

*Team DaRT*

*Dataparallelism for Real-Time*

*Futurs*

THEME 1C

*Activity*  
*R* *eport*

2003



# Table of contents

<b>1. Team</b>	<b>1</b>
<b>2. Overall Objectives</b>	<b>1</b>
<b>3. Scientific Foundations</b>	<b>2</b>
3.1. Introduction	2
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	4
3.2.5.1. Metamodels for the “Y” Design	4
3.2.5.2. PIMs and PSMs	4
3.2.5.3. Application Metamodel	5
3.2.5.4. Hardware Architecture and Association Metamodels	6
3.2.5.5. PSM Metamodels	8
3.2.5.6. Transformation Techniques	8
3.3. Optimization Techniques	8
3.3.1. Contributions	9
3.3.1.1. Dataparallel Code Transformations	9
3.3.1.2. Multi-objective Hierarchical Scheduling Heuristics	9
3.4. SoC Simulation	10
3.4.1. Abstraction levels	10
3.4.2. Contribution	10
3.4.2.1. Distributed Kahn Process Network	11
3.4.2.2. Co-simulation in SystemC	11
3.4.2.3. Multilevel distributed simulation in SystemC	12
<b>4. Application Domains</b>	<b>12</b>
4.1. Intensive Signal Processing	12
4.1.1. Software Radio Receiver	12
4.1.2. Sonar Beam Forming	12
4.1.3. JPEG-2000 Encoder/Decoder	12
<b>5. Software</b>	<b>13</b>
5.1. MDA Transf	13
5.2. Gaspard v2.0	13
<b>7. Contracts and Grants with Industry</b>	<b>15</b>
7.1. Sophocles itea Project	15
7.1.1. Partners:	15
7.2. Prompt2Implementation itea Project	17
7.2.1. Partners:	17
7.3. The PROTES Project: A Carroll Project	18
7.3.1. Partners:	18
7.4. Collaboration with Prosilog	18
7.4.1. Partners:	18
7.5. SoCLib RNRT Platform Project	18
7.5.1. Partners:	18
7.6. ECSI member	18
<b>8. Other Grants and Activities</b>	<b>19</b>

8.1. European initiatives	19
8.1.1. Eurosoc ex NoE Project	19
8.2. International initiatives	19
8.2.1. Partnership with the Center of Embedded Computer Systems, University of California	19
8.3. National initiatives	19
8.3.1. CNRS “Action Spécifique compilation pour l’embarqué”	19
8.3.2. Other CNRS initiatives	19
8.4. Enseignement	19
<b>10. Bibliography</b>	<b>19</b>

# 1. Team

## Head of project-team

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

## Faculty member

Pierre Boulet [Professor, Université des Sciences et Technologies de Lille]

Philippe Marquet [Associate professor, Université des Sciences et Technologies de Lille]

## Research scientist

Cédric Dumoulin [ITEA project grant]

## Research scientist (partner)

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

## Post-doctoral fellow

Samy Meftali [Teaching Assistant, Université des Sciences et Technologies de Lille]

Akdelkader Amar [Teaching Assistant, Université des Sciences et Technologies de Lille]

## Ph. D. student

Lossan Bondé [ITEA project grant]

Arnaud Cuccuru [ITEA project grant]

Philippe Dumont [Teaching Assistant, Université des Sciences et Technologies de Lille]

Ouassila Labbani [CNRS and regional grant]

Ashish Meena [ITEA project grant]

Mickaël Samyn [French ministry grant]

Joël Vennin [Prosilog grant]

## Technical staff

Stépane Akhoun [ITEA project grant]

# 2. Overall Objectives

The 2001 International Technology Roadmap for Semiconductors [36] 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.

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 allow us to push our developments 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 [57], MDA [16], MOF [45] for the modeling and the automatic model transformations; Array-OL [26][27][22], synchronous languages (such as Esterel [20] or Lustre [35]), Kahn process networks [37] as computation models with strong semantics for verification; SystemC [59] for the simulations; and Java [19] to code our prototypes.

## 3. Scientific Foundations

### 3.1. Introduction

*Glossary*

**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 [53]). This also permits to produce automatically a mapping and a schedule of the application onto the architecture with code generation (as with the AAA method of SynDEx [50]). 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. We have developed a more general distributed environment to support the execution of Kahn Process Networks. This kind of simulation is at the functional level. 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

**Participants:** Lossan Bonde, Pierre Boulet, Arnaud Cuccuru, Jean-Luc Dekeyser, Cédric Dumoulin, Philippe Marquet, Ouassila Labbani.

**Key words:** *Modeling, UML, MDA, MDA Transformation, Model, Metamodel, MOF.*

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 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 relieving the programmer of the usual constraints of code development. Portability of an application from one language to another (a new one for example) increases the workload of the programmer. This drawback is also true for the development of embedded applications. It is even worse, because the number of abstraction levels has to be added to the diversity of the languages. It is essential to associate a target hardware architecture model to the application specification model, and to introduce 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 reformatize 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 [16][58] 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 [43].

### 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 [56] 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 [54]) 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

Our proposal is partially based upon the concepts of the “Y-chart” [30]. 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.

#### 3.2.5.1. 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.

#### 3.2.5.2. PIMs and PSMs

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.

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.



