IN PARTNERSHIP WITH:
CNRS

Université des sciences et technologies de Lille (Lille 1)

Activity Report 2012

Project-Team DART

Contributions of the Data Parallelism to Real Time

IN COLLABORATION WITH: Laboratoire d'informatique fondamentale de Lille (LIFL)

RESEARCH CENTER
Lille - Nord Europe

THEME
Embedded and Real Time Systems

Table of contents

1. Members **1**
2. Overall Objectives **1**
3. Research Program **2**
3.1. Introduction 2
3.2. Co-modeling for HP-SoC design 3
3.2.1. Foundations 3
3.2.1.1. System-on-Chip Design 3
3.2.1.2. Model-driven engineering 3
3.2.1.3. Models of computation 4
3.2.2. Past contributions of the team on topics continued in 2012 4
3.2.2.1. High-level modeling in Gaspard2 5
3.2.2.2. Intermediate concept modeling and transformations 5
3.2.2.3. MARTE extensions for reconfigurable based systems 7
3.2.2.4. Traceability 8
3.3. Model-based optimization and compilation techniques 8
3.3.1. Foundations 8
3.3.1.1. Optimization for parallelism 8
3.3.1.2. Transformation and traceability 8
3.3.2. Past contributions of the team on topics continued in 2012 9
3.3.2.1. Transformation techniques 9
3.3.2.2. Traceability 9
3.3.2.3. Modeling for GPU 9
3.3.2.4. GPGPU code production 10
3.3.2.5. From MARTE to OpenCL. 10
3.3.2.6. Formal techniques for construction, compilation and analysis of domain-specific languages 10
4. Application Domains **10**
4.1. Gaspard2 for avionic hybrid test platform design 10
4.2. Electromagnetic modeling 11
5. Software and Platforms **11**
6. New Results **12**
6.1. Hardware Distributed Control for Dynamic Reconfigurable Systems 12
6.2. Regular interconnection network for HP-SoC architecture 13
6.3. ReCoMARTE: A Marte Based Profile for Dynamic Reconfigurable Systems Modeling 13
6.4. Using Marte Profile for NoCs modeling 14
6.5. A Hardware Membranes Based Reconfiguration Services Implementation 14
6.6. Formal Techniques for General and Domain-Specific Languages 15
7. Bilateral Contracts and Grants with Industry **15**
7.1. Collaboration EADS IW, and Eurocopter 15
7.2. National Initiatives 15
7.2.1. ANR 15
7.2.1.1. ANR Famous 15
7.2.1.2. The ANR Open-People project 16
7.2.2. Competitivity Clusters 16
7.2.3. Within Inria 16
8. Partnerships and Cooperations **16**
8.1. European Initiatives 16
8.1.1. Collaboration with Romania 16
8.1.2. Collaboration with the Netherlands 17

8.2. International Research Visitors 17
9. Dissemination **17**
9.1. Scientific Animation 17
9.2. Teaching - Supervision - Juries 17
9.2.1. Teaching 17
9.2.2. Supervision 18
9.2.3. Juries 18
9.3. Popularization 18
10. Bibliography **18**

Project-Team DART

Keywords: Synchronous Languages, Programming Languages, Compiling, Processors, Embedded Systems, Software Engineering

Creation of the Project-Team: 2004 September 02*, updated into Team:* 2012 January 01, end of the Project-Team: 2012 December 31.

1. Members

Research Scientist
Vlad Rusu [Team Leader, Research scientist, HdR]

Faculty Members
Jean-Luc Dekeyser [Professor, Lille 1, HdR]
Frédéric Guyomarch [Associate professor, Lille 1]
Philippe Marquet [Associate professor, Lille 1]
Samy Meftali [Associate professor, Lille 1, HdR]

External Collaborator
Rabie Ben Atitallah [Associate professor, UVHC]

PhD Students
Wendell Rodrigues [VALEO and Regional grant, until Feb. 2012]
Chiraz Trabelsi [Inria grant]
Sana Cherif [ANR Famous]
Hana Krichen [co-advising Lille 1-ENIS Sfax]
Majdi Elhaji [co-advising Lille 1-Univ Monastir, until in Jul. 2012]
Georges Afonso [CIFRE EADS-Eurocopter]
Santosh-Kumar Rethinagiri [ANR Open-people, UVHC]
Paméla Wattebled [ANR Famous, UBS]
Andrei Arusoaie [co-advising Univ. of Lasi (Roumanie)]
Venkat Venkatasubramanian [UVHC]

Post-Doctoral Fellow
Christophe Calves [until March 2012]

Administrative Assistant
Corinne Jamroz [Inria]

2. Overall Objectives

2.1. Overall Objectives

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

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

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

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

All these ideas are implemented into a prototype co-design environment based on a model driven engineering approach, Gaspard. This open source platform is our test bench and is freely available. To help the designer, such an environment should help to evaluate several architectural solutions as well as several application specifications with regard to their performance and cost. We are able to estimate metrics from SystemC simulations and the refactoring algorithm defined for the transformation of loops to particular multiprocessors are the first steps for exploration. Automatic exploration system based on multi-objective methods has to transform the SoC description (size, network, memory, association). The space of solutions is huge and a fast simulation in SystemC at a high abstraction level is a good opportunity to reduce the space in a short delay. After that, a precise simulation at low level in SystemC or even in VHDL (synthetizable VHDL) can start to refine the solution. Code production is also focussed for GPGPU using OpenCL language as an intermediary target.
The main technologies we promote are UML 2 [39] and MARTE profil, MDE [68] and Eclipse EMF [36] for the modeling and model handling; Array-OL [52], [53], [48], [47] and synchronous languages [46] as computation models with strong semantics for verification; SystemC [40] for the simulation; OpenMP for the shared memory parallel execution; OpenCL for the massively parallel GPU;VHDL for the synthesis; and Java to code our prototypes.

3. Research Program

3.1. Introduction

The main research topic of the DaRT team-project concerns the hardware/software codesign of embedded systems with high performance processing units like DSP or SIMD processors. A special focus is put on multi processor architectures on a single chip (System-on-Chip). The contribution of DaRT is organized around the following items:

Co-modeling for High Performance 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 respect to the MARTE standard profile of the OMG group, which is dedicated to the modeling of embedded and real-time systems.

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

SoC simulation, verification and synthesis: We develop a SystemC based simulation environment at different abstraction levels for accurate performance estimation and for fast simulation. To address an architecture and the applications mapped on it, we simulate in SystemC at different abstraction levels the result of the SoC design. This simulation allows us to verify the adequacy of the mapping and the schedule, e.g., communication delay, load balancing, memory allocation. We also support IP (Intellectual Property) integration with different levels of specification. On the other hand, we use formal verification techniques in order to ensure the correctness of designed systems by particularly considering the synchronous approach. Finally, we transform MARTE models of data intensive algorithms in VHDL, in order to synthesize a hardware implementation.

