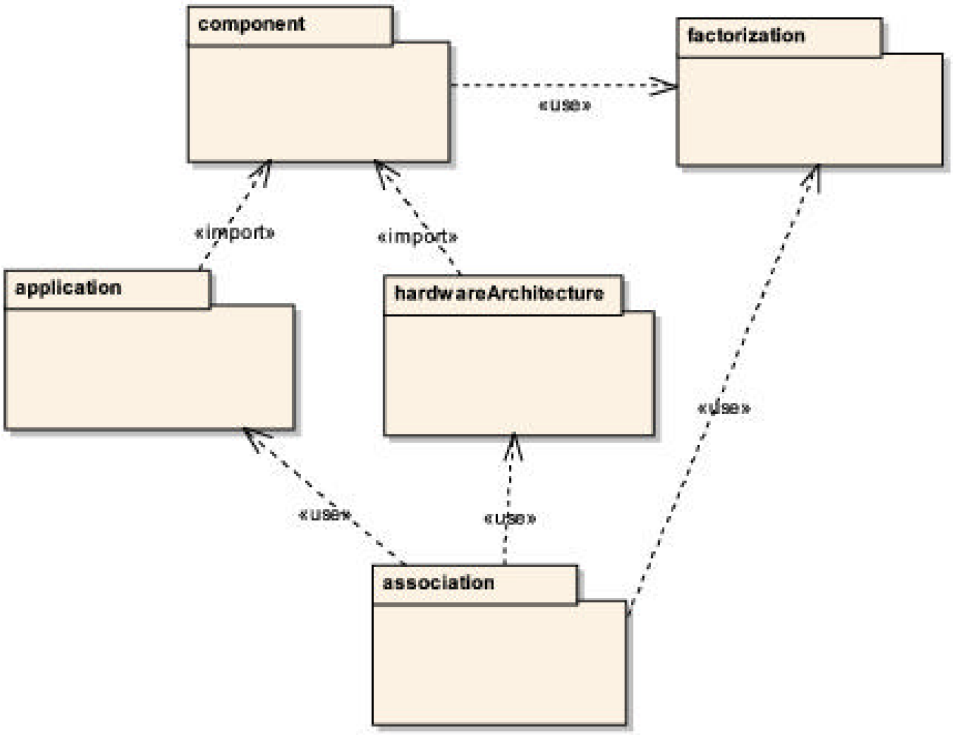

Figure 2. Gaspard Profile Architecture

Data parallelism expression is one of the key point of our approach. In domains such as intensive signal processing or telecommunication (typically targeted by embedded systems), applications generally present lot of potential data parallelism.

In order to broaden the application domain of our metamodel, we have also studied a design methodology for synchronous reactive systems, based on a clear separation between control and data flow parts. This methodology allows to facilitate the specification of different kinds of systems and to have a best readability. It also permits to separate the study of the different parts by using the most appropriated existing tools for each of them. Following this idea, we are particulary interested in the notion of running modes and in the Scade tool. Scade is a graphical development environment coupling data processing and state machines (modeled by synchronous languages Lustre and Esterel). It can be used to specify, simulate, verify and generate C code. However, this tool does not follow any design methodology, which often makes difficult the understanding and the re-use of existing applications. We will show that is also difficult to separate control and data parts using Scade. Thus, regulation systems are better specified using mode-automata which allow adding an automaton structure to data flow specifications written in Lustre. When we observe the mode-structure of the mode-automaton, we clearly see where the modes differ and the conditions for changing modes. This makes it possible to better understand the behavior of the system.

3.2.5.3. 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 ressources of the hardware in a precise enough way to be pertinent, but abstracting irrelevant details so that efficient decision could be taken.

A mechanism similar to the one used in the application meta-model enables to specify repetitive architecture in a compact way. We believe that regular parallel computation units will be more and more present in embedded in systems in the future, especially for Systems on Chips. This belief is driven by two considerations:

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. Parallelism is a good way to reduce power consumption in SoCs. Indeed at equal computing power, a chip able to run several computations simultaneously will be 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.

3.2.5.4. 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 it is associated with will be implemented in software, otherwise, they will be synthesized as hardware. The dependences 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 granularity 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. The idea of these code transformations is to label a hierarchical level of the application model with an execution strategy such as sequential, SPMD, cyclic(k) or block in order to unambiguously specify the distribution and schedule of this level on a given hierarchical level of the hardware architecture model.

3.2.5.5. PSM Metamodels

We will focus here on two particular abstraction levels: Transaction Level Model and Register Transfer Level. The metamodels appearing at the PIM level are not complete metamodels of the targeted language but rather metamodels providing the concepts needed to execute the mapped application with these abstraction levels. Then a transformation stage will generate PSM SystemC (for example) from the PIM TLM. By refinement the PIM TLM is transformed into a PIM RTL. At last the PIM RTL can be transformed to the PSM VHDL (for example). Code generations are produced from the PSM models using a transformation tool. For more details, see the section on simulation techniques 3.4.2.1.

3.2.5.6. Transformation Techniques

Model to model transformations are at the heart of the MDA approach. Anyone whishing to use MDA in its projects is sooner or later facing the question: how to perform the model transformations? There are not so much publicly and freely available tools, and the OMG QVT standardization process [91] is not completed today. To fulfill our needs in model transformations, we have developed ModTransf, a simple but powerful transformation engine. ModTransf was developed based on the recommendations done after the review of the first QVT proposals and on the latest proposals. Based on these recommendations and on our needs, we have identified the following requirements for the transformation engine:

- Multi models as inputs and outputs
- Different kind of models: MOF and JMI based, XML with schema based, graph of objects
- Simple to use
- Easy modification of rules to follow metamodel changes
- Hybrids: Imperative and declarative rules
- Inheritance for the rules
- Reversible rules when possible
- Customizable, to do experimentations
- Code generation
- Free and Open-Sources.

The proposed solution fulfills all these needs: ModTransf is a rule based engine taking one or more models as inputs and producing one or more models as outputs. The rules can be expressed using an XML syntax and can be declarative as well as imperative. A transformation is done by submitting a concept to the engine. The engine then searches the more appropriate transformation rule for this concept and applies it to produce the corresponding result concept. The rule describes how properties of the input concept should be mapped, after a transformation, to the properties of the output concept.

The code generation follows the same principle, but the output concept creation is replaced by code generation performed with a template mechanism. A rule specifies one or more template to use, and each template contains holes replaced by the values of the input concepts.

The ModTransf engine is an Open Source project available on the internet.

3.3. Optimization Techniques

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

3.3.1. Optimization for parallelism

We study optimization techniques to produce a schedule and a mapping of a given application onto a hardware SoC architecture. These heuristic techniques aim at fullfilling 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 dependences –so no dependence analyzis 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. Many optimization techniques have been studied that can be useful in our case. These techniques cover several fields of compiler construction:

- Automatic parallelization [66], [86], [70], [62], [69] with loop transformation, scheduling and mapping techniques.
- Memory management [85], [87], [101] to reuse the storage space while preserving parallelism.
- Pure functional language compilation [96], [94], [80], [88] with techniques such as static typing, higher order functions, derecursivation, partial evaluation, etc.
- Signal processing specific optimizations [95].

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

In some of our previous works have studied Array-OL to Array-OL code transformations [67], [100], [74], [73]. Array-OL [71], [72] is a dataparallel language dedicated to systematic signal processing. It allows a powerful expression of the data access patterns in such applications and a complete parallelism expression. It is at the root of our model of applications.

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 study the relations of the Array-OL model with other computation models such as Kahn Process Networks [83], [84] and multidimensional synchronous dataflow [90], [89].

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 fertilze 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. It is the case in the AAA methodology [99], [98], [79] that is implemented in the SynDEx [97] tool. Both our application and hardware architecture models are hierarchical and allow repetitive expressions. We believe that we can take advantage of these hierarchical and repetitive expressions to build more efficient schedules. Further more, local optimizations (contained inside a hierarchical level) will surely decrease the communication overhead and allow a more efficient usage of the memory hierarchy. We aim at integrating the dataparallel code transformations presented before in a global heuristic in order to deal efficiently with the dataparallelism of the application by using repetitive parts of the hardware architecture.

Furthermore, 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 [10], [11].
- The study of code compression techniques to reduce the memory requirements of an embedded application [7].
- Clock scaling to choose the slowest speed that satisfies the real-time constraints.