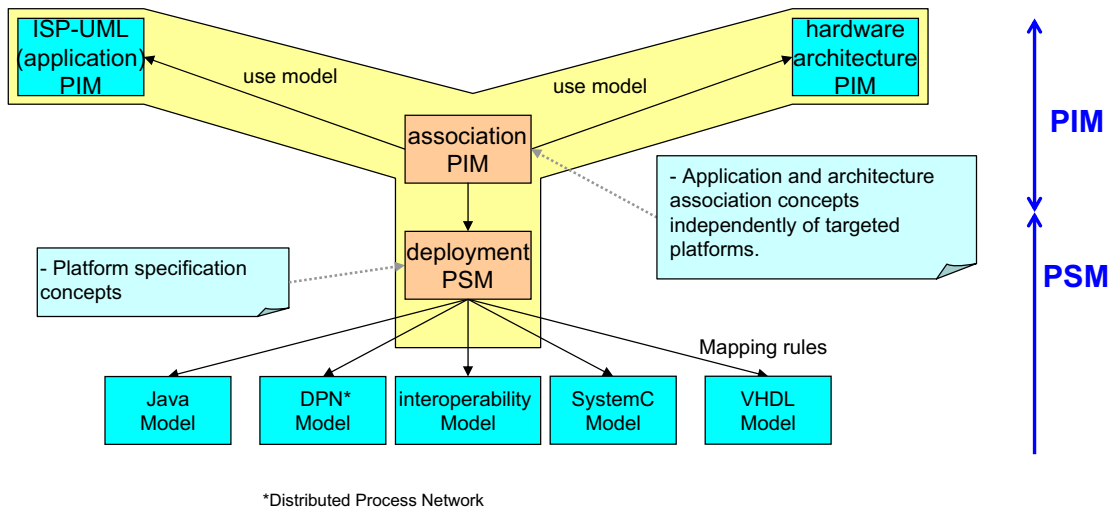


Figure 1. Overview of the metamodels for the “Y” design

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. Figure 2 proposes 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.

### 3.2.5.3. Application Metamodel

In our metamodels, we will use well defined semantics to be able to verify the models as soon as possible. We have started work with the following computation models: Kahn process networks, synchronous reactive programming (Esterel, Lustre) and Array-OL. We want to build a comprehensive (including control flow, data flow and data parallelism) application metamodel based on and integrating these three approaches. This is realistic because of the nature of these models which, in some way, share the synchronous hypothesis. We will deal with the time notion in two ways: implicitly as in synchronous languages and explicitly to express time constraints. Ouassila Labbani has started her Ph. D. thesis on this subject in september 2003.

The principles of this application metamodel are already partly defined in ISP UML [14]. ISP UML allows the expression of both task parallelism and data parallelism. A main characteristic of this metamodel is the single assignment form. Thus the time dimension is explicit in the data structures (arrays) and can be infinite.

In ISP UML, modeling is component based. The component represents some computation and its associated ports represent its input and output capabilities. Those components can be composed, data parallel or elementary.

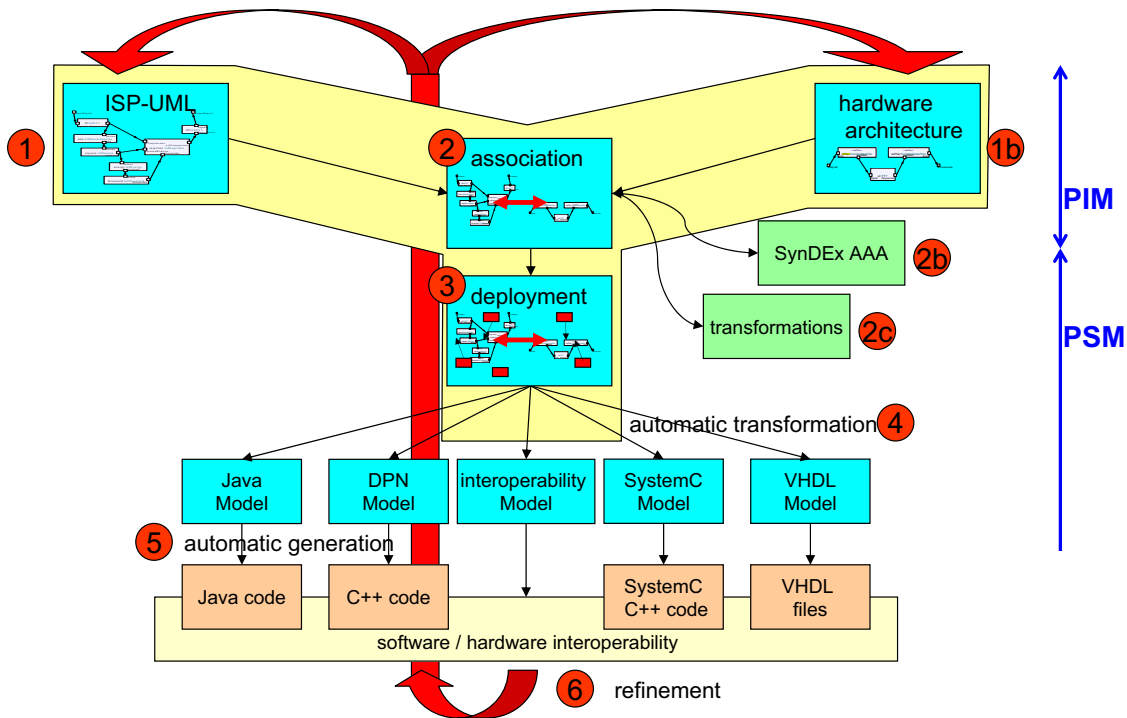


Figure 2. Overview of a possible methodology

- 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 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. See figure 3 for an example.
- An elementary component is the basic computation unit of the application and has to be defined for each target technology.

This hierarchical description allows to consider the application with different granularities. Indeed, the data dependences expressed at one level are approximations of the real data dependences described at the deepest level of the hierarchy.

In the first versions, ISP UML uses UML 2.0 structure diagrams to model components. We are now considering using activity diagrams instead. We will also improve it with ways to express constraints and characteristics (time, consumption, ...) and define a precise semantics.

#### 3.2.5.4. Hardware Architecture and Association Metamodels

Building a hardware model at the right granularity to be useful for compilation is in itself a difficult research subject. The model should be precise enough to be pertinent but not too detailed so that efficient decisions can be taken. The model should also characterize the architecture with respect to different domains (computation time, power consumption, ...).

The hardware component represents an abstraction of a physical hardware architecture element. An hardware component owns an interface materialized by its ports, and a structure defined by an assembly of components.

