INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team DaRT

Dataparallelism for Real-Time

Futurs

THEME COM

Activity Report

2004

Table of contents

1. Team

Head of project-team

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

Project assistant

Karine Levandowski

Faculty member

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]

Research scientist

Cédric Dumoulin [ITEA project grant]

Research scientist (partner)

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

Post-doctoral fellow

Anouar Dziri [Teaching Assistant, Université Lille 2]
Luc Charest [Post-doctoral fellow, INRIA Futurs]

Ph. D. student

Ahmad-Chadi Aljundi [Syria grant]
Rabie Ben Atitalah [Interreg project grant]
Lossan Bondé [Burkina Fasso grant]
Arnaud Cuccuru [ITEA project grant]
Philippe Dumont [Teaching Assistant, Université des Sciences et Technologies de Lille]
Ouassila Labbani [CNRS and regional grant]
Sébastien Le Beux [Interreg project grant]
Ashish Meena [ITEA project grant]
Éric Piel [INRIA Futurs 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 [49] 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 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 [25], MDA [22], MOF [23] for the modeling and the automatic model transformations; Array-OL [37][38][32], synchronous languages (such as Esterel [29] or Lustre [48]), Kahn process networks [50] as computation models with strong semantics for verification; SystemC [60] for the simulations; VHDL for the synthesis; and Java [28] 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 [42]). 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 [66]). 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

Keywords: *MDA*, *MDA Transformation*, *MOF*, *Metamodel*, *Model*, *Modeling*, *UML*.

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

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 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 [22][30] 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 [58].

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 [24] 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 [59]) 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” [43]. 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.

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.

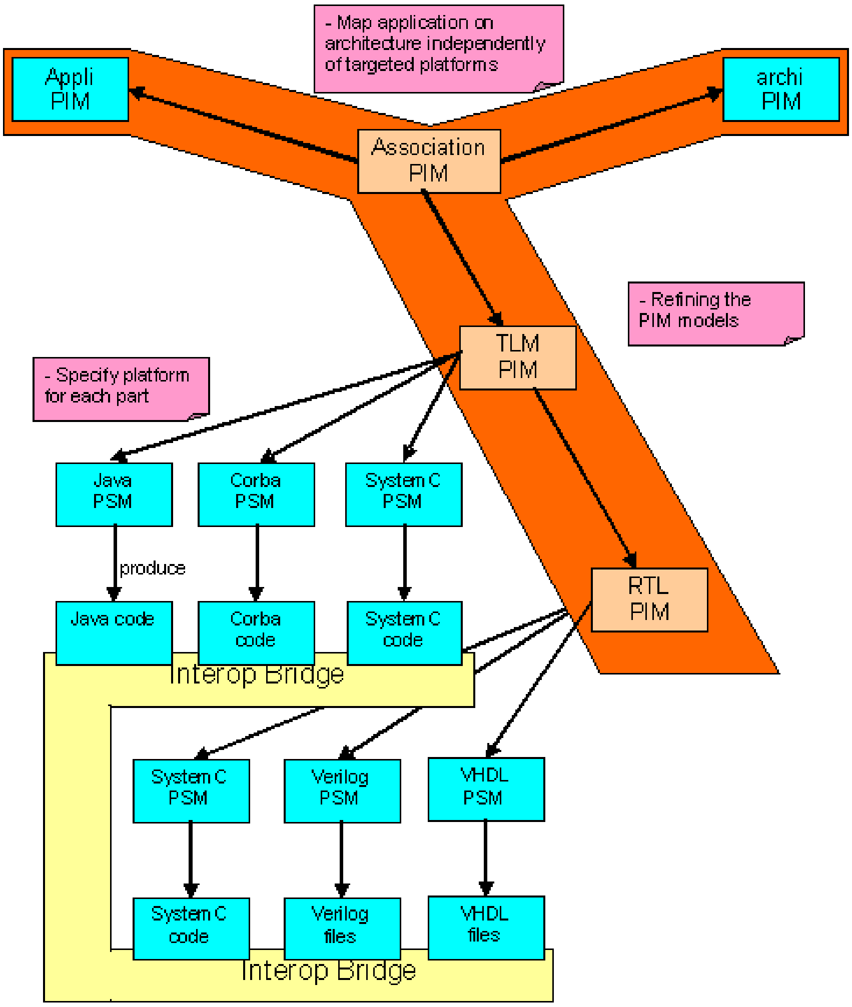

Figure 1. Overview of the metamodels for the "Y" design

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. Models and Metamodels

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.