3.2. Co-modeling for HP-SoC design

Modeling, UML, Marte, MDE, Transformation, Model, Metamodel

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

3.2.1. Foundations

3.2.1.1. System-on-Chip Design

SoC (System-on-Chip) can be considered as a particular case of embedded systems. SoC design covers a lot of different viewpoints including the application modeling by the aggregation of functional components, the assembly of existing physical components, the verification and the simulation of the modeled system, and the synthesis of a complete end-product integrated into a single chip.

The model driven engineering is appropriate to deal with the multiple abstraction levels. Indeed, a model allows several viewpoints on information defined only once and the links or transformation rules between the abstraction levels permit the re-use of the concepts for a different purpose.

3.2.1.2. Model-driven engineering

Model Driven Engineering (MDE) [68] is now recognized as a good approach for dealing with System on Chip design issues such as the quick evolution of the architectures or always growing complexity. MDE relies on the model paradigm where a model represents an abstract view of the reality. The abstraction mechanism avoids dealing with details and eases reusability.

A common MDE development process is to start from a high level of abstraction and to go to a targeted model by flowing through intermediate levels of abstraction. Usually, high level models contain only domain specific concepts, while technological concepts are introduced smoothly in the intermediate levels. The targeted levels are used for different purposes: code generation, simulation, verification, or as inputs to produce other models, etc. The clear separation between the high level models and the technological models makes it easy to switch to a new technology while re-using the previous high level designs. Transformations allow to go from one model at a given abstraction level to another model at another level, and to keep the different models synchronized

In an MDE approach, a SoC designer can use the same language to design application and architecture. Indeed, MDE is based on proved standards: UML 2 [38] for modeling, the MOF (Meta Object Facilities [63]) for metamodel expression and QVT [64] for transformation specifications. Some profiles, i.e. UML extensions, have been defined in order to express the specificities of a particular domain. In the context of embedded system, the MARTE profile in which we contribute follows the OMG standardization process.

3.2.1.3. Models of computation

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

Array-OL. The Array-OL language [52], [53], [48], [47] is a mixed graphical-textual specification language dedicated to express multidimensional intensive signal processing applications. It focuses on expressing all the potential parallelism in the applications by providing concepts to express data-parallel access in multidimensional arrays by regular tilings. It is a single assignment first-order functional language whose data structures are multidimensional arrays with potentially cyclic access.

The synchronous model. The synchronous approach [46] proposes formal concepts that favor the trusted design of embedded real-time systems. Its basic assumption is that computation and communication are instantaneous (referred to as "synchrony hypothesis"). The execution of a system is seen through the chronology and simultaneity of observed events. This is a main difference from visions where the system execution is rather considered under its chronometric aspect (i.e., duration has a significant role). There are different synchronous languages with strong mathematical foundations. These languages are associated with tool-sets that have been successfully used in several critical domains, e.g. avionics, nuclear power plants.

In the context of the DaRT project, we consider declarative languages such as Lustre [50] and Signal [61] to model various refinements of Array-OL descriptions in order to deal with the control aspect as well as the temporal aspect present in target applications. The first aspect is typically addressed by using concepts such as mode automata, which are proposed as an extension mechanism in synchronous declarative languages. The second aspect is studied by considering temporal projections of array dimensions in synchronous languages based on clock notion. The resulting synchronous models are analyzable using the formal techniques and tools provided by the synchronous technology.

3.2.2. Past contributions of the team on topics continued in 2012

The new team DaRT has been created in order to finalize the works started in the DaRT EPI, and also to explore new topics. We here remind the past contributions of the team on the topics we continued to work on during 2012.

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

Metamodeling brings a set of tools which 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 initiate 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 the hardware architecture are modeled separately using similar concepts inspired by Array-OL to express the parallelism. The placement and scheduling of the application on the hardware architecture is then expressed in an association model.

All the previously defined models, application, architecture and association, are platform independent and they conform to the MARTE OMG Profil (figure 1). No component is associated with an execution, simulation or synthesis technology. Such an association targets a given technology (OpenMP, OpenCL, SystemC/PA, VHDL, etc.). Once all the components are associated with some IPs of the GasparLib library, the deployment is fully realized. This result can be transformed to further abstraction level models via some model transformations (figure 2).

The simulation results can lead to a refinement of the initial application, hardware architecture, association and deployment models. We propose a methodology to work with all these different models. The design steps are:

1. Separation of application and hardware architecture modeling.
2. Association with semi-automatic mapping and scheduling.
3. Selection of IPs from libraries for each element of application/architecture models, to achieve the deployment.
4. Automatic generation of the various platform specific simulation or execution models.
5. Automatic simulation or execution code generation with calls to the IPs.
6. Refinement at the highest level taking account of the simulation results.

3.2.2.1. High-level modeling in Gaspard2

In Gaspard2, models are described by using the recent OMG standard MARTE profile combined with a few native UML concepts and some extensions.

The new release of Gaspard2 uses different packages of MARTE for UML modeling. The Hardware Resource Model (HRM) concepts of MARTE enable to describe the hardware part of a system. The Repetitive Structure Modeling (RSM) concepts allow one to describe repetitive structures (DaRT team was the main contributor of this MARTE package definition). Finally, the Generic Component Modeling (GCM) concepts are used as the base for component modeling.

The above concepts are expressive enough to permit the modeling of different aspects of an embedded system:

- functionality (or applicative part): the focus is mainly put on the expression of data dependencies between components in order to describe an algorithm. Here, the manipulated data are mainly multidimensional arrays. Furthermore, a form of reactive control can be described in modeled applications via the notion of execution modes. This last aspect is modeled with the help of some native UML notions in addition to MARTE.
- hardware architecture: similar mechanisms are also used here to describe regular architectures in a compact way. Regular parallel computation units are more and more present in embedded systems, especially in SoCs. HRM is fully used to model these concepts. Some extensions are proposed for NoC design and FPGA specifications. The GPU have a particular memory hierarchy. In order to model the memory details, we extend the MARTE metamodel to describe low level characteristics of the memory.
- association of functionality with hardware architecture: the main issues concern the allocation of the applicative part of a system onto the available computation resources, and the scheduling. Here also, the allocation model takes advantage of the repetitive and hierarchical representation offered by MARTE to enable the association at different granularity levels, in a factorized way.

In addition to the above usual design aspects, Gaspard2 also defines a notion of *deployment* specification (see Figure 1) in order to select compilable IPs from libraries, at this time models can produce codes. The corresponding package defines concepts that (i) enable to describe the relation between a MARTE representation of an elementary component (a box with ports) to a text-based code (and Intellectual Property - IP, or a function with arguments), and (ii) allow one to inform the Gaspard2 transformations of specific behaviors of each component (such as average execution time, power consumption...) in order to generate a high abstraction level simulation in adequacy with the real system. Recently this package was extended to design reconfigurable systems using dynamical deployment.

3.2.2.2. Intermediate concept modeling and transformations

Gaspard2 targets different technologies for various purposes: formal verification, high-performance computing, simulation and hardware synthesis (Figure 1). This is achieved via model transformations that relate intermediate representations towards the final target representations.