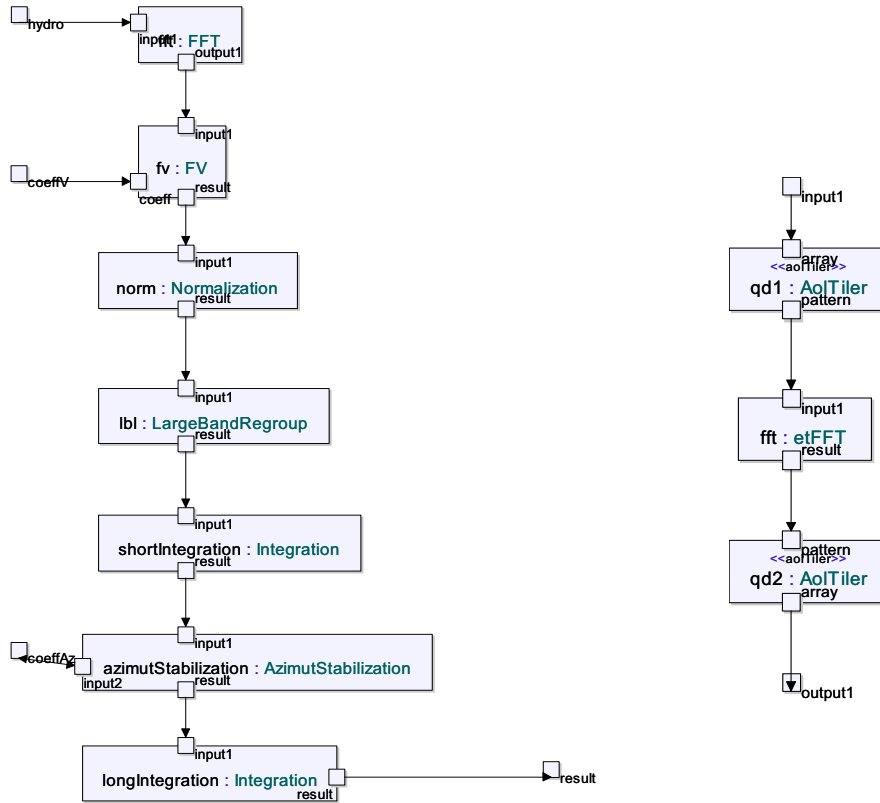


Figure 3. ISP UML compound component and data parallel component

An assembly of components is called a “compound component”. A compound component can represent an “executable architecture” (architecture fully defined) or a part of an architecture that will be reused in other contexts. A compound component can be defined with several hierarchy levels, let’s say with other compound components assembled together.

Components are assembled via connections defined between their ports. The ports enable to define a communication protocol. Some components can send service requests to their environment via their ports, while the others can only receive these requests via their ports and satisfy them. (Example : A processor which sends a read request to a RAM).

We distinguish three families of hardware components, expressed in UML with an appropriate stereotype : active components, passive components and interconnection components. For each family, we identify several basic components which can be used to define (by assembling) more complex components.

Following the same idea, the association metamodel should enable us to express application scheduling and mapping onto a hardware architecture model. There exists a strong link between the conception of the metamodels and the optimization techniques presented in the following section. Arnaud Cuccuru has been working towards his Ph. D. on this subject since september 2002.

#### 3.2.5.5. PSM Metamodels

We will focus here on a particular technology, SystemC, to be able to demonstrate a complete design flow in a few years. The metamodels appearing at the PSM level are not complete metamodels of the target language but rather metamodels providing the concepts needed to execute the mapped application. A last straightforward transformation stage will generate SystemC code from the PSM SystemC model. For more details, see the section on simulation techniques 3.4.2.2.

#### 3.2.5.6. Transformation Techniques

Though our research domain is not the model to model transformation techniques, we need some tool to realize our prototype. Thus we develop in a very pragmatic way a transformation tool for the MDA [29]. We do not aim at completeness but at a tool which enable us both to map a PIM model to a PSM model in a deterministic way and to generate code. A first rule based prototype has been written by Cédric Dumoulin and Lossan Bondé will pursue this work in his Ph. D. started in september 2003.

### 3.3. Optimization Techniques

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

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

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 fulfilling the requirements of the application, whether they be real time, memory usage or power consumption constraints. These techniques are thus multi-objective and target heterogeneous architectures.

We aim at taking advantage of the parallelism (both data-parallelism and task parallelism) expressed in the application models in order to build efficient heuristics. Our application model has some good properties that can be exploited by the compiler: it expresses all the potential parallelism of the application, it is an expression of data dependences –so no dependence analysis is needed–, it is in a single assignment form and unifies the temporal and spatial dimensions of the arrays. This gives to the optimizing compiler all the information it needs and in a readily usable form. Many optimization techniques have been studied that can be useful in our case. These techniques cover several fields of compiler construction:

- Automatic parallelization [21][40][25][18][24] with loop transformation, scheduling and mapping techniques.
- Memory management [39][41][52] to reuse the storage space while preserving parallelism.

- Pure functional language compilation [47][44][34][42] with techniques such as static typing, higher order functions, derecursion, partial evaluation, etc.
- Signal processing specific optimizations [46].

### 3.3.1. Contributions

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

#### 3.3.1.1. Dataparallel Code Transformations

In some of our previous works we have studied Array-OL to Array-OL code transformations [22][51][28]. Array-OL [26][27] 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 pursue the study of such transformations with three objectives:

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

This work is the subject of Philippe Dumont's Ph. D. Thesis.

#### 3.3.1.2. 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 [50][49][32] that is implemented in the SynDEx [48] 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. We call this approach globally irregular, locally regular. 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. We will thus study multi-objective techniques to build schedules that respect the real time constraints of the application while minimizing the resource usage.

Ashish Meena has just started a Ph. D. on this subject. Smaïl Niar, associate member of the project from the university of Valenciennes, is studying various techniques to reduce power consumption in embedded systems. This research covers:

- The evaluation of the impact of cache management schemas on power consumption [9][10].
- The study of code compression techniques to reduce the memory requirements of an embedded application [6].

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

### 3.4. SoC Simulation

**Participants:** Abdelkader Amar, Pierre Boulet, Jean-Luc Dekeyser, Samy Meftali, Smaïl Niar, Mickaël Samyn, Joël Vennin.