In both application and architecture, components propose an interface materialized by their ports. The interfaces enable to encapsulate the structure and the behaviour of the components, and make them independent of their environment.

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.

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.

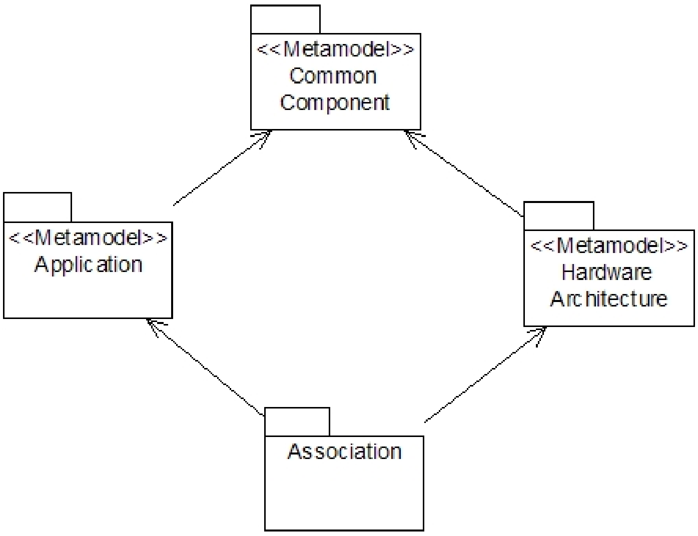

Figure 2. Metamodel Architecture

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.

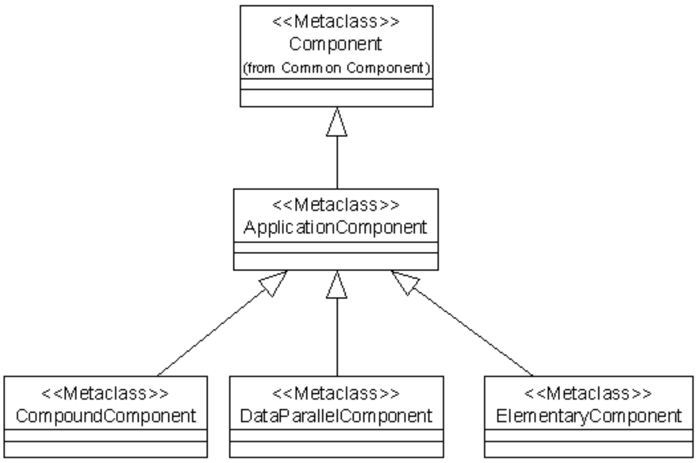

Figure 3. Application Metamodel

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. Potential data parallelism is explicitly described via Tilers, wich carry dependence vectors (paving and fitting) to express dependences between input/output arrays of the DataParallelComponent and input/output patterns of the inner repeated component part.

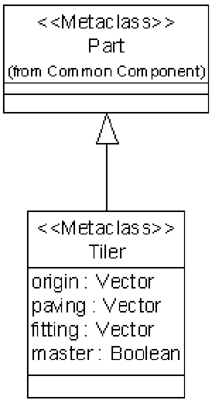

Figure 4. Tiler Definition

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

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.

Ouassila Labbani is pursuing her research about how to integrate mode automata further in our hierarchy of metamodels as her PhD, which she started in 2003.

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.

The hardware architecture meta-model introduces three kinds of components : Active, Passive and Interconnect components.