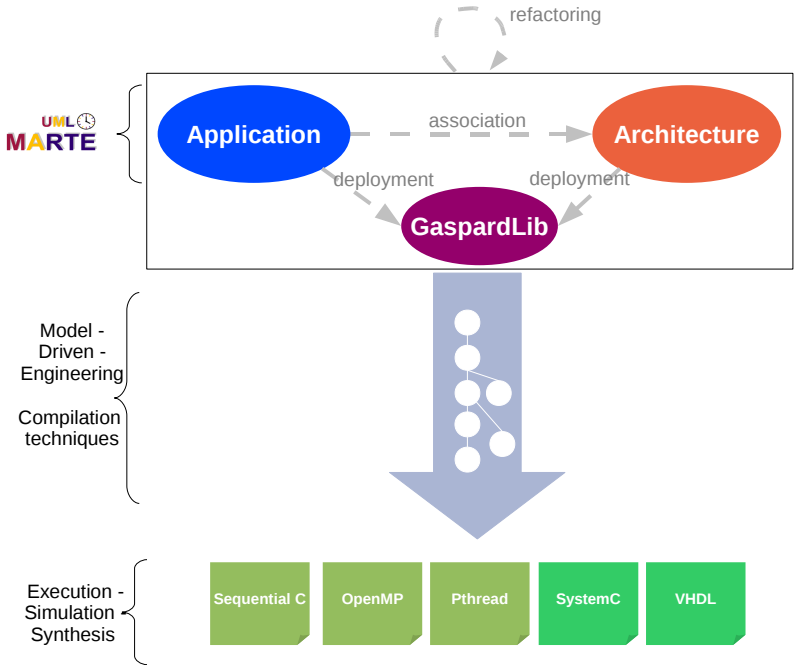

Figure 1. Overview of the design concepts.

- A metamodel for procedural language with OpenMP (OpenMP in Figure 1). It is inspired by the ANSI C and Fortran grammars and extended by OpenMP statements [41]. The aim of this metamodel is to use the same model to represent Fortran and C code. Thus, from an OpenMP model, it is possible to generate OpenMP/Fortran or OpenMP/C. The generated code includes parallelism directives and control loops to distribute task (IPs code) repetitions over processors [70].
- A VHDL metamodel (VHDL in Figure 1). It gathers the necessary concepts to describe hardware accelerators at the RTL (Register Transfer Level) level, which allows the hardware execution of applications. This metamodel introduces, *e.g.*, the notions of *clock* and *register* in order to manipulate some of the usual hardware design concepts. It is precise enough to enable the generation of synthetizable HDL code [60].
- The two metamodels SystemC and Pthread was redefined to implement both a multi-thread execution model. These are described in the " New results" part.
- Synchronous metamodel (Synchronous Equational). It was used to benefit of the verification tools of synchronous languages. It is not yet maintained in the new release of Gaspard2.

The transformation scheme. In order to target these metamodels, several transformations have been developed (Figure 2). *MartePortInstance* introduces into the MARTE metamodel the concept of *PortInstance* corresponding to an instance of port associated to a part. The *ExplicitAllocation* transformation explicits the association of each application part on the processing units, according to the association of other elements in the application hierarchy. The *LinkTopologyTask* transformation replaces the connectors between a component and an inner repeated part by a task managing the data (*TilerTask*). The scheduling of the application tasks is decomposed into three transformations, *Synchronisation* that associates, to each application component, a local graph of tasks corresponding to its parts; *GlobalSynchronization* that computes a global graph of tasks for the complete application from the local graphs of tasks; and *Scheduling* that schedules the tasks from the

global graph. *TilerMapping* maps the *TilerTasks* onto processors. The management of the data in the memory is performed through two transformations. *MemoryMapping* maps the data into memory *i.e.* creates the variables and allocates address spaces. *AddressComputation* computes addresses for each variable. Finally, some transformations are dedicated to targets: *Functional* introduces the concepts relative to procedural languages. *pThread* transforms MARTE elementary tasks into threads and the connectors into buffers. *SystemC* traduces the MARTE architecture into concepts of the SystemC language.

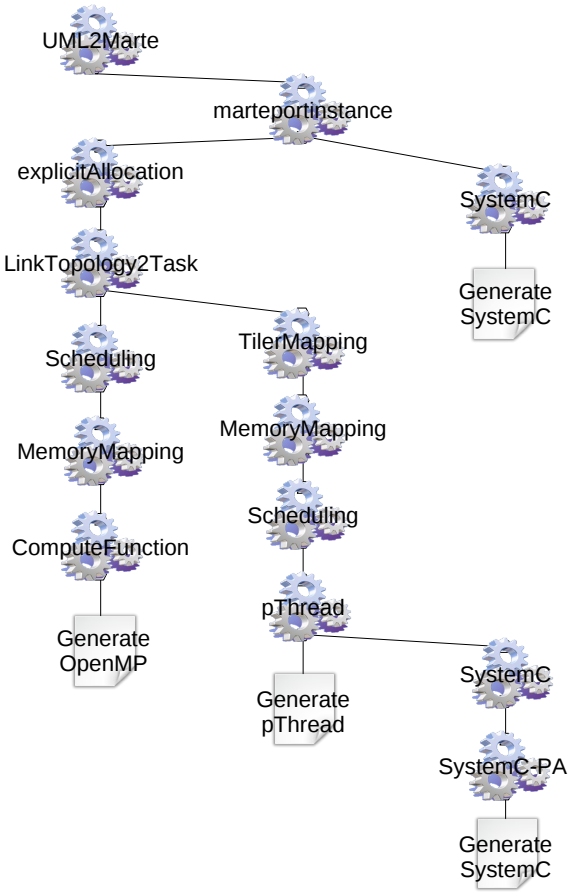

Figure 2. Overview of the transformation chains.

3.2.2.3. MARTE extensions for reconfigurable based systems

Reconfigurable FPGA based Systems-on-Chip (SoC) architectures are increasingly becoming the preferred solution for implementing modern embedded systems. However due to the tremendous amount of hardware resources available in these systems, new design methodologies and tools are required to reduce their design complexity.

In previous work, we provided an initial contribution to the modeling of these systems by extending MARTE profile to incorporate significant design criteria such as power consumption.

In its current version, MARTE lacks dynamic reconfiguration concepts. Even these later are necessary to model and implement rapid prototypes for complex systems.

Our objective is to define all necessary concepts for dynamic reconfiguration issues regarding configuration latency, resources number, etc. Afterwards, these concepts will be integrated to MARTE to obtain an extended and complete profile, which can be called Reconfigurable MARTE (RecoMARTE).

Our current proposals permit us to model fine grain reconfigurable FPGA architectures with an initial extension of the MARTE profile to model Dynamic Reconfiguration at a high-level description.

Since a controller is essential for managing a dynamically reconfigurable region, we modeled a state machine at high abstraction levels using UML state machine diagrams. This state machine is responsible for switching between the available configurations.

As a future work, we will analyze the reconfigurable design flow of Xilinx from the design partitioning to the bitstream generation stage. It is a starting point for understanding how to generate configuration files. Then, we will extract relevant data to define our own design flow.