We plan to use these results to build our scheduling heuristic.

3.4. SoC Simulation

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 and 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 consist in refining, more or less automatically, each model to obtain another, starting from a functional model to reach a Register Tranfert Level model. One of the biggest design challenge 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

Design flow systems allow the description of system modules (IPs) mainly at six levels of abstraction (this is the case in the standardization effort of SystemC[2]:

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.

[2] http://www.systemc.org

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.

3.4.2. Contribution of the team

The results of DaRT simulation package concerns mainly the UTF level and the TLM level. We also propose techniques to intercat with IPs specified at other level of abstraction (mainly RTL).

At the UTF level: we have developed a Distributed Kahn Process Network environment. The result of this simulation guarantees the functionality of the application model. By the observation of the FIFO sizes we are able to transform the application to improve the load balance of the system. The distributed aspect of this simulator permits to associate IPs from different builders available on different websites.

At TLM level: From the association model of our "Y-model", we are able to simulate the application and the architecture of the SoC in the same time. The results expected from this simulation cover the schedule of elementary tasks, the mapping of the data parallel structure on hierarchical and parallel memories, and the communications involved by this mapping. At this level, our models still PIM.

At RT level: In order to get physical implementations of our applications, we are developing an RTL metamodel. Models at this level will be obtained by transformation from those represented at TLM.

At SystemC level: we propose some generic wrappers to allow multilevel abstraction interoperability. A special effort was done to support distributed and heterogeneous simulation framework (see figure 3).

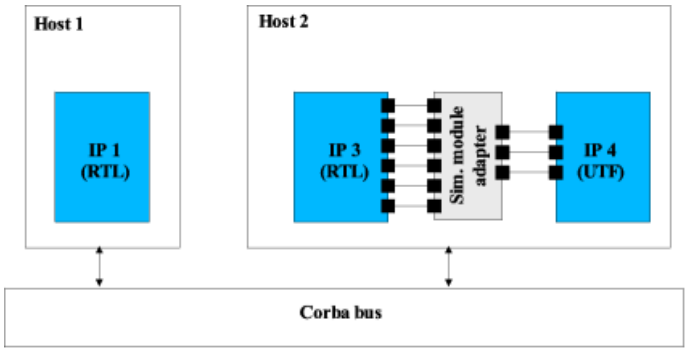

Figure 3. Distributed SystemC Simulation

3.4.2.1. Co-simulation in SystemC

From the association model, the Gaspard environment is able to produce automatically SystemC simulation code. The MDA techniques offer the transformation of the association model to the SystemC Gaspard 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.

A PSM metamodel allows automatic SystemC code generation. A PIM association model is first transformed into a model of this PSM metamodel and this model is then automatically transformed into SystemC code. This developpment is integrated in the Gaspard prototype and uses the ModTransf tool (see the software section).

3.4.2.2. Multilevel distributed simulation in SystemC

A multilevel simulation model is an executable specification containing a set of modules described at different abstraction level (ex an UTF IP coupled with an RTL IP). Our contribution is the proposal of a new methodology to validate SoCs by simulation [8]. With this new approach, we can perform a fast and low cost simulation of an assembly of IPs. At the opposite of existing solutions, we do not impose the usage of external libraries. Our solution is based on an internal SystemC library and a rule description language. We generate a simulation module adapter to encapsulate one of the two interconnected modules.

In the same idea of IP integration, we develop a distributed runtime for SystemC using sockets or Corba [9]. With this first implementation of a distributed SystemC, it is now possible to create a SoC with IPs selected from different providers.

Both the multilevel of abstraction runtime and the distributed runtime offer to SystemC the possibility to support a real co-design from world distributed IP providers.

3.4.2.3. TLM: Transactional Level Modelling

Transactional Level modelling (TLM) appeared during the very few last years. It consists in describing systems following defined specifications of some abstraction levels called TLM levels. In these later communication uses 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

Now-a-days, 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.

Due to all TLM's benefits, we defined a TLM meta model as a top level point for automatic transformations to both simulation and synthesis platforms. Our TLM meta model 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 composed mainly by computation tasks. They should be hierarchical and repetitive. A set of parameters could be attached to each task in order to specify mainly the scheduling dependently of the used computation model. Thus this meta model 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 his TLM model into a simulation model, and to keep it when transforming into a synthesis model.

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 [39].

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 will 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 example applications from our industrial partners.

Software Radio Receiver This emerging application is structured in a front end systematic signal processing including signal digitalization, 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 [61]. 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. Automative Safety Embedded Systems

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] lists 21 electronic systems that may be present in the modern automobile, some of which have the potential to endanger the safety of the 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.

5. Software

5.1. ModTransf

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

Participants: Cédric Dumoulin [contact person], Lossan Bondé.

The ModTransf tool performs model to model transformations according to transformation rules expressed in XML, and code generation from models.

The ModTransf tool allows to perform transformation of models by writing transformation rules. The tool takes one or more models and some transformation rules as input, and provides one or more transformed

models as output. The ModTransf tool works as well on models based on metamodels, on models based on XML schema or DTD, or on graphs of objects.

Transforming a model is done by submitting a concept to the engine. The engine then selects the more appropriate rule for this concept and applies it. Schematically, a rule specifies the concepts it requires as input, the concepts it provides as output, and how attributes of the source concepts are mapped on attributes of the target concepts. This attribute mapping may call recursively the engine, allowing to walk across the input models to produce the output models.

The transformation rules can be written using an XML syntax. The concepts are identified by their names from the MOF metamodels, or from the XML schemas.

The tool can also be used to generate code from a model. This is achieved by specifying transformation rules that will produce the code. A rule is then associated to a template containing the code and some holders to be replaced by values from the model concepts.

Though our research domain is not the model to model transformation techniques, we need some tool to realize our prototypes. Thus we have developped in a very pragmatic way this transformation tool for the MDA. We do not aim at completeness but at a tool which enables us both to map a PIM model to a PSM model in a deterministic way and to generate code. Nevertheless, this tool follows the remarks done on the QVT proposals [78], and will follow the evolutions of this standard.

The tool is available as an open source distribution [75]. It is currently evaluated by other INRIA teams and external teams (CEA, academics).

5.2. Gaspard2

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

Participants: Pierre Boulet [contact person], Stéphane Akhoun, Arnaud Cuccuru, Mickaël Samyn, Lossan Bondé, Christophe Osuna.

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.

Gaspard2 is an Integrated Development Environment (IDE) for SoC visual co-modeling. 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 PSM models)
- Code generation
- Simulation
- Reification of any stages of the development

The Gaspard2 tool is based on Eclipse [68]. A set of plugins provides the different functionalities. 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 the XMI file format. Gaspard2 manipulates these models through repositories (Java interfaces and implementations) automatically generated thanks to the JMI standard.