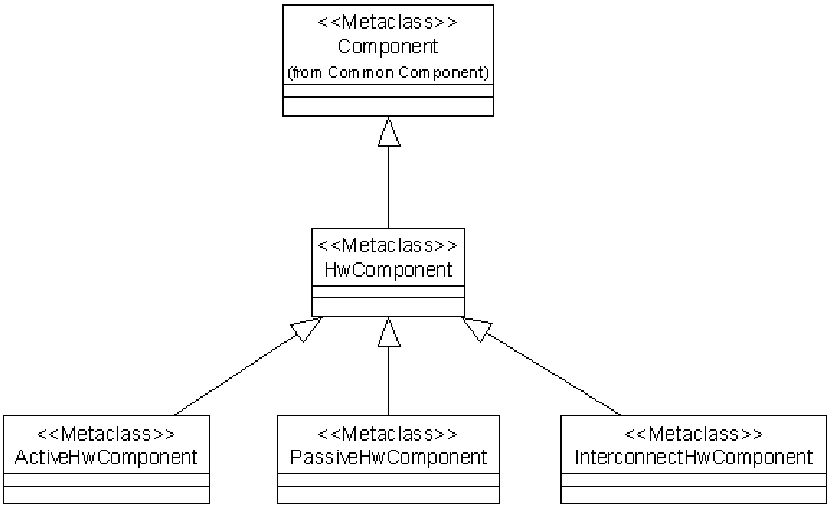

Figure 5. Hardware Architecture Metamodel

Active components symbolize resources which are able to read or write data into passive components. It may modify, or not, the data. It includes elements such as CPUs, FPGAs, ASICs or DMAs. It also includes more coarsegrained elements, such as SMP nodes inside a parallel machine.

Passive components symbolize resources which has the function of supporting data. It includes all kind of memories.

Interconnection components enable to connect active and passive components, or active components together. It includes elements as simple as a bus, or as complex as a multistage interconnection network.

Components communicate via a send/receive mechanism, and connections between components (via their ports) represent data paths offered by the architecture.

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.

Arnaud Cuccuru has been working towards his Ph. D. on this subject since september 2002.

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 active 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, passive and active 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.

3.2.5.5. Characterization

In order to automate the construction of such an association model and to optimize it, one needs to add some characteristics to the application and hardware architecture models. Informations such as real-time or power consumption constraints characterize the application model. In the hardware architecture model, the hardware components are characterized by frequency, bus width, memory size, bus protocol, etc. The characteristics that depend both on the application and the hardware architecture are located in the association model. These are the running time or the power consumption of the application components on the different hardware components.

3.2.5.6. Optimization

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. We have developed code transformations that 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. To compute the optimization, we use a globally irregular, locally regular heuristic, combining a global list heuristic to handle the task parallelism with a local regular heuristic to handle the data parallelism.

3.2.5.7. 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.

3.2.5.8. 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 [58] 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. Lossan Bondé will pursue this work in his Ph. D. started in september 2003.

3.3. Optimization Techniques

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

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

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 [31][53][36][27][35] with loop transformation, scheduling and mapping techniques.
- Memory management [52][54][68] to reuse the storage space while preserving parallelism.
- Pure functional language compilation [63][61][47][55] with techniques such as static typing, higher order functions, derecursivation, partial evaluation, etc.
- Signal processing specific optimizations [62].

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 have studied Array-OL to Array-OL code transformations [32][67][40][39]. Array-OL [37][38] 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 currently study the relations of the Array-OL model with other computation models such as Kahn Process Networks [50][51] and multidimensional synchronous dataflow [57][56].

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.

This works 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 [66][65][45] that is implemented in the SynDEx [64] 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 (GILR). We have shown in [33] that GILR heuristics can improve the optimization in several ways:

- better optimization (reduced latency),
- faster optimization (reduced complexity),
- more compact generated code.

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 data-parallel code transformations presented before in a global heuristic in order to deal efficiently with the data-parallelism 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 is working towards 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 etchniques 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

Keywords: *SystemC, TLM*.

Participants: Ahmad-Chadi Aljundi, Pierre Boulet, Jean-Luc Dekeyser, Samy Meftali, Smaïl Niar, Mickaël Samyn, Joël Vennin.

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 four levels of abstraction (this is the case of SystemC [46]):

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 Transfert 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 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 6).

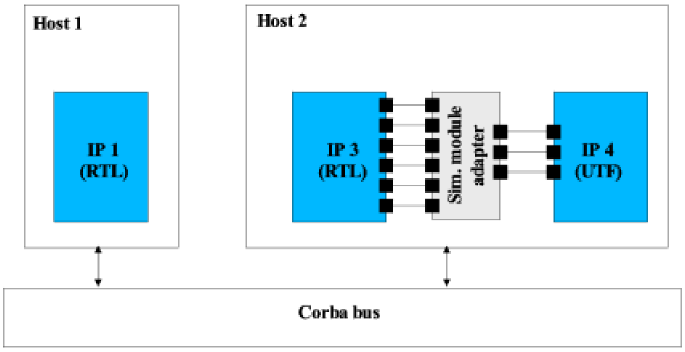

Figure 6. Distributed SystemC Simulation

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.

Mickaël Samyn is developping 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 developpment is integrated in the Gaspard prototype and uses the MDA Transf 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 [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 [14] 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 suject.