3.2.2.4. Traceability

We use the transformation mechanism to assist a tester in the mutation analysis process dedicated to model transformations. The mutation analysis aims to qualify a test model set. More precisely, errors are voluntary injected in transformation and the ability of the test models set to highlight these errors is analyzed. If the number of highlighted errors, *i.e.* if the test model set is not enough qualified, new models have to be added in order to raise the set quality [62]. Our approach relies on the hypothesis that it is easier to modify an existing model than to create a new one from scratch. The local trace, coupled to a mutation matrix, helps the tester to identify adequate test models and their relevant parts to modify in order to improve the test data set. We propose a semi-automation approach that can automatically generate new test model in some cases and efficiently assist the testers in others cases [45].

3.3. Model-based optimization and compilation techniques

Scheduling, Mapping, Compilation, Optimization, Heuristics, Power Consumption, Data-parallelism

3.3.1. Foundations

3.3.1.1. Optimization for parallelism

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

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

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

3.3.1.2. Transformation and traceability

Model to model transformations are at the heart of the MDE approach. Anyone wishing to use MDE in its projects is sooner or later facing the question: how to perform the model transformations? The standardization process of Query View Transformation [64] was the opportunity for the development of transformation engine as Viatra, Moflon or Sitra. However, since the standard has been published, only few of investigating tools, such as ATL [1] (a transformation dedicated tool) or Kermeta [2] (a generalist tool with facilities to manipulate models) are powerful enough to execute large and complex transformations such as in the Gaspard2 framework. None of these engine is fully compliant with the QVT standard. To solve this issue, new engine relying on a subset of

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

the standard recently emerged such as QVTO [3] and smartQVT. These engines implement the QVT Operational language.

Traceability may be used for different purposes such as understanding, capturing, tracking and verification on software artifacts during the development life cycle [58]. MDE has as main principle that everything is a model, so trace information is mainly stored as models. Solutions are proposed to keep the trace information in the initials models source or target [71]. The major drawbacks of this solution are that it pollutes the models with additional information and it requires adaptation of the metamodels in order to take into account traceability. Using a separate trace model with a specific semantics has the advantage of keeping trace information independent of initial models [59].

3.3.2. Past contributions of the team on topics continued in 2012

The new team DaRT has been created in order to finalize the works started in the DaRT EPI, and also to explore new topics. We here remind the past contributions of the team on the topics we continued to work on during 2012.

3.3.2.1. Transformation techniques

In the previous version of Gaspard2, model transformations were complex and monolithic. They were thus hardly evolvable, reusable and maintainable. We thus proposed to decompose complex transformations into smaller ones jointly working in order to build a single output model [56]. These transformations involve different parts of the same input metamodel (e.g. the MARTE metamodel); their application field is localized. The localization of the transformation was ensured by the definition of the intermediary metamodels as delta. The delta metamodel only contains the few concepts involved in the transformation (i.e. modified, or read). The specification of the transformations only uses the concepts of these deltas. We defined the Extend operator to build the complete metamodel from the delta and transposed the corresponding transformations. The complete metamodel corresponds to the merge between the delta and the MARTE metamodel or an intermediary metamodel. The transformation then becomes the chaining of metamodel shifts and the localized transformation. This way to define the model transformations has been used in the Gaspard2 environment. It allowed a better modularity and thus also reusability between the various chains.

3.3.2.2. Traceability

Our traceability solution relies on two models the Local and the Global Trace metamodels. The former is used to capture the traces between the inputs and the outputs of one transformation. The Global Trace metamodel is used to link Local Traces according to the transformation chain. The local trace also proposes an alternative "view" to the common traceability mechanism that does not refers to the execution trace of the transformation engine. It can be used whatever the used transformation language and can easily complete an existing traceability mechanism by providing a more finer grain traceability [43].

Furthermore, based on our trace metamodels, we developed algorithms to ease the model transformation debug. Based on the trace, the localization of an error is eased by reducing the search field to the sequence of the transformation rule calls [44].

3.3.2.3. Modeling for GPU

The model described in UML with Marte profile model is chained in several inout transformations that adds and/or transforms elements in the model. For adding memory allocation concepts to the model, a QVT transformation based on «Memory Allocation Metamodel» provides information to facilitate and optimize the code generation. Then a model to text transformation allows to generate the C code for GPU architecture. Before the standard releases, Acceleo is appropriate to get many aspects from the application and architecture model and transform it in CUDA (.cu, .cpp, .c, .h, Makefile) and OpenCL (.cl, .cpp, .c, .h, Makefile) files. For the code generation, it's required to take into account intrinsic characteristics of the GPUs like data distribution, contiguous memory allocation, kernels and host programs, blocks of threads, barriers and atomic functions.

[3] http://www.eclipse.org/m2m/qvto/doc

3.3.2.4. GPGPU code production

The solution of large, sparse systems of linear equations « Ax=b » presents a bottleneck in sequential code executing on CPU. To solve a system bound to Maxwell's equations on Finite Element Method (FEM), a version of conjugate gradient iterative method was implemented in CUDA and OpenCL as well. The aim is to accelerate and verify the parallel code on GPUs. The first results showed a speedup around 6 times against sequential code on CPU. Another approach uses an algorithm that explores the sparse matrix storage format (by rows and by columns). This one did not increase the speedup but it allows to evaluate the impact of the access to the memory.

3.3.2.5. From MARTE to OpenCL.

We have proposed an MDE approach to generate OpenCL code. From an abstract model defined using UML/MARTE, we generate a compilable OpenCL code and then, a functional executable application. As MDE approach, the research results provide, additionally, a tool for project reuse and fast development for not necessarily experts. This approach is an effective operational code generator for the newly released OpenCL standard. Further, although experimental examples use mono device(one GPU) example, this approach provides resources to model applications running on multi devices (homogeneously configured). Moreover, we provide two main contributions for modeling with UML profile to MARTE. On the one hand, an approach to model distributed memory simple aspects, i.e. communication and memory allocations. On the other hand, an approach for modeling the platform and execution models of OpenCL. During the development of the transformation chain, an hybrid metamodel was proposed for specifying of CPU and GPU programming models. This allows generating other target languages that conform the same memory, platform and execution models of OpenCL, such as CUDA language. Based on other created model to text templates, future works will exploit this multi language aspect. Additionally, intelligent transformations can determine optimization levels in data communication and data access. Several studies show that these optimizations increase remarkably the application performance.

3.3.2.6. Formal techniques for construction, compilation and analysis of domain-specific languages

The increasing complexity of software development requires rigorously defined *domain specific modelling languages* (DSML). Model-driven engineering (MDE) allows users to define their language's syntax in terms of *metamodels*. Several approaches for defining operational semantics of DSML have also been proposed [69], [51], [42], [49], [65]. We have also proposed one such approach, based on representing models and metamodels as algebraic specifications, and operational semantics as rewrite rules over those specifications [54], [67]. These approaches allow, in principle, for model execution and for formal analyses of the DSML. However, most of the time, the executions/analyses are performed via transformations to other languages: code generation, resp. translation to the input language of a model checker. The consequence is that the results (e.g., a program crash log, or a counterexample returned by a model checker) may not be straightforward to interpret by the users of a DSML. We have proposed in [66] a formal and operational framework for tracing such results back to the original DSML's syntax and operational semantics, and have illustrated it on SPEM, a language for timed process management.