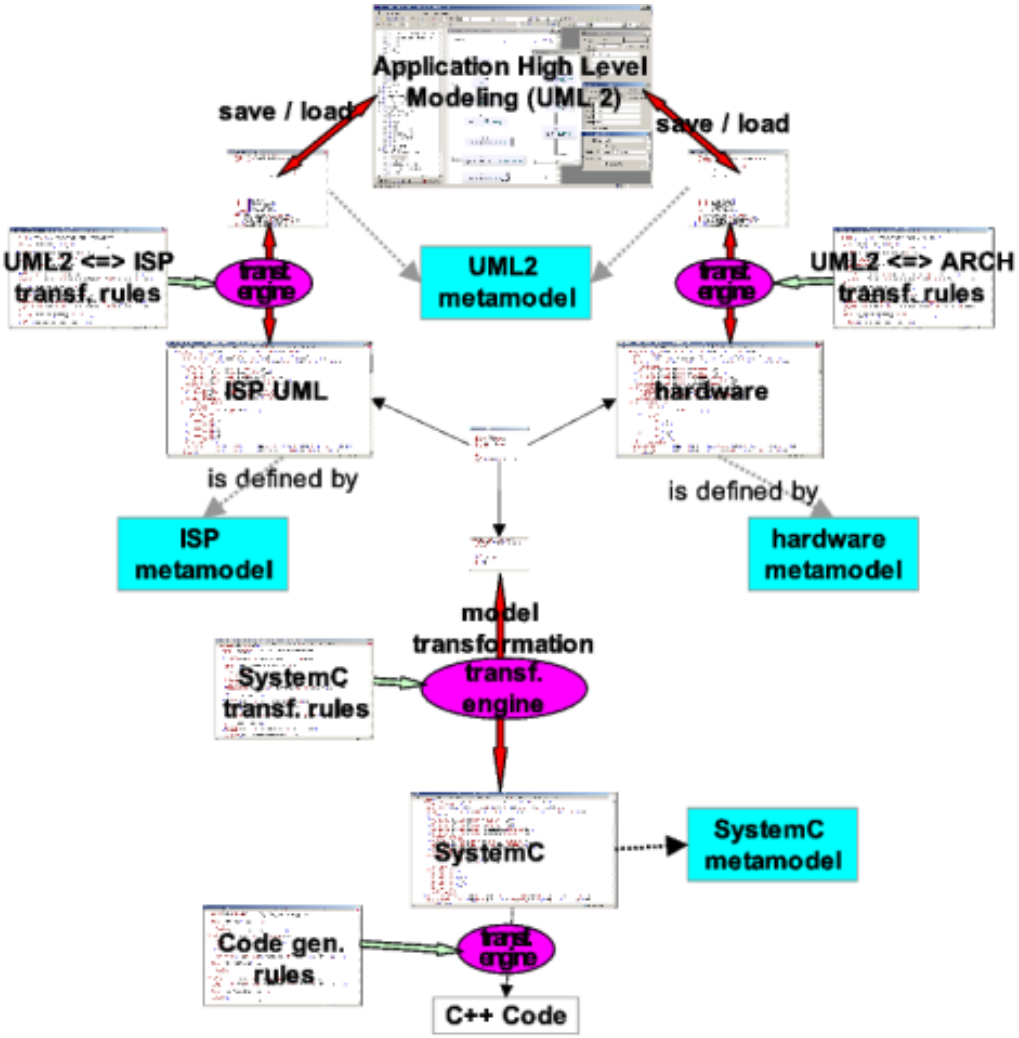

Figure 4. Overview of the Developement Flow with Gaspard2

6. New Results

6.1. Contributions to meta-models for SoC design

Participants: Lossan Bonde, Pierre Boulet, Arnaud Cuccuru, Jean-Luc Dekeyser, Cédric Dumoulin, Abdoulaye Gamatié, Ouassila Labbani, Philippe Marquet, Eric Rutten, Safouan Taha, Julien Taillard, Huafeng Yu.

6.1.1. Introducing repetition in meta-models for SoC design

One of the most interesting evolution of our proposal for the Gaspard2 profile is related to the factorization package. The mechanisms for compact modeling of regular complex topologies between repeated elements it offers are now clearly made independent of application and hardware architecture. It enables to specify that the multiplicity of a collection of structural element is multidimensional (shape and size of the collection is represented by a Vector of integers, as illustrated in fig.5), and introduce the `LinkTopology` (fig.6) concept enabling to specifiy at design time all the links that may exist at run time. Component (and as a consequence application and hardware architecture) and association packages extensively use these common mechanisms for various purposes:

- **application:** Compact expression of all the potential data parallelism of an application, where temporal and spatial aspects of computation are not unnecessarily differenciated,
- **hardware architecture:** Compact modeling of regular MPSoCs and their regular interconnection networks,
- **association:** Compact expression of the regular spatial allocation of data and tasks. Works are in progress in order to use the same factorization mechanisms to express also a temporal allocation.

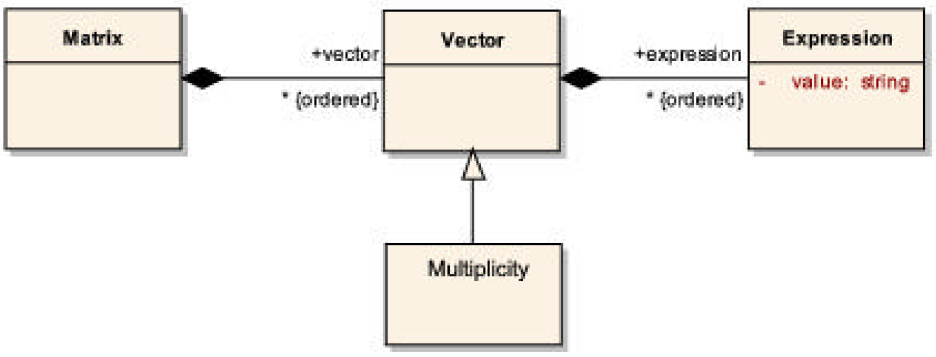

Figure 5. Multiplicity definition

Works performed for the definition of the profile (and especially the factorization mechanisms) are currently integrated in a response to the OMG's Marte request for proposal (Modeling and Analysis of Real Time and Embedded systems) [14] [43].

Note that [31], based on a previous version of the profile, has also been published this year.

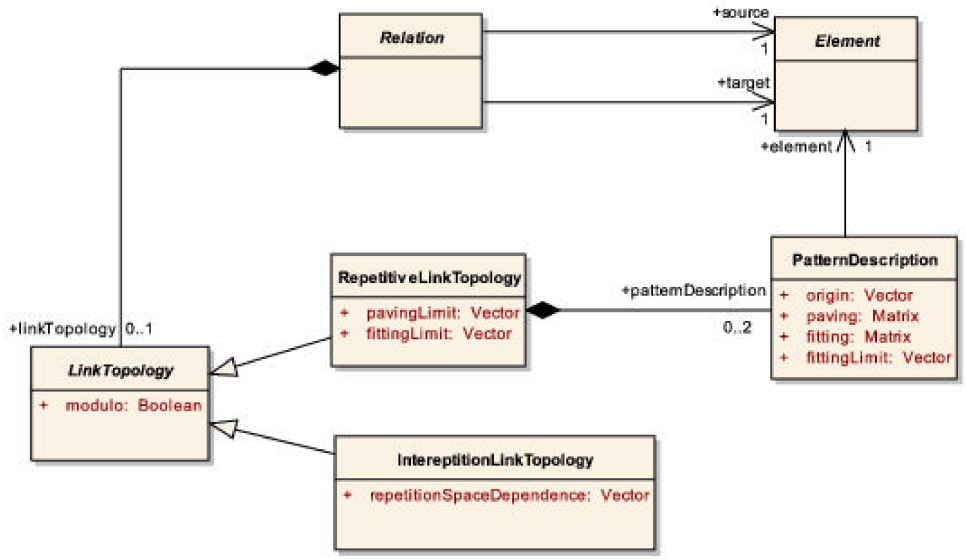

Figure 6. LinkTopology definition

6.1.2. Meta-computing: a new application domain

In collaboration with Valeo and L2EP (CNRT Futurelec) we are providing a UML profile for meta-computing based on a specialization of the Gaspard MetaModel [44]. It is quite intellectually pleasant to use the same models for System on Chip and meta-computing. We will model high performance software and hardware architecture, map software on hardware, and automatically generate code for the targeted hardware architecture. We will validate this method on a 3D electromagnetic simulation code which has been developed in L2EP (Laboratoire d'Electrotechnique et d'Electronique de Puissance de Lille). The Platform Specific Models are quite different, they are based on OpenMP, MPI and other metacomputing environments.

6.1.3. Towards other metamodels

The Accord/UML methodology is developed by CEA. It proposes UML extensions for modelling real-time systems with high level abstractions. Following the MDA paradigm, Accord/UML ensures automatic developing, from system specification to implementation, through three phases :

- PAM : Preliminary Analysis Model.
- DAM : Detailed Analysis Model
- PrM : Prototype Model.

Then, the PrM is connected with the Accord Kernel (virtual machine and Accord library) to generate the executable model.

Collaboration works want to benefit of the Accord/UML RTOS support in Gaspard. Reciprocally repetitive hardware could benefit the Accord/UML environment. Premilinary results cover:

- Adaptation of the Accord Kernel to VxWorks.
- Implementation of a chronometer prototype and its experimentation.

- Automation of the modeling flow : DAM → PrM.
- Code generation from the PrM model over an eclipse environment (Rational Software Development Platform).