3.4.2.3. TLM: Transactional Lavel 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 later 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 by Ahmad-Chadi Aljundi, and Isaac Scherson [11], [12] 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.

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 frame 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.

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 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.

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 [26]. 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, 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 [44], and will follow the evolutions of this standard.

The tool is available as an open source distribution [41]. 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.

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 [34]. 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.

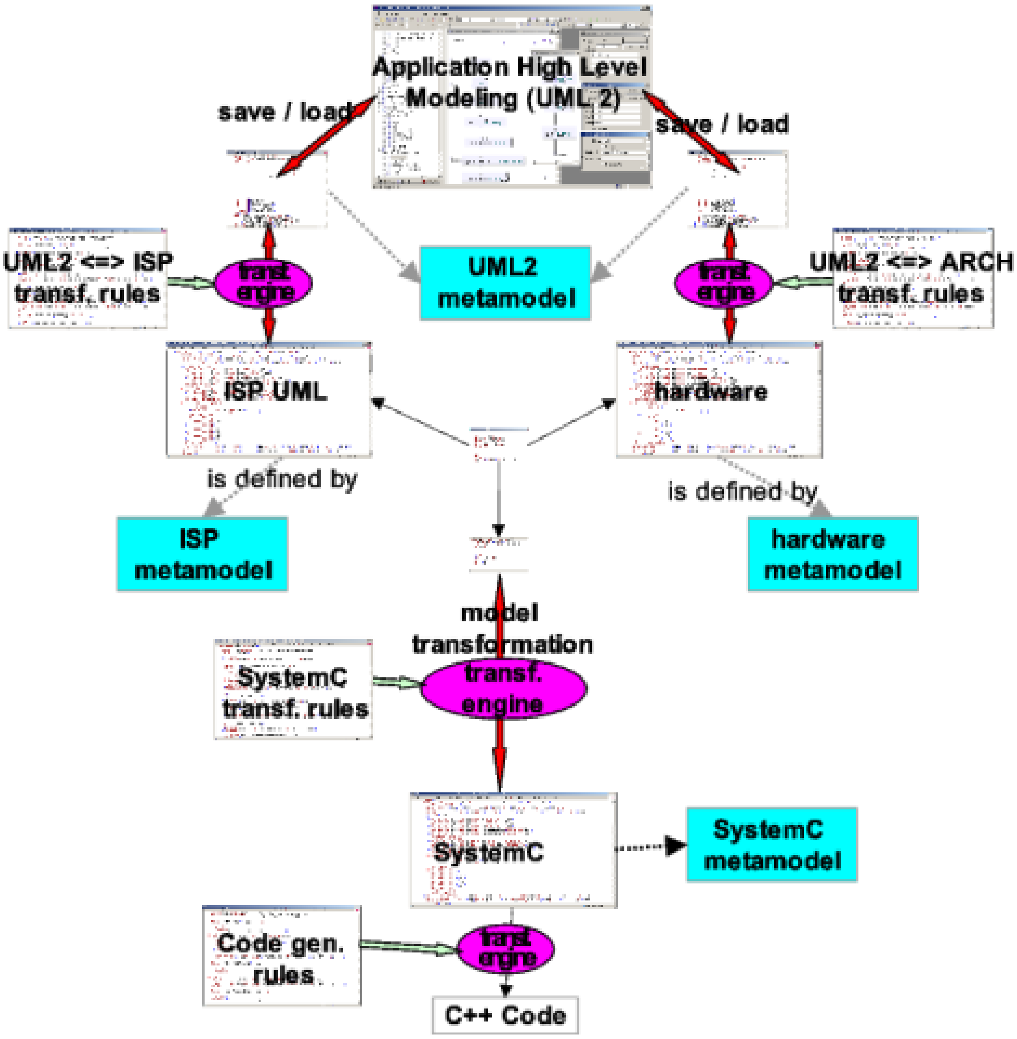

Figure 7. Overview of the Developement Flow with Gaspard

6. Contracts and Grants with Industry

6.1. 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.

6.1.1. 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.