4. Application Domains

4.1. Gaspard2 for avionic hybrid test platform design

The emergence and the maturity of FPGA circuits for distributed and reconfigurable architectures offer the opportunity to explore real time problems in the field of avionic systems. FPGA becomes de facto a major processing element as same as general CPUs. As of now, the FPGA is widely used in the field of I/O component in order to connect the real equipment with the CPU host. Among the main features mapped into the FPGA in the original architecture, we quote the fast serial link and RAM IPs (Intellectual property) which are needed to ensure communication between CPU and FPGA. Additionally, the Base Time IP is needed for the global system synchronization. This minimal configuration based on FPGA can be duplicated several times and

connected together to build bigger test system or a complete simulator. Eurocopter expectation for the above-described architecture is to prototype some models which can be eligible and relocated in the FPGA. The objective is to increase the performances of these models and to reduce the communication latencies by the means of embedding the different parts in the same chip. To do so, we studied in this first year a real avionic test loop in order to extract the complex models that will be implemented in the FPGA. Different hardware model configurations have been explored to reach an optimal well-balanced global system using the ML403 Virtex-4 Xilinx board. Different tradeoffs in terms of performance and resource occupation in the FPGA are obtained. Later, these results will be used for dynamically adapt the system functioning according to the available resources and performance requirements.

As a second part, we used the MARTE profile to represent an hybrid system (CPU/FPGA).In the MARTE specification, an application is a set of tasks connected through ports. Tasks are considered as mathematical functions reading data from their input ports and writing data on their output ports. This specification has been used to model the avionic test loop. In addition, MARTE allows describing the hardware architecture in a structural way. Typical components such as HwProcessor, HwFPGA and HwRAM can be specified with their non-functional properties. We used this subset of MARTE in order to represent an hybrid multiprocessor architecture. The main component of this architecture is composed of the Xeon-X3370 processor (multicore CPU) and the Virtex-4 Xilinx FPGA. Furthermore, MARTE provides the Allocate concept as well as the concept specially crafted for repetitive structures Distribute. This latter concept gives a way to express regular distribution of tasks onto a set of processors or FPGA resources. The mapping step relies on two types of distribution (timeScheduling and spatialDistibution) depending on the target hardware platform (CPU/FPGA). The different models of our avionic test loop can be mapped onto the host multicore processor, the embedded processor (Microblaze) or the hardware resources in the FPGA.

4.2. Electromagnetic modeling

We collaborate with the L2EP specialized in electromagnetic modeling, on algorithms definition and the parallelization of their computations, especially on GPUs.

For the first point, we have designed a parallel version of the Finite Integration Technique (F.I.T). This is used to simulate electromagnetic phenomena. This technique is efficient if the mesh is generated by a regular hexahedron. Moreover the matrix system, obtained from a regular mesh can be exploited to use the parallel direct solver. In fact, in reordering the unknowns by the nested dissection method, it is possible to construct directly the lower triangular matrix with many processors without assembling the matrix system. During this year, we have used our parallel direct solver as a preconditionner for a sparse linear system coming from a FEM problem with a good efficiency[25].

For the second point, we have include our Gaspard2 generated code in Code_CARMEL, a software for electromagnetic fields simulations. This GPGPU code is now robust enough to run most of the testbenches implemented inside this framework[23].

5. Software and Platforms

5.1. Gaspard 2

Participants: Jean-Luc Dekeyser [correspondant], All Dart Team.

Gaspard2 is an Integrated Development Environment (IDE) for SoC visual co-modeling. It allows or will allow modeling, simulation, testing and code generation of SoC applications and hardware architectures. Its purpose is to provide a single environment for all the SoC development processes:

- High level modeling of applications and hardware architectures
- Application and hardware architecture association (mapping and scheduling)
- Application refactoring

- Deployment specification
- Model to model transformation (to automatically produce models for several target platforms)
- Code generation
- Simulation
- Reification of any stages of the development

The Gaspard2 tool is based on the Eclipse [35] IDE. A set of plugins provides the different functionalities. Gaspard2 provides an internal engine to execute transformation chains. This engine is able to run either QVT (OMG standard) or Java transformations. It is also able to run model-to-text transformations based on Acceleo [37]. The Gaspard2 engine is defined to execute models conform to an internal transformation chains meta-model. A GUI has been developed to specify transformation chain models by drawing them. For the final user, application, hardware architecture, association, deployment and technology models are specified and manipulated by the developer through UML diagrams, and saved by the UML tool in an XMI file format. Gaspard2 manipulates these models through repositories (Java interfaces and implementations) automatically generated thanks to the Ecore specification. Several transformation chains are provided with Gaspard2 to target, from UML models, several execution or simulation platforms (OpenMP, OpenCL, Pthread, SystemC, VHDL, ...). This input language is based on the MARTE UML profile. A tool to generate SIMD configurations derived from the mppSoC model was developed. It allows to automatically generate the VHDL code from a high specification modeled at a high abstraction level (UML model using MARTE profile) based on the IP mppSoC library. The developed tool facilitates to the user to choose a SIMD configuration adapted to his application needs. It has been integrated in the Gaspard environment. **Gaspard2 as an educational resource.** The Gaspard2 platform was one of the topics taught in the context of the courses on embedded systems in Telecom Lille and in a Master 2 (TNSI) lecture " Design tools for embedded systems" at the University of Valenciennes. These lectures focused on the potentiality to generate several targets from a subset of the Marte profile and the ability to target system on chip architectures at the TLM level respectively. Furthermore, the model driven engineering characteristics of Gaspard2 are largely detailed in the lecture of Software engineering at Polytech Lille and in the Master of research at university of Lille too.

- See also the web page http://www.gaspard2.org/
- Inria softwre evaluation: A-2, SO-4, SM-2, EM-1, SDL-2, DA-4, CD-4, MS-4, TPM4
- Version: 2.1.0

6. New Results

6.1. Hardware Distributed Control for Dynamic Reconfigurable Systems

The progress in FPGA technology has allowed FPGA-based reconfigurable embedded systems to target increasingly sophisticated applications, which leads to a high design complexity of such systems especially at the adaptation control level. This complexity results into long design phases and delayed time-to-market. In this context, a centralized control model might be not adapted to the growing size and complexity of embedded systems. The use of a single controller for the whole system might result into a high complexity due to the number of parameters to take into account for runtime adaptation, which makes difficult its modification and test. Besides, the design of such a controller is system-dependent since it treats the system as a whole, which represents an obstacle for design reuse. In order to solve these problems, we propose a control design approach aiming to decrease design complexity and enhance design flexibility, reuse and productivity. This approach is based on a semi-distributed control model [34]. In order to achieve the objectives mentioned above, the proposed approach combines autonomy, modularity, formalism and high-level design. The semi-distributed control model divides the control problem between autonomous controllers handling each the self-adaptation of a reconfigurable component of the system, which allows to decrease their design complexity. Each controller handles three main tasks allocated to three different modules: i)monitoring of events that might trigger the adaptation of the controlled component, ii)decision-making about the required adaptations, and