**Key words:** *SystemC, CORBA, Kahn Process Networks, 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 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 Transfer 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 four levels of abstraction (this is the case of SystemC [33]):

Untimed functional level (UTF): a model is similar to an executable specification, but no time delays are present at this level. Shared communication links (buses) are not modeled either. The communications between modules are point to point, and usually modeled using FIFOs.

Timed Functional Level (TF): it is similar to UTF 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.

Transaction Level (TLM): the communication between modules is modeled using function calls. At this level the communication model is accurate in term of functionality and often in term of timing (model the transaction on the buses but not the pins of the modules).

Register Transfer Level (RTL): it is the lowest level in a SystemC design flow. 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. Each module's behaviour corresponds exactly to the behaviour of the physical module.

#### 3.4.2. Contribution

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 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 4).

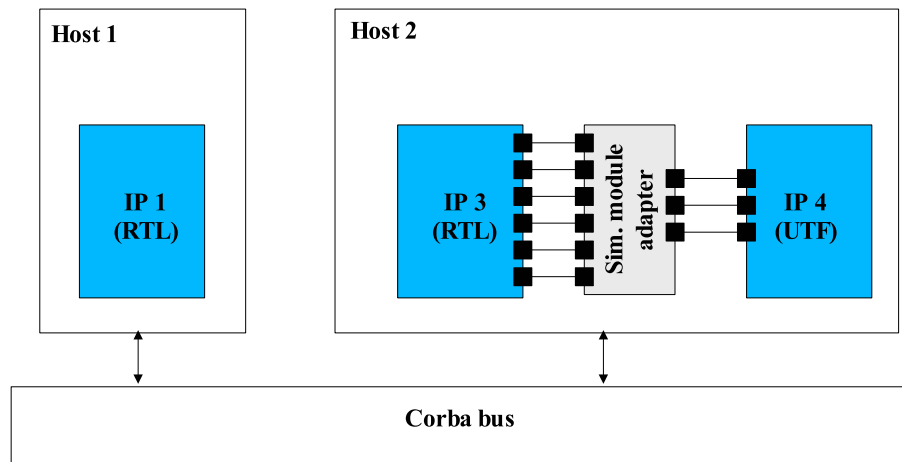


Figure 4. Distributed SystemC Simulation

#### 3.4.2.1. Distributed Kahn Process Network

Kahn Process Networks [37][38] are well suited to model many parallel applications, specially intensive signal processing applications or complex scientific applications (signal processing, image processing). We define a distributed execution model of a Kahn Process Networks that includes

- a distributed simulation relying on a component-based design providing an interactive deployment and hiding the communication layer;
- a dynamic distributed system allowing the evolution, during the application, of the application deployment and ensuring a load balance of the application;
- a support for multidimensional signal processing providing a data-flow execution for Array-OL applications.

The current runtime implementation [2] relies on heterogeneous distributed hardware connected by a Common Object Request Broker Architecture (CORBA) [55] middleware to handle the communications and to ensure the interoperability. It provides an efficient data transfer strategy, hybrid of the usual data-driven and demand-driven protocols. This work has led to the Ph. D. thesis of Abdelkader Amar [11] who is continuing to improve this area.

#### 3.4.2.2. 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.

Mickaël Samyn is developing a PSM metamodel to allow 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 development is integrated in the Gaspard prototype and uses the MDA Transf tool (see the software section).

### 3.4.2.3. 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 [7]. 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 [8]. 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. Joël Vennin has started a Ph. D. with Prosilog on this subject.

## 4. Application Domains

### 4.1. Intensive Signal Processing

**Key words:** *telecommunications, multimedia.*

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.

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.

#### 4.1.1. 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...).

#### 4.1.2. 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 Fourier 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.

#### 4.1.3. JPEG-2000 Encoder/Decoder

JPEG-2000 is a new standard format for image compression. The encoder works in a two-steps approach [17]. 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.

## 5. Software

### 5.1. MDA Transf

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

**Key words:** *Model Transformation, MDA, QVT, Query View Transformation.*

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

The MDA Transf 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 MDA Transf 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 developed 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 [31], and will follow the evolutions of this standard.

The tool is available as an open source distribution [29]. It is currently evaluated by other INRIA teams and external teams (CEA, academics).

### 5.2. Gaspard v2.0

**Participants:** Cédric Dumoulin [contact person], Stéphane Akhoun, Arnaud Cucuru, Mickaël Samyn, Lossan Bondé.

**Key words:** *Eclipse, IDE, SoC Design, Visual Design.*

Gaspard version 2.0 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.

Gaspard version 2.0 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)