6.1.4. Introducing control in meta-models for SoC design

First results in introducing control in data-parallelism concern the definition of a notion of automaton managing modes of computation. In this work, we introduce the control concepts in the Gaspard2 data parallel application metamodel by using the synchronous reactive system principles. It gives a *reactive* behavior and requires a good definition of the *degree of granularity* for these applications. This concept allows to delimit the different execution cycles or *clock signals* in which it becomes possible to take the control values and then the various changes in the running modes into account. Changing the computation modes expresses generally the is done in reaction to an execution context which can depend on an event from its external environment, or on an internal computation result. In both cases, it is important to delimit the various moments in which the change of modes become possible, in particular in the case of parallel applications [36]. We have proposed a UML solution for the modeling of control automata, the different running modes of an application and the link between the control and the data parallel parts. This metamodel is integrated in Gaspard2 profile and allows to study more general parallel systems mixing control and data processing.

The synchronous assumption and the degrees of granularity concept can impose a partial order on the calculation of the different parallel tasks since they make it possible to introduce the *flow* concept in the Gaspard2 application metamodel. However, the proposed model, introducing control in the data parallel applications, respects the parallelism and the concurrency, it is deterministic, compositional and can be easily introduced in the Gaspard2 application metamodel. Our approach is also based on a clear separation between control and data parts [35] which allows to have a more readable model since its different parts can be studied separately by using the most appropriate existing tools for each part. This concept has been validated by studying a real automative system application (Cruise Control with GPS).

We are exploring how notions of clocks of reactive systems can be used in our framework, in particular in relationship with the representation of time in the multidimensional framework. Clocks are generally very useful in high-level descriptions. They allow to address several design issues among which we can mention the verification of synchronizations between application sub-parts, the efficient scheduling algorithms, distribution strategies on multi-processor architectures, optimized generated code, etc. Clocks therefore appear together with modes very attractive in the description of data-parallel applications. Our current study investigates the way clock notions can be introduced in the Gaspard meta-model. Some critical points have been identified and discussed (e.g. where and how clocks can be taken into account within the Gaspard design flow). There are a few promising solution ideas from the literature on which we are working in order to adapt them to Gaspard.

Based on these models of control, we have started to consider transferring our results, in cooperation with the POP ART project in Grenoble, on the application of dicrete controller synthesis within a domain-specific language [1], [33], [49]. Once the control of data-parallel computation tasks is modelled with transition systems, these domain-specific language notions can be transferred into a specification level in Gaspard2. Other work on the application of discrete controller synthesis to fault-tolerance, in cooperation with the POP ART project, has lead to the definition of control mechanisms, that should also be transferable in our data-parallel computation framework.

6.1.5. Transformations between meta-models for SoC design

In the new version of Gaspard (Gaspard2 v0.2.0), transformations are at the heart of the design flow. The specification of the application and of the hardware architecture are done in UML tools. The two obtained models can then be imported in the Gaspard2 tool, that allows for the association of application and hardware architecture. This model of the association then becomes the input of a transformation towards a TLM model, from which, in turn, we plan the projection towards several simulation and execution platforms: SystemC, Java, Corba, Ptolemy. Recent progress concerned:

- integration of the transformations into Gaspard2,
- interoperability between model transformations. Given that we want to make projection from TLM towards several different simulation platforms, it is imperative to deal wqith interoperability between target systems. We propose a solution based on the use of a model of transformation trace [26], which is currently being implemented.

We have continued our effort to develop transformations between our metamodels. Also, the ModTransf transformation tool has evolved to take into account latest QVT proposals.

6.2. Quantitative aspects and Optimization Techniques

Participants: Rabie Ben Atitalah, Pierre Boulet, Jean-Luc Dekeyser, Philippe Dumont, Philippe Marquet, Ashish Meena, Smaïl Niar.

Our work has progressed in three directions last year: multidimensional code transformations, the coupling of different scheduling algorithms and performance evaluation.

6.2.1. Multidimensional code transformations

The toolbox allowing to transform data-parallel Array-OL code is now complete with the fusion, the change paving, the one level and the tiling transformations. All these transformations have been precisely defined and are described in detail in [16]. They have allowed us to study the projection of the Array-OL specification lanuage onto two models of computation: the synchronous data-flow and the Kahn process networks [24], [48], [50].

6.2.2. Scheduling Algorithm Coupling

Our objective is to handle the hierarchical and repetitive expressions [27] to build more efficient schedules. We call this approach globally irregular, locally regular (GILR). We have shown in [28] that GILR heuristics can improve the optimization in several ways:

1. Better optimization, (reduced latency)
2. Faster optimization, (reduced complexity)
3. More compact generated code (regular mapping i.e. loops)

Furthermore, in embedded systems, minimizing the total execution time of the application is usually not a good and only objective function. Because we must respect some real time constraints thus it is not useful to run faster than these timing constraints.

Thus we have done a survey of such multi-objective techniques. In order to build a schedule which respects the real time constraints of the application while minimizing the resource usage as well the above mention objectives. Keeping these objective in mind we have proposed a Scheduling and Mapping Framework [38]. This framework is an effort to mix and evaluate the outputs of different scheduling heuristics.

6.2.3. Performance Evaluation

Nowadays, multiprocessor system-on-chip (MPSoC) architecture becomes an incontrovertible solution for embedded systems designed for applications that requires intensive parallel computations. MPSoC are generally heterogeneous i.e. they contain different types of memories (SRAM, eDram, scratchpad, FIFO,...), different types of processors (MCU, DSP,...), different types of interconnecting network (Bus, Crossbar, Network-on-Chip,) and I/O peripherals. They may also contain dedicated hardware (in form of FPGA or ASIP). This heterogeneity makes the selection of the best configuration, or the domain space exploration (DSE), one of the most important design challenges. Increases in clock frequency, IP multiplicity and silicon integration are accompanied by a dramatic increase in energy consumption. Thus, in addition to the traditional performance criteria, such as area and execution speed, it has become imperative to take into account the energy consumption criterion when designing MPSoC. In this project we intend to provide the MPSoC

designer alongside of his/her design flow, a set of tools capable of estimating performance and energy consumption at different abstraction levels. We think that these tools will offer interesting features for the DSE at these various levels. Moreover, the availability of such estimators at these levels will allow the designers to make fast decisions at early design stages, reduce the exploration space, shorten the time to market and increase the design team's productivity. To our knowledge, there are very few tools that allow rapid and accurate energy consumption evaluation for MPSoC. To remedy this problem, we start developing a flexible architectural exploration environment that allows energy consumption to be estimated from the cycle-accurate bit-accurate (CABA) level up to functional or behavioural level. Our preliminary experimental results have been recently presented [41], [25]. Our future work will focus on defining simulation models for higher abstraction level (Timed Programmer View level and Timed Communicating Process). This definition should respect the UML meta-model specification of Gaspard, the simulation models will be improved with performance and power estimation techniques allowing fast and accurate design space exploration.

6.3. Execution and simulation platforms for SoC

Participants: Pierre Boulet, Jean-Luc Dekeyser, Anouar Dziri, Sébastien Le Beux, Samy Meftali, Smaïl Niar, Éric Piel, Mickaël Samyn, Julien Taillard, Joël Vennin.

6.3.1. Operating system support

Multiprocessor real-time operating systems being identified as a possible platform for Gaspard applications, some work has already been done to define ARTiS, an **A**symmetric **R**eal-**Ti**me **S**cheduler. ARTiS consists in introducing asymmetry in an SMP platform to bring together High Performance Computing and Real-Time properties. In order to validate the principles an implementation based on Linux had been developed in 2004. This year, after finalising the implementation of an enhanced load-balancing mechanism taking into account the asymmetry, the validation phase was carried on. In particular, we measured the interrupt latency, the execution time jitter and the load-balancing efficienty. Those measurements required the development of new tools, which are now available on the project website [51], as the implementation itself. This work led to a publication describing the system in its globality [37] and to a second publication detailling the principles of the new load-balancing mechanism [42].

6.3.2. Early supports of FPGA

Gaspard is being extend to target FPGA. In this context, the first definition of the application and architecture metamodels must be update. A real-life application has been chosen to evaluate the default of the current Gaspard metamodel with respect to FPGA targeting. In the frame of the ModEasy project, we have implemented a new anti-collision radar algorithm on FPGA that should permit the detection of an obstacle up to 100 meters far away. The detection is based on higher order statics and requires much more hardware resources than state-of-the-art algorithms based on correlation. The implementation work has been done in cooperation with the University of Montreal, Canada on a Stratix pro FPGA. Several manual optimization were necessary to constrain the application to the FPGA board and we are now working on the identification of high level Gaspard constructs to synthesize the informations need to automate such an implementation.