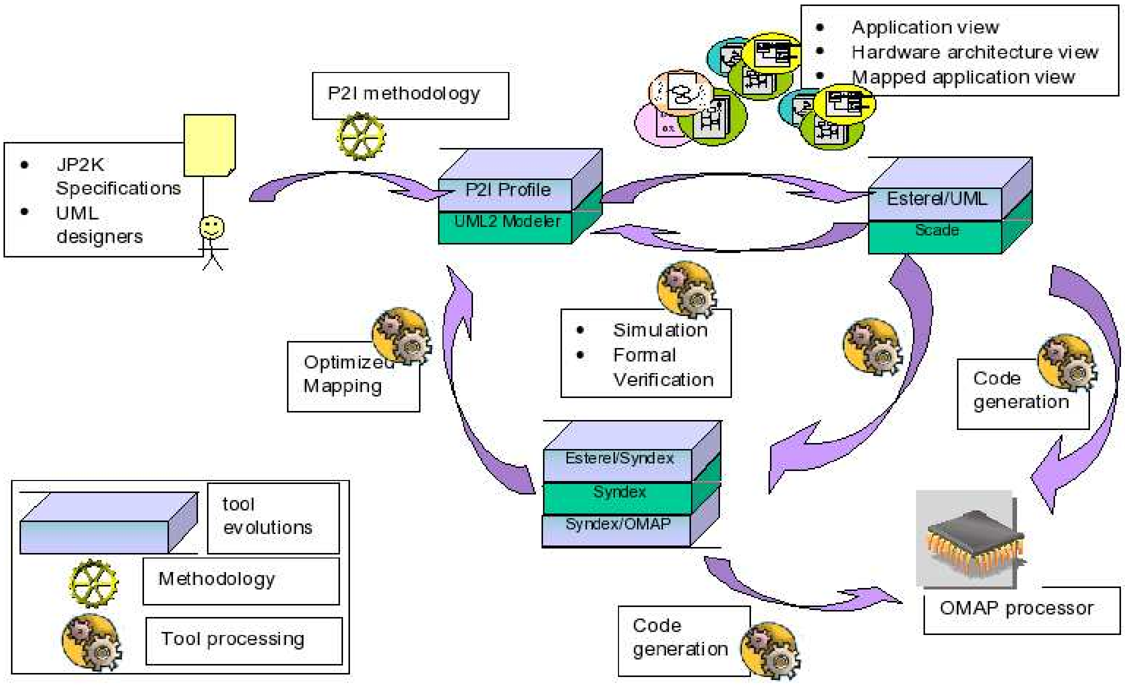

Figure 8.

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

6.2. ModEsay Interreg III A Franco/English

The project 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.

6.2.1. Partners:

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 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.

6.3. The PROTES Project: A Carroll Project

6.3.1. 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.

6.4. Collaboration with Prosilog

6.4.1. 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.

6.5. Collaboration with CEA List

6.5.1. 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.

6.6. SoCLib RNRT Platform Project

6.6.1. 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.

6.7. 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.

7. Other Grants and Activities

7.1. International initiatives

7.1.1. Organization of the 2004 Forum on Specification and Design Languages (FDL’04)

Pierre Boulet was the general chair of the FDL’04 in September 2004 in Lille. This workshop was co-organized by ECSI (the European Chip and Socket Design Initiative), the university of Lille and the INRIA Futurs. It was structured around 4 themes:

- Analog and Mixed-Signal Systems (chair: Alain Vachoux).

- C/C++-Based System Design (chair: Eugenio Villar).
- Languages for Formal Specification and Verification (chair: Wolfgang Müller).
- UML-based System Specification & Design (chair: Piet van der Putten).

A book with a selection of the best papers of FDL04 is being edited by Pierre Boulet and should be available during the second quarter of 2005.

7.1.2. 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 have submit a proposal of associated INRIA team in 2004.

7.2. National initiatives

7.2.1. 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.

7.3. 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).

8. 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", Special Issue on Parallel and Distributed Computing Practice on Algorithms, 2003.

[2] 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.

[3] P. BOULET, A. CUCCURRU, 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.

[4] 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.

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

[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, Paris, France", EDA IEEE, February 2004.

Publications in Conferences and Workshops

[11] A.-C. ALJUNDI, J.-L. DEKEYSER. *The Effect of the Degree of Multistage Interconnection Networks on their Performance: the Case of Delta and Over-sized Delta Networks*, in "2004 Euromicro on Parallel and Distributed Processing, Coruna, Spain", February 2004.

[12] A.-C. ALJUNDI, J.-L. DEKEYSER, M.-T. KECHADI. *On the Scalability of Multistage Interconnection Networks*, in "IEEE first International Conference on Information & Communication Technologies: from Theory to Applications, Damascus, Syria", April 2004.

[13] L. BONDÉ, C. DUMOULIN, J.-L. DEKEYSER. *Metamodels and MDA Transformations for Embedded Systems*, in "FDL04, Lille, France", September 2004.