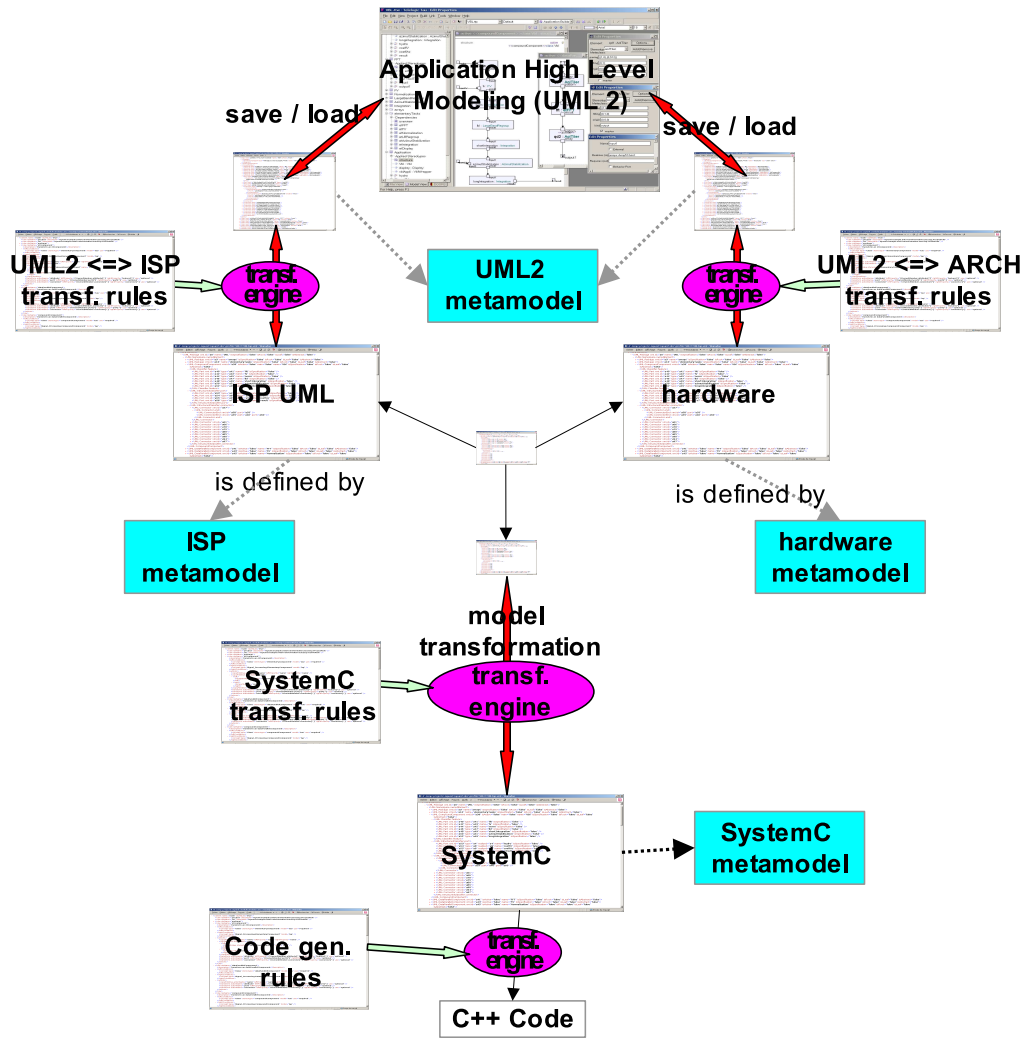


Figure 5. Overview of the Development Flow with Gaspard

- Code generation
- Simulation
- Reification of any stages of the development

The Gaspard version 2.0 tool is based on Eclipse [23]. 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 tool in the XMI file format. The tool manipulates these models through repositories (Java interfaces and implementations) automatically generated thanks to the JMI standard.

## 7. Contracts and Grants with Industry

### 7.1. Sophocles itea Project

Complex systems composed of heterogeneous components are notoriously difficult to design, integrate and validate. The aim of Sophocles is to validate the methodologies, platforms and technologies that make these operations possible over a distributed environment.

#### 7.1.1. Partners:

Thales Communication France, Thales Underwater System, Esterel Technologies, Philips, ENEA, LIFL, Ipitec.

The project was to define design methodologies and technologies for complex heterogeneous systems based on global system modeling and high level programming, and on integration of distributed VCs (Virtual Components). The design environment we set up is founded on the Cyber Enterprise model.

The project has required the development of techniques for the description of systems and the execution of multiple level executable models of Virtual Components. Such as:

- Distributed simulation techniques;
- Scheduling techniques for heterogeneous simulations;
- Co-simulation techniques;
- Performance Analysis.

Work has been done on:

- Effective data oriented formalism for signal-processing application;
- Use of control oriented formalism for test generation;
- Use of UML for signal signal-processing application.

There was a large activity on the Cyber Enterprise specification and prototyping.

The contribution of DaRT concerns distributed Kahn process networks, UML profile for intensive signal processing, evaluation of SystemC for distributed and heterogeneous simulation of IP oriented SoC.