iii)adaptation (reconfiguration) realization. To ensure that reconfiguration decisions made by the controllers respect global system constraints such as security and quality of service constraints, these decisions are coordinated before launching the corresponding partial reconfigurations. The allocation of these tasks to separate modules facilitates their modification and reuse and thus the scalability of the control design. For the decision-making modeling, we use the mode-automata formalism. This formalism is suitable to model the control of the different modes of a reconfigurable system such as energy modes or image display modes. Thanks to its clear semantics, the use of such a formalism facilitates the high-level modeling of the controllers and their automatic generation. In order to facilitate code generation and enhance thus design productivity, our control approach makes use of Model-Driven-Engineering (MDE) [33]. Control systems composed of controllers and coordinators are modeled using the UML (Unified Modeling Language) profile MARTE (Modeling and Analysis of Real-Time and Embedded systems). The automation of MDE, allowed to generate the code of these systems. The generated code was then used to validate the semi-distributed control and to determine its resource overhead compared to centralized control systems.

6.2. Regular interconnection network for HP-SoC architecture

Our Synchronous Communication Asynchronous Computation (SCAC) model is a data-parallel execution model dedicated to the High Performance System-on-Chip. The architecture of this model is composed of huge number of complex routers, called node elements (the NEs), communicating and working in perfect synchronizations. Each NE is potentially connected to its neighbors via a regular connection. Furthermore, each NE is connected to a heterogeneous set of computing groups (clusters) allow asynchronous processing. Each group includes a combination of processors programmable, the PEs (software processing units) and specialized hardware accelerators (hardware processing units) to perform critical tasks demanding the more performance. All the system is controlled by a Network Controller Unit, the NCU. The NCU and The PEs are implemented with the Forth processor.

The synchronous communication in SCAC model is presented by two kinds of communications:

- The NCU/NEs communication. In fact, we defined a hNoC model integrated in the SCAC architecture [31]. This model is based on sub-netting the network of processing nodes which separate the control of communication and processing. From this model, our communication system allows a better management of data congestion in the NEs grid through the broadcast with mask of parallel instructions to activated processing nodes.
- The NE/NE communication which is our last contribution. In fact, we defined the X-net interconnection network which is a regular network dedicated to the massively parallel SCAC architecture. This network interconnects directly each PE with its 8 nearest neighbors in a two-dimensional mesh through a specific router in the NE module.

The aim of these last works is to design a regular NoC for SCAC architecture to allow global synchronization of the system communications and increase high performance in terms of area cost and bandwidth. This network based on IP blocks which offer well flexibility and scalability, was implemented in synthesizable VHDL code that was simulated and targeted Xilinx Virtex6 (XC6VLX240T) board. The difficulty of designing X-net is a compromise between an optimal quality of broadcasting, high bandwidth and important flexibility of use, while reducing power consumption and silicon area.

6.3. ReCoMARTE: A Marte Based Profile for Dynamic Reconfigurable Systems Modeling

During the last decade, DPR has been widely studied as a research topic. Despite its intuitive appeal, the technique had eluded widespread adoption, particularly in industrial applications. This is due to the complexities of the provided design flow and the in-depth knowledge of many low level aspects of FPGA technologies used to implement DPR systems. The aim of our current work is to propose a methodology in order to allow us to introduce PDR in MARTE for modeling all types of FPGAs supporting our chosen PDR flow. Afterwards, using the MDE model transformations, the design flow can be used to bridge the gap

between high level specifications and low implementation details to finally generate files used by the Xilinx EDK design flow for implementing the top-level SoC description of the system. Indeed, in its current version, UML MARTE profile lacks dynamic reconfiguration concepts and requirements for the reconfiguration at different abstraction levels. We have concentrated our efforts in the creation of the structural description of the system that is used as an input to the DPR design flow to facilitate the design entry phase of the DPR design flow. Therefore, we defined an extended version of MARTE called RecoMARTE (Reconfigurable MARTE) [16] model these concepts mainly at:

- Application level: For reconfigurable applications combining control and data processing, it is very difficult, even impossible to use the MARTE profile for their specification. Non-functional properties such as control concepts are induced by different configurations or running modes of the system and allow the description of more complex behaviours. We recommend a set of extensions to a MARTE profile. We also focus on modelling heterogeneous reconfigurable components, and address the problem of constraints specification for verification issue.
- Control mechanism: We define necessary requirements for the reconfiguration control mechanism in order to manage reconfiguration at every design level. In addition, our solution allows to describe global contracts and constraints for combining automata. Our modeled reconfiguration controller will be then synthesized using Discrete Controller Synthesis formal technique (collaboration work)I n order to always provide a correct configuration to the system, with respect to constraints specified by the designer
- Deployment level: Our design methodology using RecoMARTE enables the deployment, parameterization and integration of hardware IPs into SoC platform at multiple levels of abstraction. We have introduced IP deployment capabilities in MARTE, which aim at facilitating the import of selected low-level features into the high-level models, their modification, and the creation of an IP-XACT design description that is used to parameterize and integrate the underlying IP descriptions.
- Physical level: introduced extensions in MARTE provide some facilities to allow modeling physical architecture of a chosen FPGA. Our solution allows to carry out the physical placement of static and reconfigurable areas on the platform. This task is done through ranges in terms of physical resources, with respect to placement constraints such as consumed resources.

6.4. Using Marte Profile for NoCs modeling

The modeling of repetitive structures such as network on chip topologies in graphics forms poses a particular challenge. This aspect may be encountered in intensive data/control oriented applications such as H.264 video coder. In this work we have described an adequate methodology for modeling NoCs by using the MARTE standard profile. The proposed study has shown that the Repetitive Structure Modeling (RSM) package of MARTE profile is powerful enough for modeling different topologies. By using this methodology, several aspects such as routing algorithm are modeled based finite state machines. This allows to the MARTE profile to be complete enough for modeling a large number of NoCs architectures. Some work is on-going to synthesize such networks in VHDL from such models [55]. While validating the proposed methodology, a co-design approach has been studied by mapping a H264 video coding system onto a Diagonal Mesh NoC by using the Y Chart of Gaspard2 tool. Before allowing the association of the application/architecture, an architectural optimization targeting power minimization of the most critical module of the application and the router of the architecture has been performed. For instance, a flexible VLSI architecture for full-search VBSME (FSVBSME) has been proposed.

6.5. A Hardware Membranes Based Reconfiguration Services Implementation

Partial and dynamic reconfiguration provides a relevant new dimension to design efficient parallel embedded systems. However, due to the encasing complexity of such systems, ensuring the consistency and parallelism management at runtime is still a key challenge. So architecture models and design methodology are required to allow for efficient component reuse and hardware reconfiguration management. We proposed a novel approach

inspired from the well-known component based models used in software applications development. Our model is based on membranes wrapping the systems components. The objective is to improve design productivity and ensure consistency by managing context switching and storage using modular distributed hardware controllers. These membranes are distributed and optimized with the aim to design self-adaptive systems by allowing dynamic changes in parallelism degree and contexts migration [26]. These results are obtained in the Famous project by a collaboration with LABSticc Lorient.