[14] S. MEFTAL, J. VENNIN, J.-L. DEKEYSER. *Méthodologie de simulation multi niveaux, pour la conception de systèmes monopuces en SystemC*, in "CISC04, Jijel, Algeria", sep 2004.

[15] S. MEFTALI, J.-L. DEKEYSER. *An Optimal Charge Balancing Model for Fast Distributed SystemC Simulation in IP/SoC Design*, in "The 4th IEEE International Workshop System-on-Chip for Real-Time Applications (IWSOC 04), Banff, Alberta, Canada", July 2004.

[16] S. MEFTALI, J.-L. DEKEYSER. *SoC P2P: A Peer-to-Peer IP Based SoCs Design and Simulation Tool*, in "5th IFIP Working Conference on Virtual Enterprises (PRO-VE'04), Toulouse, France", August 2004.

[17] S. MEFTALI, M. SAMYN, J.-L. DEKEYSER. *Approche MDA, avec plateforme SystemC, pour la conception de systèmes monopuces dédiés au traitement de signal intensif*, in "CISC04, Jijel, Algeria", sep 2004.

[18] M. SAMYN, S. MEFTALI, J.-L. DEKEYSER. *Performances Estimation Metamodel for MDA Based SoC Design*, in "International Workshop on IP Based SoC design, Grenoble, France", December 2004.

[19] M. Samyn, S. Meftali, J.-L. Dekeyser. *MDA Based, SystemC Code Generation, Applied to Intensive Signal Processing Applications*, in "FDL04, Lille, France", September 2004.

[20] E. Turbatu, S. Meftali, S. Niar, J.-L. Dekeyser. *An automatic communication synthesis for high level SoC design using transaction level modeling*, in "FDL04, Lille, France", September 2004.

[21] 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.

Bibliography in notes

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

[23] Object Management Group, Inc. (editor). *MOF 2.0 Core Final Adopted Specification*, 2003, http://www.omg.org/cgi-bin/doc?ptc/03-10-04.

[24] 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.

[25] 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.

[26] M. D. Adams. *The JPEG-2000 Still Image Compression Standard*, Technical report, n° N2412, ISO/IEC JTC 1/SC 29/WG 1, September 2001.

[27] 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

[28] K. Arnold, J. Gosling, D. Holmes. *The Java Programming Language*, 3rd, Addison-Wesley, 2000.

[29] 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.

[30] OMG. A. Board. *Model Driven Architecture (MDA)*, Technical report, n° ormsc/2001-07-01, OMG, 2001.

[31] P. Boulet, A. Darte, G.-A. Silber, F. Vivien. *Loop parallelization algorithms: From parallelism extraction to code generation*, in "Parallel Computing", vol. 24, n° 3-4, May 1998, p. 421–444.

[32] 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.

[33] P. Boulet, A. Meena. *The case for Globally Irregular Locally Regular Algorithm Architecture Adequation*, in "Journ es Francophones sur l'Ad quation Algorithme Architecture, Dijon, France", January 2005.

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

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

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

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

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

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

[44] 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.

[45] 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.

[46] T. GROTKER, S. LIAO, AL. *System Design with SystemC*, Kluwer Academic Publishers, 2002.

[47] M. GUPTA, S. MUKHOPADHYAY, N. SINHA. *Automatic Parallelization of Recursive Procedures*, in "International Journal of Parallel Programming", vol. 28, n^o 6, 2000, p. 537-562, http://citeseer.nj.nec.com/gupta99automatic.html.

[48] 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.

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

[50] 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.

[51] 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.

[52] 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.nj.nec.com/lefebvre97optimizing.html.

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

[54] 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/.

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

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

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

[58] OMG. *MOF 2.0 Query / Views / Transformations RFP*, OMG paper, 2003, http://www.omg.org/techprocess/meetings/schedule/MO

[59] 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.

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

[61] 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.

[62] 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.

[63] M. Serrano, P. Weis. *Bigloo: A Portable and Optimizing Compiler for Strict Functional Languages*, in "Static Analysis Symposium", 1995, p. 366-381, http://citeseer.nj.nec.com/serrano95bigloo.html.

[64] Y. Sorel, C. Lavarenne. *SynDEx Documentation Index*, INRIA, 2000, http://www-rocq.inria.fr/syndex/doc/.

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

[66] 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.

[67] 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.

[68] 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.