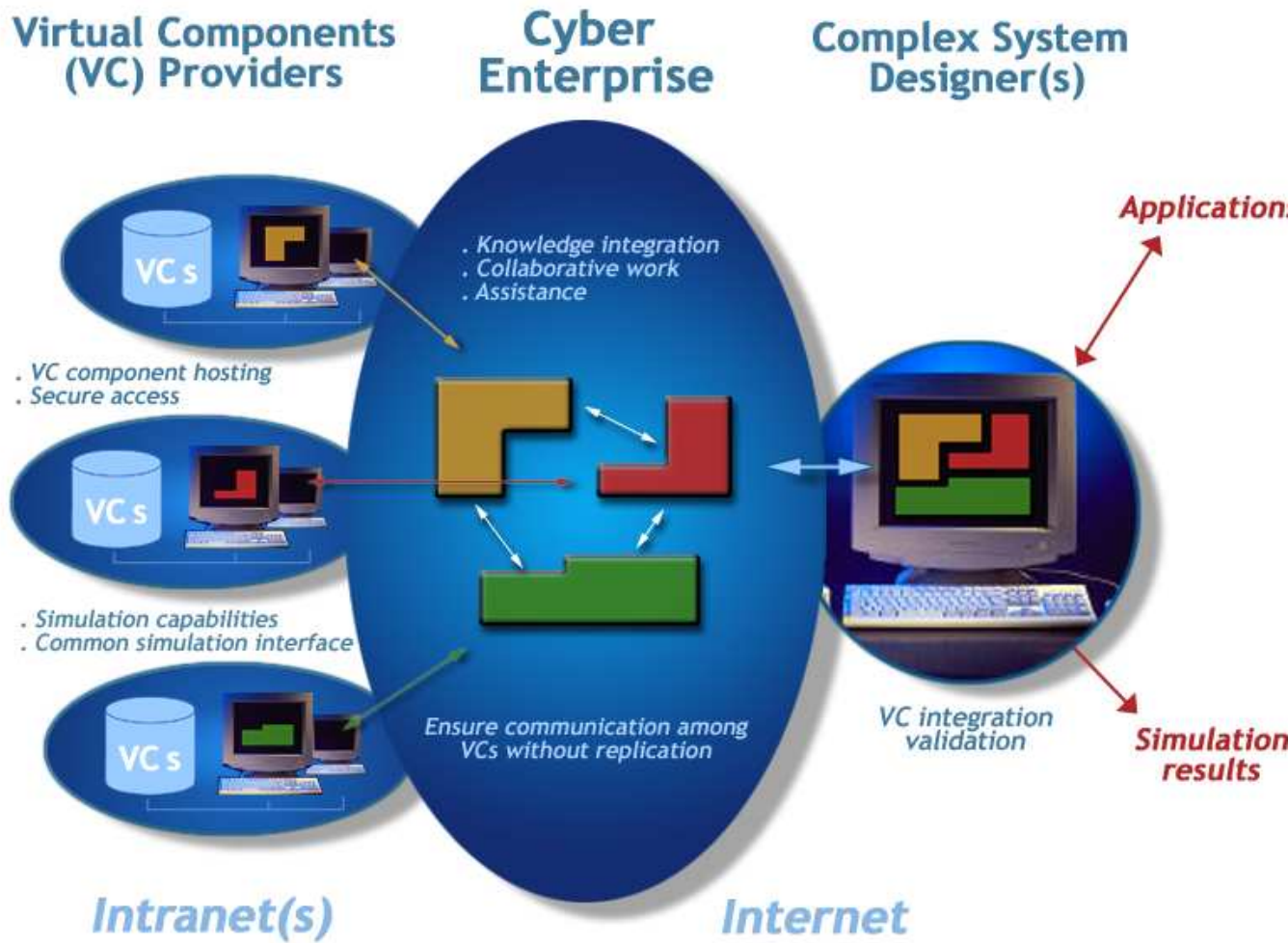


Figure 6.

## 7.2. Prompt2Implementation itea Project

Currently, methodologies and tools are only available for high level specification of complex systems using UML or other application-oriented languages. Ensuring coherence between the design and implementation phases is therefore a major issue. The traditional approach –validating real-time embedded applications using hand-made optimisation very late in the process– requires the availability of all hardware and software, is expensive and increases precious time to market. There is clearly a need for integrated methods and tools.

### 7.2.1. Partners:

Esterel Technologies, Thales Communication France, INRIA Rocquencourt (AOSTE), Nokia, Tampere University of Technology, University of Turku.

The goal of Prompt2Implementation is to define a design methodology for Real-Time Embedded Systems, based into an immersion of the partners previous know-how and existing skills into a relevant extension of the UML unified modeling framework. The resulting RTE profile will address the HW/SW codesign domain that is currently hardly addressed in the UML community.

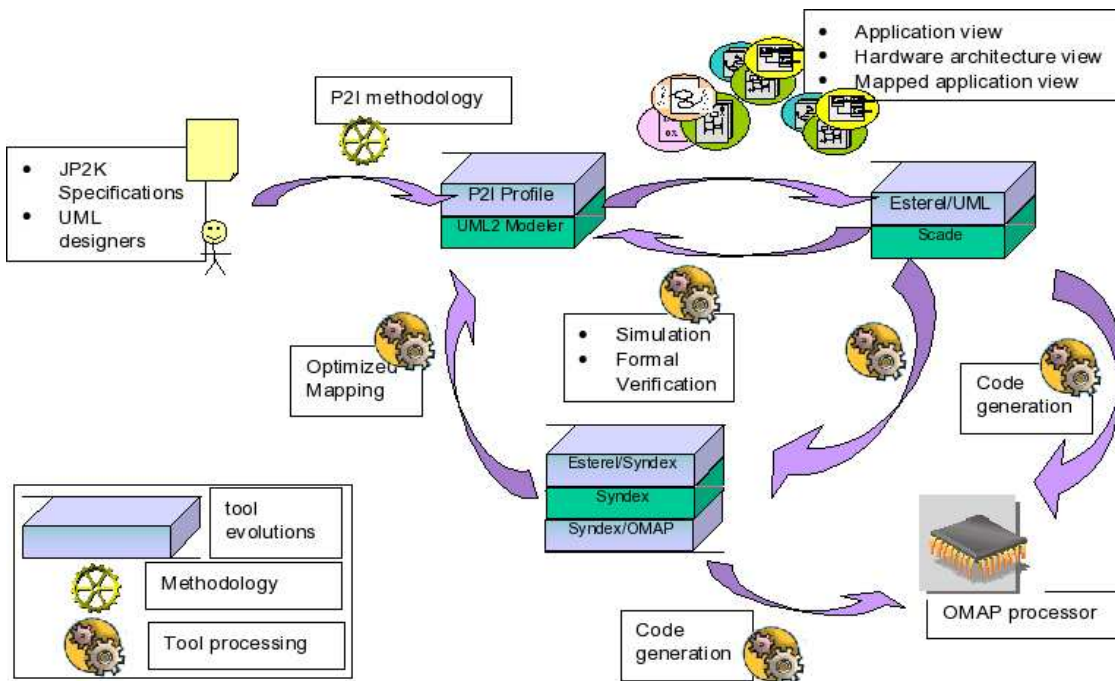


Figure 7.

This objective will require the following action steps:

- Provide the list of formalisms and methods used so-far by P2I partners, and study their common features as well as their complementarities;
- Extract the conceptual modeling needs to usefully cover the range of techniques aimed at;
- Study the existing UML representation (or lack of) for this RTE domain, and provide tentative solutions. Currently we shall not face the standardization compromise issues;
- Demonstrate the methodology (in its current, possibly transient state) on a non-trivial case study involving several partners.

We feel that such a specific profile, taking appropriately into account both the characteristic features of the aimed architectural platform and the characteristics of the application data dependencies at the proper level of details, could be exploited to benefit specific tools. In particular it could allow early verification and validation (sometimes on non-functional aspects), automatic code generation and automatic optimized code partitioning on heterogeneous embedded hardware target.

The contribution of DaRT in this project concerns the definition of profiles for application and architecture models. We are working on data low control flow integration in a UML profile. We exploit our MDA transformation tools to interact with Scade and Syndex tools

### 7.3. The PROTES Project: A Carroll Project

#### 7.3.1. 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.

### 7.4. Collaboration with Prosilog

#### 7.4.1. 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.

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.