We have designed a methodology to generate a FPGA configuration from a Gaspard repetition of an elementary component. The methodology determines the number of elementary component a given FPGA may host while supplying I/O connection to each instance of the component. The principle is to adapt the iteration factor of the loop to the FPGA constrains: unrolling the loop allows to fully exploit the computation power of the FPGA while the number of input and output pins of the FPGA limits this unrolling. Our proposition results in a formalization of the FPGA resource usage based on the unrolling factor, in an implementation pattern of the iterations that are multiplexed to fit on the hardware resources.

6.3.3. Transaction Level Modelling

Transaction level modelling (TLM) became a key issue in modern systems design. In fact, it is well known that it decreases significantly the simulation time while keeping an acceptable accuracy. This modelling style

has been standardized recently[3], thus we choose to integrate it in Gaspard as an intermediate model before physical implementations.

We developed a TLM metamodel allowing SystemC code generation. It contains all necessary concepts for ISP-specific SoC design. It integrates also information about several performances estimation criteria. Thus, after the simulation, we can easily check whether the system meets timing or energy consumption constraints for example [18].

We worked on a set of interoperability tools permitting communication between IPs described using different languages and having different protocols and interfaces. Thus, we are able to generate automatically, using the MDA approach, protocol adaptors for TLM IPs, in order to make them OCP compliant. Therefore, we can make all our libraries compliant to the standard. In this way automatic IP assembling becomes more and more systematic and easy [34], [39].

We worked on scripting inside SystemC, using Perl. It is a new concept in SystemC modelling and simulation. Our approach enables designers to add/remove IPs from a system during the run-time without stopping the simulation. This methodology can be very beneficial especially for architectures exploration and interconnects design [47]. It can be also combined with our distributed simulation tool, to be able to use several hosts for a single system simulation [40].

We have also worked on an operating system (OS) simulation for system on chips, in order to make a hardware/software co-simulation with a high hardware abstraction. It was developed for a hardware at a Transaction Level Modeling (TLM), but it can be adapted to a Register Transfer Level (RTL). An Embedded Linux solution has been studied and developed [45] using Virtual OS concepts.

We are developing a multistage omega network for SoC design. It is a micro-network that is a generic, scalable and multi-stage interconnect architecture for systems on chip (SoC). The network architecture relies on packet switching and point-to-point bi-directional links between the routers implementing the micro-network.

The NoC provides a configurable number of OCP compliant communication interfaces for both initiators (masters) and targets (slaves) [39].

7. Contracts and Grants with Industry

7.1. The PROTES Project: A Carroll Project

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

This project concerns the effort of standardisation of a UML profile for embedded and real time systems. This effort is associated to the P2I effort and integrates other techniques like the Accord UML profile developed by CEA. A goal of this project is to initiate a request for proposal by the OMG and then to answer to this request with common ideas.

In this project, three INRIA teams are involved. All of them are concerned with synchronous data-flow/control-flow models. This opportunity to develop together a UML profile for embedded and real-time systems and to support this proposal to OMG strengthens internal collaborations between DaRT, AOSTE and EXPRESSO.

7.2. Collaboration with Prosilog

Partners: Prosilog SA, DaRT

Prosilog SA, one of the leading provider of innovative solutions for SoC design and verification, announces the availability of its complete family of Compilers from SystemC to VHDL/Verilog and from VHDL/Verilog to SystemC as well as the first versions of adapters for the OCP transaction level communication channels.

[3] http://www.systemc.org

This year we have started a point to point collaboration with Prosilog around an optimized SoC simulation framework for a distributed and heterogeneous environment. This work is done together with a PhD student (CIFRE convention). Results of this research could be integrated in the Prosilog SystemC Compiler.

7.3. Collaboration with CEA List

Partners: CEA List, DaRT

This year we have started a point to point collaboration with CEA around an a UML profile for co-design. This work is done together with a PhD student (CEA funding). Results of this research could be integrated in the Gaspard tools at INRIA and in the AccordUML environment at CEA.

7.4. Collaboration with Valeo - CNRT Futurelec

Partners: Valeo, DaRT, L2EP

This year we have started a three partners'collaboration with Valeo and L2EP to fit Gaspard for high performance computing and meta-computing. This work is done with a PhD student (Valeo/region funding). Results of this research could be integrated in the Gaspard tools at INRIA and will be tested on finite element code for electric alternator simulator developed by Valeo and L2EP.

7.5. SoCLib RNRT Platform Project

Partners: CEA, CNRS, Thales Communications, ST Microelectronics, Prosilog, TurboConcept.

This project consists to develop an integration platform for a fast and secure SoC Design from IPs. Models of hardware components have to be interoperable, validated and available at different levels of abstraction

The DaRT team participates to this effort via the CNRS SoCLib "equipe-projet". Our contribution concerns the optimisation of the SystemC runtime. We propose adapters for interoperability.

8. Other Grants and Activities

8.1. ModEsay Interreg III A Franco/English Cooperation

The ModEasy project[4] will develop software tools and techniques to aid in the development of reliable microprocessor based electronic (embedded) systems using advanced development and verification systems.

The tools will be evaluated in practical domains, e.g. the automotive sector for reactive cruise control and anti-collision radar but will be applicable for generic embedded systems in any safety and mission critical applications in the wider industrial domain. The project will succeed in reducing development and production costs while maintaining existing high dependability and safety levels as embedded systems become more complex for many existing and new products across the Euro-region.

The design process of embedded systems moves from abstract high level descriptions (Specification models) such as structure and behaviour diagrams, to low level specific implementations details expressed by microchip circuit diagram (Synthesis models). The goal of the project is to create a bridge between the high level abstract description (specification co-design systems) and the low level implementation details (synthesis co-design) on various hardware platforms. The objective is to produce integrated software tools for the development and verification of embedded systems in a number of areas in industry.

We have two partners in this project. The University of Kent has achieved recognition for its work on formal system verification, embedded system development support and hardware integration from research councils and industry. In particular the Embedded Systems research group within the Department of Electronics is a well-established group producing techniques in the areas of embedded systems development support and high performance computer systems architectures. IEMN (Institut Electronique Microelectronique Nanotechnologies) has substantial expertise in the safety of land-based transportation systems especially for collision

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

avoidance. One team of the IEMN-DOAE called (RDTS) is involved in Telecommunications, Signal processing and applications to transportation systems. RDTS research themes concerns especially location systems in transportation, collision avoidance systems, driver alarm and information systems.

8.2. International initiatives

8.2.1. 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.

8.2.2. Center of Embedded Computer Systems, University of California

SpecC is a system-level design language (SLDL) and a system-level design methodology developed by Daniel Gajski. In august during a six-week visit of Samy Meftali to CECS, we have developed together a first test of integration of SystemC and SpecC systems. From these very promising results, we have decided to establish a full collaboration between DaRT and CECS. This one covers the interoperability of the two systems and with Isaac Scherson it covers the IP definition in SpecC and SystemC of alignment network hardware components for shared memory multi processors.

8.2.3. Université de Montreal

A collaboration with University of Montreal laboratory started this year. The interrested laboratory is the LASSO (Laboratoire d'Analyse et de Synthèse des Systèmes Ordinés [5], from DIRO departement (Département d'Informatique et de Recherche Opérationnelle [6]. Prof. M. El Mostapha Aboulhamid came in the team in June, for one month, in order to start exchange. Moreover, Sébastien Le Beux, PhD student from DART project, went to LASSO for two months, supported by LIFL and CNRS.

8.2.4. ENS Tunis

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. Several student exchanges are already programmed for 2006.

8.3. National initiatives

8.3.1. CNRS initiatives

We are members of the "iHPerf" theme of the *Groupement de Recherche Architectures, Réseaux, Parallélisme* and of the two *Réseaux Thématiques Pluridisciplinaires* SoC and *architecture des machines et compilation* of the CNRS.