6.6. Formal Techniques for General and Domain-Specific Languages

In 2012 we have finished the previous year's activities on domain-specific languages based on formal model-driven engineering with two papers [18], [24]. Our conclusion is that formal MDE-based language definition is interesting because of its generality but adds extra layers of complexity due to the fact that language concepts and semantics are only formalised indirectly, through the formalisation of MDE concepts used in language definition. We have decided thus to move on towards more direct ways of defining and reasoning about languages. We have been experimenting with the K framework [4] for formally defining both the assembly language and a higher-level language for programming on the upcoming dynamically reconfigurable hardware architecture that our team is developing.

We have also worked on proving the correctness of a compiler between high-level and assembly language, based on new symbolic program-equivalence proof techniques we are developping in collaboration with the K team [29]. We have also been working on generic symbolic execution techniques for programming languages having term-rewriting based semantics [28] (PhD of Andrei Arusoaie, supervised in collaboration with Prof. Dorel Lucanu from the K team of Univ. Iasi (Romania).

7. Bilateral Contracts and Grants with Industry

7.1. Collaboration EADS IW, and Eurocopter

The subject deals with dynamic reconfigurable system design for avionic test applications. It is motivated by the need of methodologies and tools for the design of high-performance applications on dynamic reconfigurable computing systems. A complete methodology takes the reconfigurability of the hardware as an essential design concept and proposes the necessary mechanisms to fully exploit those capabilities at runtime. A set of tools must provide high-quality designs with improved designer productivity, which guarantees consistency with the initial requirements for adaptability and for the final implementation. This methodology allows designers to easily implement a system specification on a platform that includes general purpose processors dynamically combined with multiple accelerators running on an FPGA.

7.2. National Initiatives

7.2.1. ANR

7.2.1.1. ANR Famous

Collaboration with Inria Rhône Aples, Université de Bretagne Sud, Université de Bourgogne, SME SODIUS

[4] http://www.k-framework.org

FAMOUS project aims at introducing a complete methodology that takes the reconfigurability of the hardware as an essential design concept and proposes the necessary mechanisms to fully exploit those capabilities at runtime. The project covers research in system models, compile time and run time methods, and analysis and verification techniques. These tools will provide high-quality designs with improved designer productivity, while guaranteeing consistency with the initial requirements for adaptability and the final implementation. Thus FAMOUS is a research project with an immediate industrial impact. Actually, it will make reconfigurable systems design easier and faster. The obtained tool in this project is expected to be used by both companies designers and academic researchers, especially for modern applications system specific design as smart camera, image and video processing, FAMOUS tools will be based on well established standards in design community. In fact, modeling will start from very high abstraction level using an extended version of MARTE. Simulation and synthesizable models will be obtained by automatic model to model transformations, using MDE approach. These techniques will contribute to shorten drastically time-to-market. FAMOUS is a basic research project. In fact, most of partners are academic, and its main objective is to explore novel design methodologies and target modern embedded systems architectures. FAMOUS project is funded by french Agence Nationale de la Recherche (ANR). It has also been labeled by Media & Network cluster in 2009. The involved resources reach 408 person-month, from five partners: the public research labs LIFL Inria (Lille), LabSTICC (Lorient), Inria Rhône-Alpes (Grenoble), LE2I University of Bourgogne (Dijon) and the SME company Sodius SAS (Nantes). It has started on December 2009, and it will last 48 months.

7.2.1.2. The ANR Open-People project

Partners: Université de Bretagne Sud (UBS)Lab-STICC, Inria Nancy Grand Est, Inria Lille Nord Europe, Université de Rennes 1 (UR1), Université de Nice Sophia Antipolis (UNSA), THALES Communications (Colombes), InPixal (Rennes)

The Open-PEOPLE (Open Power and Energy Optimization PLatform and Estimator project is a national project funded by the ANR (Agence Nationale de la Recherche), the French National Research Agency. The objective of Open-PEOPLE is to provide a platform for estimating and optimizing the power and energy consumptions. Users will be able to estimate the consumption of an application deployed on a hardware architecture chosen in a set of parametric reference architectures. The components used in the targeted architecture will be chosen in a library of hardware and software components. Some of these components will be parametric (such as reconfigurable processors or ASIP) to further enlarge the design space for exploration. The library will be extensible; users will have the possibility to add new components, according to the evolution of both applications and technology. Open-PEOPLE is definitely an open project. The software platform for conducting estimation and optimization, will be accessible through an Internet portal. This software platform will be coupled to an automated hardware platform for physical measurements. The measurements needed to build models for new components to be added in the library will be remotely controlled through the software platform. A library of benchmarks will be proposed, to help building models for new components and architectures.

7.2.2. Competitivity Clusters

We collaborate with the L2EP (Université de Lille1) inside the research pole MEDEE, especially in the first action: industrialization of Code_CARMEL.

7.2.3. Within Inria

We collaborate with colleagues within Inria with the Triskell team at Inria Rennes-Bretagne Atlantique) on the analysis of DSMLs and on the formal definition of Kermeta.

8. Partnerships and Cooperations

8.1. European Initiatives

8.1.1. Collaboration with Romania

We collaborate with the University of Laşi (Romania) on formal techniques for general and domain specific languages.

8.1.2. Collaboration with the Netherlands

We collaborate with the Eindhoven University of Technology (The Netherlands) on formal techniques for general and domain specific languages.

8.2. International Research Visitors

8.2.1. Visits of International Scientists

Tim Willemse

Subject: visit to explore future collaborations.

Institution: Eindhoven University of Technology, NL

Duration: 1 week

Frank Stappers

Subject: formal verification for reconfigurable languages

Institution: Eindhoven University of Technology, NL

Duration: 6 weeks

8.2.1.1. Internships

Bram Gerron

Subject: formal verification of compilation

Institution: Eindhoven University of Technology, NL

Duration: 3 months

9. Dissemination

9.1. Scientific Animation

Samy Meftali was member of the program committee of:

- Conference on Design and Architectures for Signal and Image Processing. DASIP 2012. October 23-25, 2012. Karlsruhe, Germany
- International Conference on EMBEDDED SYSTEMS in TELECOMMUNICATIONS and INSTRUMENTATION, 5-7 novembre 2012, Annaba. Algeria.
- The International Symposium on System-on-Chip Tampere, Finland. October 11-12, 2012.

Vlad Rusu was member of the program committee of the IFIP Joint International Conference on Formal Techniques for Distributed Systems (FMOODS & FORTE 2012).