### 7.5. SoCLib RNRT Platform Project

#### 7.5.1. 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.

### 7.6. ECSI member

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 is becoming an ECSI member this year. In this context we organize the next ECSI conference in Lille: FDL'04.



## 8. Other Grants and Activities

### 8.1. European initiatives

#### 8.1.1. *Eurosoc ex NoE Project*

This European Network of Excellence on Design, Verification, Testing, Standardization and Training for Nanometer CMOS System-on-Chip was not successful. But we continue to be contributor to this action in Europe.

### 8.2. International initiatives

#### 8.2.1. *Partnership with the 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. We will submit a proposal of associated INRIA team in January 2004.

### 8.3. National initiatives

#### 8.3.1. *CNRS “Action Spécifique compilation pour l’embarqué”*

We have actively participated to the thinking about new research directions in the field of compilation for embedded systems that was carried in the year 2003 in the frame of the *Action Spécifique compilation pour l’embarqué* of the CNRS.

#### 8.3.2. *Other 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.

### 8.4. Enseignement

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 and Samy Meftali) and “introduction to real-time operating systems” (Philippe Marquet).

## 10. Bibliography

### Major publications by the team in recent years

- [1] A. AMAR, P. BOULET, J.-L. DEKEYSER. *Towards Distributed Process Networks with CORBA*. in « Parallel and Distributed Computing Practice on Algorithms », 2003, Special Issue on Parallel and Distributed Computing Practice on Algorithms.
- [2] A. AMAR, P. BOULET, J.-L. DEKEYSER, F. THEEUWEN. *Distributed Process Networks Using Half FIFO Queues in CORBA*. in « ParCo’2003 », series Parallel Computing, Dresden, Germany, September, 2003.
- [3] 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.

- [4] P. BOULET, J.-L. DEKEYSER, C. DUMOULIN, P. MARQUET, P. KAJFASZ, D. RAGOT. *Sophocles: Cyber-Enterprise for System-on-Chip Distributed Simulation – Model Unification*. in « IFIP International Workshop On IP Based System-on-Chip Design », pages 325-330, November, 2003.
- [5] C. DUMOULIN, J.-L. DEKEYSER, B. KOKOSZKO, S. PULON, G. CRISTAU. *Interoperability between Design and Simulation tools using Model Transformation techniques*. in « FDL'03 », ECSI, Frankfurt, September, 2003.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] S. NIAR, L. EECKHOUT, K. DEBOSSCHERE. *Comparing multiported cache schemes*. in « PDPTA-2003 », June, 2003.
- [10] H. SBeyti, S. NIAR, L. EECKHOUT. *Adaptive Prefetching for Multimedia Applications in Embedded Systems*. in « DATE'04 », EDA IEEE, Paris, France, February, 2004, to appear.

### **Doctoral dissertations and “Habilitation” theses**

- [11] A. AMAR. *Support d'exécution pour le metacomputing à l'aide de CORBA*. Ph. D. Thesis, Université des sciences et technologies de Lille, Laboratoire d'informatique fondamentale de Lille, December, 2003, (In French).

### **Publications in Conferences and Workshops**

- [12] P. BOULET, J.-L. DEKEYSER, C. DUMOULIN, P. MARQUET. *MDA for SoC Embedded Design, Intensive Signal Processing Experiment*. in « SIVOES-MDA », San Francisco, USA, November, 2003, Extended version of BDDM03.

### **Internal Reports**

- [13] A. AMAR, P. BOULET, J.-L. DEKEYSER, F. THEEUWEN. *Distributed Process Networks Using Half FIFO Queues in CORBA*. Research Report, number RR-4765, INRIA, March, 2003, <http://www.inria.fr/rrrt/rr-4765.html>.
- [14] C. DUMOULIN, P. BOULET, J.-L. DEKEYSER, P. MARQUET. *UML 2.0 Structure Diagram for Intensive Signal Processing Application Specification*. Research Report, number RR-4766, INRIA, March, 2003, <http://www.inria.fr/rrrt/rr-4766.html>.



## Miscellaneous

- [15] J.-L. DEKEYSER, C. DUMOULIN. *MDA for SoC simulation*. Sophocles fringe workshop, DATE'03, March, 2003.

## Bibliography in notes

- [16] J. MILLER, J. MUKERJI, editors, *MDA Guide (Draft Version 0.2)*. <http://www.omg.org/docs/ab/03-01-03.pdf>, 2003.
- [17] M. D. ADAMS. *The JPEG-2000 Still Image Compression Standard*. Technical report, number N2412, ISO/IEC JTC 1/SC 29/WG 1, September, 2001, <http://www.jpeg.org/wg1n2412.pdf>.
- [18] R. ALLEN, K. KENNEDY. *Optimizing Compilers for Modern Architectures: A Dependence-based Approach*. Morgan Kaufmann Publishers, October, 2001, [http://www.mkp.com/books\\_catalog/catalog.asp?ISBN=1-55860-286-0](http://www.mkp.com/books_catalog/catalog.asp?ISBN=1-55860-286-0).
- [19] K. ARNOLD, J. GOSLING, D. HOLMES. *The Java Programming Language*. edition 3rd, Addison-Wesley, 2000.
- [20] G. BERRY. *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT Press, 1998, chapter The Foundations of Esterel, <http://www-sop.inria.fr/meije/esterel/doc/main-papers.html>.
- [21] P. BOULET, A. DARTE, G.-A. SILBER, F. VIVIEN. *Loop parallelization algorithms: From parallelism extraction to code generation*. in « Parallel Computing », number 3-4, volume 24, May, 1998, pages 421–444.
- [22] 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 », pages 105–112, Mantova, Italy, February, 2001.
- [23] E. CONSORTIUM. *The Eclipse Project*. 2003, <http://www.eclipse.org>.
- [24] 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 », volume 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>.
- [25] A. DARTE, Y. ROBERT, F. VIVIEN. *Scheduling and Automatic Parallelization*. Birkhauser Boston, 2000, <http://www.birkhauser.com/detail.tpl?isbn=0817641491>.
- [26] 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 « Grets-i », Juan-Les-Pins, France, September, 1995.
- [27] A. DEMEURE, Y. DEL GALLO. *An Array Approach for Signal Processing Design*. in « Sophia-Antipolis conference on Micro-Electronics (SAME 98) », France, October, 1998.