9. Dissemination

9.1. Scientific Community

We have participated in the following conferences, in addition to those mentioned in the publications:

- 12th Synchronous Workshop, Malta, november 21–25, 2005 [7]

[5] http://www.iro.umontreal.ca/~lablasso/lablasso
[6] http://www.iro.umontreal.ca/
[7] http://www.cs.um.edu.mt/~synchrone05

- HiPEAC 2005, International Conference on High Performance Embedded Architectures and Compilers, Barcelona, SPAIN, 17-18 November 2005 [8]
- CODES 2005, ACM Conference on Embedded Software ISSS 2005, IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, New-York Metropolitain area, NJ, 17-21 September 2005 [9]
- ARCHI05 (Architectures des systèmes matériels enfouis et méthodes de conception associées [10], a thematic school from CNRS. This third edition took place in Autrans (Vercors), from 21 to 25 March 2005, and courses mainly targeted computer architecture, embedded systems, GPP, DSP, FPGA...

We organized a team seminar in january in Marrakech.

Pierre Boulet was in the steering committee and the program committee of FDL'05 [11]. He is a member of the executive committee of ECSI. He is the editor of the book Advances in Design and Specification Languages for SoCs, selected contributions from FDL'04 [13]. He runs the UML for SoC design mailing list (umlsoc@univ-lille1.fr) with 132 subscribers from all over the world. He is a member of the INRIA evaluation committee since September 2005. He was in some PhD thesis committees.

Jean-Luc dekeyser was in programme committees for ISPAN05 and GTTSE05. In 2005, he was invinted as speaker at Newcas (Montreal), Recosoc(Montpellier). He is project learder of the ModEasy interreg. He was in about tens PhD and HdR thesis committees. As director of the Ph.D. program at LIFL, he was involved in the SFERE exchange project concerning Pakistan. He was visiting University of California Irvine in june and in december, and visiting Université de Montréal in June.

Cedric Dumoulin was in the program committee of IDM 05 (French workshop on Model Driven Engineering) held in Paris[12]. He is organization chair of IDM'06 which will be held in Lille. He has participated to panels on Open Source Software at LMO 05 (Bern) and BEA World Event 2005 (Paris).

Philippe Marquet and Éric Piel did the local organisation of RTLWS'05 [13], an international workshop on Real-Time Linux which was held in November 2005 in Lille. Éric Piel and Ashish Meena participated in the organisation of ISPDC'05[14], a conference concerning parallel and distributed computing which took place in July 2005 in Lille.

Sebastien Lebeux has visited the Université of Montréal for two months.

Samy Meftali was in the program committee for DATE'05[15].

Eric Rutten co-chaired the Francophone Conference on Modelling of Reactive Systems, MSR '05, (5–7 october 2005, Autrans (Grenoble), France) [12] [16]. He is co-editor of a special issue of the *j*ournal of Discrete Event Dynamical Systems on the modelling of reactive systems, to appear at the end of 2006. He was in programme committees for *R*TS Embedded Systems 05 (5–6 april 2005, Paris, France) [17]. He was in PhD thesis committees as a reporter for Marius-Petru Stănicã (Control Theory, University of Rennes 1, Supelec, may 2005) and Frédéric Py (Computer Science, University of Toulouse, LAAS, october 2005). He was a member of the INRIA Evalualuation Board until sept. 2005. He is a member of the recruiting commission of University of Western Brittany (Université de Bretagne Occidentale (UBO), Brest). He made an invited seminar at IRISA/INRIA-Rennes [18].

[8] http://www.hipeac.net/hipeac2005/
[9] http://www.codes-isss.org
[10] http://www.ens-lyon.fr/ARCHI05/
[11] http://www.ecsi-association.org/ecsi/fdl/fdl05/default_home.htm
[12] http://www.planetmde.org/idm05
[13] http://realtimelinuxfoundation.org/events/rtlws-2005/ws.html
[14] http://www.lifl.fr/ispdc2005
[15] http://www.date-conference.com
[16] http://www.lag.ensieg.inpg.fr/msr05
[17] http://rts05.loria.fr
[18] http://www.irisa.fr/NQRT

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, Eric Rutten) and “introduction to real-time operating systems” (Philippe Marquet).

The following internships were advised in the team:

- Julien Taillard, M2R Computer Science, Université des Sciences et Technologies de Lille,
- Ali Koudri, M2R Computer Science, Université des Sciences et Technologies de Lille,
- Ahmed Jerbi and Yousri Miled , Computer Science Engineering School, Tunisia.

10. Bibliography

Major publications by the team in recent years

[1] K. ALTISEN, A. CLODIC, F. MARANINCHI, E. RUTTEN. *Using Controller-Synthesis techniques to build Property-Enforcing Layers*, in "Proceedings of the European Symposium on Programming, ESOP’03, Warsaw, Poland", LNCS, April 2003.

[2] A. AMAR, P. BOULET, J.-L. DEKEYSER. *Towards Distributed Process Networks with CORBA*, in "Parallel and Distributed Computing Practice on Algorithms", Special Issue on Parallel and Distributed Computing Practice on Algorithms, 2003.

[3] A. AMAR, P. BOULET, J.-L. DEKEYSER, F. THEEUWEN. *Distributed Process Networks Using Half FIFO Queues in CORBA*, in "ParCo’2003, Dresden, Germany", Parallel Computing, September 2003.

[4] P. BOULET, A. CUCCURU, J.-L. DEKEYSER, C. DUMOULIN, P. MARQUET, M. SAMYN, R. DE SIMONE, G. SIEGEL, T. SAUNIER. *MDA for SoC Design: UML To SystemC Experiment*, in "USOC 2004 - International Workshop on UML for SoC Design (Sponsored by DAC 2004), San Diego, California", June 2004.

[5] P. BOULET, J.-L. DEKEYSER, C. DUMOULIN, P. MARQUET. *MDA for System-on-Chip Design, Intensive Signal Processing Experiment*, in "FDL’03, Fankfurt, Germany", September 2003.

[6] A. CUCCURU, P. BOULET, J.-L. DEKEYSER. *Regular Hardware Architecture Modeling with UML2*, in "FDL04, Lille, France", September 2004.

[7] 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.

[8] S. MEFTALI, J. VENNIN, J.-L. DEKEYSER. *A fast SystemC simulation Methodology fo Multi-Level IP/SoC Design*, in "IFIP International Workshop On IP Based System-on-Chip Design, Grenoble, France", November 2003.

[9] 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.

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

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

Books and Monographs

[12] H. ALLA, E. RUTTEN (editors). *Modélisation des systèmes réactifs*, RS série JESA, vol. 39, n° 1-2-3/2005, Hermès, Autrans (Grenoble), France, October 2005, http://jesa.revuesonline.com.

[13] P. BOULET (editor). *Advances in Design and Specification Languages for SoCs, Selected contributions from FDL'04*, ChDL, Springer, 2005, http://www.springeronline.com/alert/article?a=0_b_38b3_698vjb_6.

[14] OBJECT MANAGEMENT GROUP, INC. (editor). *UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) RFP*, February 2005, http://www.omg.org/cgi-bin/doc?realtime/2005-02-06.

Doctoral dissertations and Habilitation theses

[15] A. CUCCURU. *Modélisation unifiée des aspects répétitifs dans la conception conjointe logicielle / matérielle des systèmes sur puce hautes performances*, Doctorat en Informatique, Université des Sciences et Technologies de Lille, December 2005.

[16] P. DUMONT. *Spécification multidimensionnelle pour le traitement du signal systématique*, Doctorat en Informatique, Université des Sciences et Technologies de Lille, December 2005.

[17] SMAÏL. NIAR. *Contribution à la conception de nouvelles micro-architectures*, Habilitation à Diriger des Recherches, Informatique, Université de Valenciennes et du Hainaut Cambrésis, November 2005.

[18] M. SAMYN. *Une simulation fonctionnelle dun système monopuce dédié au traitement du signal intensif : Une approche dirigée par les modèles*, Doctorat en Informatique, Université des Sciences et Technologies de Lille, December 2005.

Articles in refereed journals and book chapters

[19] L. BONDÉ, P. BOULET, A. CUCCURU, J.-L. DEKEYSER, C. DUMOULIN, P. MARQUET. *Model Driven Engineering for Distributed Real-time Embedded Systems*, S. GÉRARD, J.-P. BABAU, J. CHAMPEAU (editors). , chap. 12: Model-Driven Architecture for Intensive Embedded Systems, ISTE, 2005.