- [28] P. DUMONT. *Étude des Transformations d'un Code Array-OL dans Gaspard*. Research Report, number 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>.
- [29] C. DUMOULIN. *MDA Transf: A Model to Model Transformation Engine*. dec, 2003, <http://www.lifl.fr/west/mdaTransf>.
- [30] D. D. GAJSKI, R. KUHN. *Guest Editor Introduction: New VLSI-Tools*. in « IEEE Computer », number 12, volume 16, December, 1983, pages 11-14.
- [31] T. GARDNER, C. GRIFFIN, J. KOEHLER, R. HAUSER. *A Review of OMG MOF 2.0 Query / Views / Transformations Submissions*. <http://www.omg.org/docs/ad/03-08-02.pdf>, July, 2003, OMG paper.
- [32] 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) », ACM Press, pages 74–78, New York, May 3–5, 1999.
- [33] T. GROTKER, S. LIAO, AL. *System Design with SystemC*. Kluwer Academic Publishers, 2002.
- [34] M. GUPTA, S. MUKHOPADHYAY, N. SINHA. *Automatic Parallelization of Recursive Procedures*. in « International Journal of Parallel Programming », number 6, volume 28, 2000, pages 537-562, <http://citeseer.nj.nec.com/gupta99automatic.html>.
- [35] 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.
- [36] ITRS. *Design, 2001 edition*. Technical report, International Technology Roadmap for Semiconductors, 2001, <http://public.itrs.net/>.
- [37] G. KAHN. *The Semantics of a Simple Language for Parallel Programming*. in « Information Processing 74: Proceedings of the IFIP Congress 74 », IFIP, North-Holland, J. L. ROSENFELD, editor, pages 471–475, August, 1974.
- [38] G. KAHN, D. B. MACQUEEN. *Coroutines and networks of parallel processes*. B. GILCHRIST, editor, in « Information Processing 77: Proceedings of the IFIP Congress 77 », North-Holland, 1977, pages 993–998.
- [39] V. LEFEBVRE, P. FEAUTRIER. *Optimizing Storage Size for Static Control Programs in Automatic Parallelizers*. in « European Conference on Parallel Processing », pages 356-363, 1997, <http://citeseer.nj.nec.com/lefebvre97optimizing.html>.
- [40] A. W. LIM. *Improving Parallelism and Data Locality with Affine Partitioning*. Ph. D. Thesis, Stanford University, September, 2001.
- [41] 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 », ACM Press, ACM, editor, pages 2–15, New York, NY, USA, 1993, <http://www.acm.org:80/pubs/citations/proceedings/plan/158511/p2-maydan/>.
- [42] M. MOTTL. *Automating Functional Program Transformation*. Technical report, University of Edinburgh, September, 2000, [http://www.ai.univie.ac.at/~markus/msc\\_thesis/](http://www.ai.univie.ac.at/~markus/msc_thesis/).
- [43] OMG. *MOF 2.0 Query / Views / Transformations RFP*. 2003, OMG paper.
- [44] 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/inf/eden/paper/ParejaPenaRubioSeguraSFP2000.ps>.
- [45] OBJECT MANAGEMENT GROUP, INC., editor, *MOF 2.0 Core Final Adopted Specification*. <http://www.omg.org/cgi-bin/doc?ptc/03-10-04>, 2003.
- [46] 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>.
- [47] M. SERRANO, P. WEIS. *Bigloo: A Portable and Optimizing Compiler for Strict Functional Languages*. in « Static Analysis Symposium », pages 366–381, 1995, <http://citeseer.nj.nec.com/serrano95bigloo.html>.
- [48] Y. SOREL, C. LAVARENNE. *SynDEX Documentation Index*. INRIA, 2000, <http://www-rocq.inria.fr/syndex/doc/>.
- [49] Y. SOREL, C. LAVARENNE. *Modèle unifié pour la conception conjointe logiciel-matériel*. in « Traitement du Signal (numéro spécial Adéquation Algorithme Architecture) », number 6, volume 14, 1997, pages 569–578, <http://www-rocq.inria.fr/syndex/pub.htm>.
- [50] 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 », IEEE Computer Society Press, pages 44–54, Los Alamitos, CA, USA, May, 1994.
- [51] J. SOULA. *Principe de Compilation d’un Langage de Traitement de Signal*. Thèse de doctorat (PhD Thesis), Laboratoire d’informatique fondamentale de Lille, Université des sciences et technologies de Lille, December, 2001, (In French).
- [52] P. TU, D. PADUA. *Chapter 8. Automatic Array Privatization*. in « Lecture Notes in Computer Science », volume 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>.
- [53] ESTEREL TECHNOLOGIES. *SoC Design, Validation and Verification*. <http://www.esterel-technologies.com/v3/?id=29453>, 2002.

- [54] OBJECT MANAGEMENT GROUP, INC.. *MOF Meta Object Facility, Specification, Version 1.3*. <http://www.omg.org/cgi-bin/doc?formal/00-04-03>, January, 2000.
- [55] OBJECT MANAGEMENT GROUP, INC., editor, *Common Object Request Broker Architecture (CORBA), Version 2.6*. [http://www.omg.org/technology/documents/formal/corba\\_iiop.htm](http://www.omg.org/technology/documents/formal/corba_iiop.htm), December, 2001.
- [56] OBJECT MANAGEMENT GROUP, INC., editor, *U2 Partners' (UML 2.0): Superstructure, 2nd revised submission*. January, 2003.
- [57] OBJECT MANAGEMENT GROUP, INC., editor, *(UML 2.0): Superstructure Draft Adopted Specification*. July, 2003.
- [58] OMG ARCHITECTURE BOARD. *Model Driven Architecture (MDA)*. Technical report, number ormsc/2001-07-01, OMG, 2001.
- [59] OPEN SYSTEMC INITIATIVE. *SystemC*. <http://www.systemc.org/>, 2002.