[20] L. BONDÉ, C. DUMOULIN, J.-L. DEKEYSER. *Advances in Design and Specification Languages for SoCs*, P. BOULET (editor). , chap. 6 - Metamodels and MDA Transformations for Embedded Systems, Kluwer Academic, 2005.

[21] L. CHAREST, P. MARQUET, J.-L. DEKEYSER, EL MOSTAPHA. ABOULHAMID, G. BOIS. *Using design pattern for type unification, structural unification, semantic clarification and introspection in SystemC*, in "Annals for Micro and Nano Systems", to appear, 2006.

[22] A. CUCCURU, P. MARQUET, J.-L. DEKEYSER. *IFIP International Federation for Information Processing*, vol. 176, chap. UML 2 as an ADL: Hierarchichal Hardware Modeling, Springer, 2005, p. 133-147.

[23] S. MEFTALI, J. VENNIN, J.-L. DEKEYSER. *An optimized distributed simulation environment for SoC design*, in "Annals for Micro and Nano Systems", to appear, 2006.

Publications in Conferences and Workshops

[24] 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/.

[25] R. B. ATITALLAH, S. NIAR, A. GREINER, S. MEFTALI, J. L. DEKEYSER. *Estimating Energy Consumption for an MPSoC Architectural Exploration*, in "ARCS'06: Architecture of Computing Systems, Frankfurt, Germany", to appear, March 2006, http://arcs06.cs.uni-frankfurt.de/.

[26] L. BONDE, P. BOULET, J.-L. DEKEYSER. *Traceability and interoperability at different levels of abstraction in model transformations*, in "FDL'05, Lausanne, Switzerland", September 2005, http://www.ecsi-association.org/ecsi/fdl/fdl05/default_home.htm.

[27] P. BOULET, A. CUCCURU, J.-L. DEKEYSER, A. MEENA. *Model Driven Engineering for Regular MPSoC Co-design*, in "ReCoSoC-05, Montpellier, France", June 2005, http://www.lirmm.fr/RECOSOC05/.

[28] P. BOULET, A. MEENA. *The case for Globally Irregular Locally Regular Algorithm Architecture Adequation*, in "Journées Francophones sur l'Adéquation Algorithme Architecture (JFAAA'05), Dijon, France", January 2005, http://cerbere.u-bourgogne.fr/jfaaa05.

[29] L. CHAREST, P. MARQUET. *Comparisons of different approaches of realizing IP block configuration in SystemC*, in "3rd International IEEE Northeast Workshop on Circuits and Systems, Québec City, Canada", June 2005, http://www.newcas.org.

[30] 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.

[31] A. CUCCURU, R. D. SIMONE, T. SAUNIER, G. SIEGEL, Y. SOREL. *P2I : An innovative MDA methodology for embedded real-time systems*, in "DSD'2005, 8th Euromicro conference on digital system design, Porto, Portugal", September 2005.

[32] J.-L. DEKEYSER, P. MARQUET, S. MEFTALI, C. DUMOULIN, P. BOULET, S. NIAR. *Why to do without Model Driven Architecture in embedded system codesign?*, in "first annual IEEE BENELUX/DSP Valley Signal Processing Symposium, SPS-DARTS 2005, Antwerp, Belgium", April 2005, http://acivs.org/sps2005.

[33] G. DELAVAL, E. RUTTEN. *A domain-specific language for task handlers generation, applying discrete controller synthesis*, in "Proceedings of the 21st ACM Symposium on Applied Computing, SAC'06, Special track on Embedded Systems: Applications, Solutions, and Techniques, Dijon, France", to appear, April 2006.

[34] A. KOUDRI, S. MEFTALI, J.-L. DEKEYSER. *IP integration in embedded systems modeling*, in "IP-SOC 2005, IP Based SoC Design Conference, Grenoble, France", December 2005, http://www.us.design-reuse.com/ipsoc2005/.

[35] O. LABBANI, J.-L. DEKEYSER, P. BOULET. *Mode-Automata based Methodology for Scade*, in "Hybrid Systems: Computation and Control, 8th International Workshop, Zurich, Switzerland", LNCS series, Springer, March 2005, p. 386-401.

[36] O. LABBANI, J.-L. DEKEYSER, P. BOULET, E. RUTTEN. *Introducing Control in the Gaspard2 Data-Parrallel MetaModel: Synchronous Approach*, in "MARTES '05, International Workshop on Modelling and Analysis of Real-Time and Embedded Systems, (in conjunction with 8th International Conference on Model Driven Engineering Languages and Systems, MoDELS/UML 2005), Montego bay, Jamaica", October 2005, http://www.martes.org.

[37] P. MARQUET, É. PIEL, J. SOULA, J.-L. DEKEYSER. *ARTiS, un système d'exploitation temps-réel asymétrique*, in "CFSE'4 : 4e édition de la Conférence Française sur les Systèmes d'Exploitation, Le Croisic, France", (in french), April 2005, http://www.renpar.org.

[38] 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/.

[39] 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/.

[40] S. MEFTALI, A. DZIRI, L. CHAREST, P. MARQUET, J.-L. DEKEYSER. *Soap based distributed simulation environment for system-on-chip (soc) design*, in "FDL'05, Lausanne, Switzerland", September 2005, http://www.ecsi-association.org/ecsi/fdl/fdl05/default_home.htm.

[41] S. NIAR, S. MEFTALI, J. L. DEKEYSER. *Power aware cache memory design with SystemC*, in "18th IEEE International Conference on MicroElectronics, Tunis, Tunisia", December 2004, http://www.gmslab.org/conferences/icm2004/.

[42] E. PIEL, P. MARQUET, J. SOULA, J.-L. DEKEYSER. *Asymmetric Scheduling and Load Balancing for Real-Time on Linux SMP*, in "Workshop on Scheduling for Parallel Computing, SPC 2005, Poznan, Poland", Springer-Verlag, Lecture Notes in Computer Science, September 2005, http://www.cs.put.poznan.pl/mdrozdowski/spc-ppam05/.

[43] L. RIOUX, T. SAUNIER, S. GERARD, A. RADERMACHER, R. DE SIMONE, T. GAUTIER, Y. SOREL, J. FORGET, J.-L. DEKEYSER, A. CUCCURU, C. DUMOULIN, C. ANDRE. *MARTE: A New Profile RFP for the Modeling and Analysis of Real-Time Embedded Systems*, in "UML-SoC'05, DAC 2005 Workshop UML for SoC Design, Anaheim, CA", June 2005.

[44] J. TAILLARD, J.-L. DEKEYSER, F. PIRIOU. *A Design of a UML Profile for Meta-computing*, in "PDP 2006, 14th Euromicro Conference on Parallel, Distributed and Network-based Processing, Montbéliard-Sochaux, France", to appear, February 2006, http://www.pdp2006.org.

[45] J. TAILLARD, P. MARQUET, J.-L. DEKEYSER. *Embedded Linux Co-simulation*, in "IP-SOC 2005, IP Based SoC Design Conference, Grenoble, France", December 2005, http://www.us.design-reuse.com/ipsoc2005/.

[46] J. VENNIN, S. MEFTALI, J.-L. DEKEYSER. *Understanding and extending SystemC user thread package to IA-64 platform*, in "International Workshop on IP Based SoC design, Grenoble, France", December 2004, http://www.us.design-reuse.com/ipsoc2004.

[47] J. VENNIN, S. PENAIN, L. CHAREST, S. MEFTALI, J.-L. DEKEYSER. *Embedded scripting inside systemc*, in "FDL'05, Lausanne, Switzerland", September 2005, http://www.ecsi-association.org/ecsi/fdl/fdl05/default_home.htm.

Internal Reports

[48] A. AMAR, P. BOULET, P. DUMONT. *Projection of the Array-OL Specification Language onto the Kahn Process Network Computation Model*, Research Report, n° RR-5515, INRIA, March 2005, http://www.inria.fr/rrrt/rr-5515.html.

[49] G. DELAVAL, E. RUTTEN. *A domain-specific language for multi-task systems, applying discrete controller synthesis*, Research Report, n° 5690, INRIA, sep 2005, http://www.inria.fr/rrrt/rr-5690.html.

[50] P. DUMONT, P. BOULET. *Another Multidimensional Synchronous Dataflow: Simulating Array-OL in Ptolemy II*, Research Report, n° RR-5516, INRIA, March 2005, http://www.inria.fr/rrrt/rr-5516.html.

Miscellaneous

[51] LIFL, USTL. *ARTiS Home Page*, http://www.lifl.fr/west/artis/.

Bibliography in notes

[52] J. MILLER, J. MUKERJI (editors). *MDA Guide (Draft Version 0.2)*, 2003, http://www.omg.org/docs/ab/03-01-03.pdf.

[53] OBJECT MANAGEMENT GROUP, INC. (editor). *MOF 2.0 Core Final Adopted Specification*, 2003, http://www.omg.org/cgi-bin/doc?ptc/03-10-04.

[54] OBJECT MANAGEMENT GROUP, INC. (editor). *(UML) Profile for Schedulability, Performance, and Time Version 1.1*, http://www.omg.org/technology/documents/formal/schedulability.htm.

[55] 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.

[56] 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.

[57] OBJECT MANAGEMENT GROUP, INC. (editor). *UML 2 Superstructure (Available Specification)*, October 2004, http://www.omg.org/cgi-bin/doc?ptc/2004-10-02.

[58] OBJECT MANAGEMENT GROUP, INC. (editor). *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms*, September 2004, http://www.omg.org/cgi-bin/doc?ptc/2004-09-01.

[59] OBJECT MANAGEMENT GROUP, INC. (editor). *SysML v0.9*, January 2005, http://www.omg.org/cgi-bin/doc?ad/05-01-03.

[60] OBJECT MANAGEMENT GROUP, INC. (editor). *UML Extension Profile for SoC RFC*, March 2005, http://www.omg.org/cgi-bin/doc?realtime/2005-03-01.

[61] M. D. ADAMS. *The JPEG-2000 Still Image Compression Standard*, Technical report, nº N2412, ISO/IEC JTC 1/SC 29/WG 1, September 2001.

[62] R. ALLEN, K. KENNEDY. *Optimizing Compilers for Modern Architectures: A Dependence-based Approach*, Morgan Kaufmann Publishers, October 2001, http://books.elsevier.com/us//mk/us/subindex.asp?maintarget=&isbn=1-55860-286-0&country=United+States&srccode=&ref=&subcode=&head=&pdf=&basiccode=&txtSearch=&SearchField=&oper

[63] K. ARNOLD, J. GOSLING, D. HOLMES. *The Java Programming Language*, 3rd, Addison-Wesley, 2000.

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

[65] OMG. A. BOARD. *Model Driven Architecture (MDA)*, Technical report, nº ormsc/2001-07-01, OMG, 2001.

[66] P. BOULET, A. DARTE, G.-A. SILBER, F. VIVIEN. *Loop parallelization algorithms: From parallelism extraction to code generation*, in "Parallel Computing", vol. 24, nº 3-4, May 1998, p. 421–444.

[67] 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.

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

[69] A. DARTE, C. DIDERICH, M. GENGLER, F. VIVIEN. *Scheduling the Computations of a Loop Nest with Respect to a Given Mapping*, in "Lecture Notes in Computer Science", vol. 1900, 2001, http://link.springer-ny.com/link/service/series/0558/bibs/1900/19000405.htm; http://link.springer-ny.com/link/service/series/0558/papers/1900/19000405.pdf.

[70] A. DARTE, Y. ROBERT, F. VIVIEN. *Scheduling and Automatic Parallelization*, Birkhauser Boston, 2000, http://www.springeronline.com/sgw/cda/frontpage/0,11855,5-40109-22-2026983-0,00.html.

[71] 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.

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

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

[74] P. DUMONT. *Étude des Transformations d'un Code Array-OL dans Gaspard*, Research Report, n° 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.

[75] C. DUMOULIN. *ModTransf: A Model to Model Transformation Engine*, December 2003, http://www.lifl.fr/west/modtransf.

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

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

[78] T. GARDNER, C. GRIFFIN, J. KOEHLER, R. HAUSER. *A Review of OMG MOF 2.0 Query / Views / Transformations Submissions*, OMG paper, July 2003, http://www.omg.org/docs/ad/03-08-02.pdf.

[79] T. GRANDPIERRE, C. LAVARENNE, Y. SOREL. *Optimized Rapid Prototyping for Real-Time Embedded Heterogeneous Multiprocessors*, in "Proceedings of the 7th International Workshop on Hardware/Software Codesign (CODES99), New York", ACM Press, May 3–5 1999, p. 74–78.

[80] M. GUPTA, S. MUKHOPADHYAY, N. SINHA. *Automatic Parallelization of Recursive Procedures*, in "International Journal of Parallel Programming", vol. 28, n° 6, 2000, p. 537-562, http://citeseer.ifi.unizh.ch/gupta99automatic.html.

[81] N. HALBWACHS, J.-C. FERNANDEZ, A. BOUAJJANNI. *An executable temporal logic to express safety properties and its connection with the language Lustre*, in "Sixth International Symp. on Lucid and Intensional Programming, ISLIP'93, Quebec", April 1993.

[82] ITRS. *Design, 2001 edition*, 2001, http://public.itrs.net/.

[83] 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.

[84] 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.

[85] V. LEFEBVRE, P. FEAUTRIER. *Optimizing Storage Size for Static Control Programs in Automatic Parallelizers*, in "European Conference on Parallel Processing", 1997, p. 356-363, http://citeseer.ifi.unizh.ch/lefebvre97optimizing.html.

[86] A. W. LIM. *Improving Parallelism and Data Locality with Affine Partitioning*, Ph. D. Thesis, Stanford University, September 2001.

[87] D. E. MAYDAN, S. P. AMARASINGHE, M. S. LAM. *Array-data flow analysis and its use in array privatization*, in "Conference record of the Twentieth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages: papers presented at the symposium, Charleston, South Carolina, January 10–13, 1993, New York, NY, USA", ACM (editor). , ACM Press, 1993, p. 2–15, http://www.acm.org:80/pubs/citations/proceedings/plan/158511/p2-maydan/.

[88] M. MOTTL. *Automating Functional Program Transformation*, Technical report, University of Edinburgh, September 2000, http://www.ai.univie.ac.at/~markus/msc_thesis/.

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

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

[91] OMG. *MOF 2.0 Query / Views / Transformations RFP*, OMG paper, 2003.

[92] OBJECT MANAGEMENT GROUP, INC.. *MOF Meta Object Facility, Specification, Version 1.3*, January 2000, http://www.omg.org/cgi-bin/doc?formal/00-04-03.

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

[94] C. PAREJA, R. PEÑA, F. RUBIO, C. SEGURA. *Optimizing Eden by program transformation*, in "2nd Scottish Functional Programming Workshop, St. Andrews 2000", Intellect, 2001, http://www.mathematik.uni-marburg.de/~eden/paper/ParejaPenaRubioSeguraSFP2000.ps.

[95] H. J. REEKIE. *Realtime Signal Processing: Dataflow, Visual, and Functional Programming*, PhD Thesis, School of Electrical Engineering, University of Technology, Sydney, Australia, September 1995, http://ptolemy.eecs.berkeley.edu/~johnr/papers/thesis.html.

[96] M. SERRANO, P. WEIS. *Bigloo: A Portable and Optimizing Compiler for Strict Functional Languages*, in "Static Analysis Symposium", 1995, p. 366-381, http://citeseer.ifi.unizh.ch/serrano95bigloo.html.

[97] Y. SOREL, C. LAVARENNE. *SynDEx Documentation Index*, INRIA, 2000, http://www-rocq.inria.fr/syndex/doc/.

[98] Y. SOREL, C. LAVARENNE. *Modèle unifié pour la conception conjointe logiciel-matériel*, in "Traitement du Signal (numéro spécial Adéquation Algorithme Architecture)", vol. 14, n° 6, 1997, p. 569-578, http://www-rocq.inria.fr/syndex/pub.htm.

[99] 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.

[100] 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.

[101] P. TU, D. PADUA. *Chapter 8. Automatic Array Privatization*, in "Lecture Notes in Computer Science", vol. 1808, 2001, http://link.springer-ny.com/link/service/series/0558/bibs/1808/18080247.htm; http://link.springer-ny.com/link/service/series/0558/papers/1808/18080247.pdf.