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Licence : Frédéric Guyomarch, Algorithmics and programming, 70h, L1, Université Lille 1, France

Licence : Frédéric Guyomarch, Data structures, 50h, L1, Université Lille 1, France

Licence : Frédéric Guyomarch, Architecture, 30h, L1, Université Lille 1, France

Licence : Frédéric Guyomarch, Graph theory, 40h, L2, Université Lille 1, France

Master : Frédéric Guyomarch, Advanced achitectures, 30h, M1, Université Lille 1, France

Licence : Samy Meftali, Architectures élémentaires, 90h eqTD, L2, Université Lille 1, France

Licence : Samy Meftali, Architecture, 40h, L3, Université Lille 1, France

Master :Samy Meftali, Architectures évoluées, 68h, M1, Université Lille 1, France

Licence : Philippe Marquet, Introduction to Computer Science, 15h, Secondary Education Teatcher Training, Université Lille 1, France

Licence : Philippe Marquet, System Programming, 60h, L3, Université Lille 1, France

Master: Philippe Marquet, Design of Operating System, 42h, M1, Université Lille 1, France

Master: Philippe Marquet, Web of Things: Embedded System Programming, 20h, M1, Université Lille 1, France

Master: Philippe Marquet, Parallel and Distributed Programming, 24h, M1, Université Lille 1, France

Master: Philippe Marquet, Introduction to Innovation and Research, 15h, M2, Université Lille 1, France

Licence : Vlad Rusu, Architectures élémentaires, 30h, L2, Université Lille 1, France

Master : Vlad Rusu, Introduction to Formal Verification with PVS, 20h, M2, Universidad Complutense de Madrid, Espagne

9.2.2. Supervision

PhD : Wendell Rodrigues, *Une Méthodologie pour le Développement d'Applications Hautes Performances sur des Architectures GPGPU: Application à la Simulation des Machines Éléctriques*, Université de Lille 1, 26-01-2012, Frédéric Guyomarch and Jean-Luc Dekeyser

9.2.3. Juries

Frédéric Guyomarch was the external examiner for Jing Guo's PhD (University of Hertfordshire, UK). entitled *Fully Automated Transformation of Hardware-Agnostic, Data-Parallel Programs for Host-Driven Executions on GPUs*.

Vlad Rusu was on the PhD commitee of Michaël Lauer (Université de Toulouse) entitled *Une méthode globale pour la vérification d'exigences temps réel Application à l'Avionique Modulaire Intégrée*

Samy Meftali was on the PhD commitee of Yaset OLIVA VENEGAS(INSA Rennes) entitled *High Level Modeling of Run-Time Managers for the Design of Heterogeneous Embedded Systems*

Samy Meftali was on the PhD commitee of Amine Anane(Univ. Montréal) entitled *Application du concept des transactions pour la modélisation et la simulation multicoeur des systèmes sur puce*

9.3. Popularization

Pamela Wattebled is chair of the organizing comitee of Majecstic 2012. MajecSTIC 2012 is the 9th edition of the conference MajecSTIC. This conference is organized by young researchers. This year it takes place in Lille and organized by Pamela WATTEBLED-MEFTALI (project-team DART), Nicolas GOUVY (project-team FUN) and Adel NOUREDDINE (project-team ADAM).

10. Bibliography

Major publications by the team in recent years

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

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

[3] A. W. DE OLIVEIRA RODRIGUES, F. GUYOMARC'H, J.-L. DEKEYSER, Y. LE MENACH. *Automatic Multi-GPU Code Generation applied to Simulation of Electrical Machines*, in "Compumag 2011", Sydney, Australia, July 2011, http://hal.inria.fr/inria-00605645/en

[4] A. GAMATIÉ, É. RUTTEN, H. YU, P. BOULET, J.-L. DEKEYSER. *Synchronous Modeling and Analysis of Data Intensive Applications*, in "EURASIP Journal on Embedded Systems,eurasip", july 2008, vol. 2008, Article ID 561863

[5] A. GAMATIÉ. *Design of Streaming Applications on MPSoCs using Abstract Clocks*, in "Design, Automation and Test in Europe Conference (DATE'2012)", Dresden, Allemagne, 2012, http://hal.inria.fr/hal-00647480/en/

[6] A. GAMATIÉ, S. LE BEUX, É. PIEL, R. BEN ATITALLAH, A. ETIEN, P. MARQUET, J.-L. DEKEYSER. *A Model Driven Design Framework for Massively Parallel Embedded Systems*, in "ACM Transactions on Embedded Computing Systems (TECS)", 2011, vol. 10, n^{o} 4, http://hal.inria.fr/inria-00637595/en

[7] C. GLITIA, P. BOULET, E. LENORMAND, M. BARRETEAU. *Repetitive model refactoring strategy for the design space exploration of intensive signal processing applications*, in "Journal of Systems Architecture", January 2011, vol. 57, n^{o} 9, pp. 815-829 [*DOI :* 10.1016/J.SYSARC.2010.12.002], http://hal.inria.fr/inria-00605069/en

[8] S. LE BEUX, P. MARQUET, J.-L. DEKEYSER. *A Model Driven Co-Design Approach for High Perforamnce Embedded Systems Dedicated to Transport*, in "Studies in Informatics and Control Journal", 2008, vol. 2008, n^{o} 4

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

[10] SANTHOSH KUMAR. RETHINAGIRI, R. BEN ATITALLAH, S. NIAR, E. SENN, J.-L. DEKEYSER. *Hybrid System Level Power Consumption Estimation for 29FPGA-Based MPSoC*, in "29th IEEE International Conference on Computer Design ICCD 2011", October 2011

[11] V. RUSU. *Embedding Domain-Specific Modelling Languages in Maude Specifications*, in "ACM SIGSOFT Software Engineering Notes", January 2011, vol. 36, n^{o} 1, Extended version accepted in the Systems and Software Engineering Journal. [*DOI :* 10.1145/1921532.1921557], http://hal.inria.fr/inria-00527859/en/

[12] C. TRABELSI, S. MEFTALI, R. BEN ATITALLAH, A. JEMAI, J.-L. DEKEYSER, S. NIAR. *An MDE Approach for Energy Consumption Estimation in MPSoC Design*, in "2nd Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools", Italie Pisa, Jan 2010, 6 p. p. , http://hal.inria.fr/inria-00486200/en

Publications of the year

Articles in International Peer-Reviewed Journals

[13] A. W. DE OLIVEIRA RODRIGUES, F. GUYOMARC'H, J.-L. DEKEYSER, Y. LE MENACH. *Automatic Multi-GPU Code Generation Applied to Simulation of Electrical Machines*, in "Magnetics, IEEE Transactions on", February 2012, vol. 48, n^{o} 2, pp. 831 - 834 [*DOI :* 10.1109/TMAG.2011.2179527], http://hal.inria.fr/hal-00670150

[14] P. DEVIENNE, A. AMMAR. *Extended Model Driven Architecture to B Method*, in "Ubiquitous Computing and Communication Journal", 2012, http://hal.inria.fr/hal-00641131

[15] C. GLITIA, J. DEANTONI, F. MALLET, J.-V. MILLO, P. BOULET, A. GAMATIÉ. *Progressive and explicit refinement of scheduling for multidimensional data-flow applications using uml marte*, in "Design Automation for Embedded Systems", 2012, vol. 16, n^{o} 2, pp. 137-169 [*DOI :* 10.1007/S10617-012-9093-Y], http://hal.inria.fr/hal-00727239

[16] G. OCHOA-RUIZ, O. LABBANI, E.-B. BOURENNANE, P. SOULARD, S. CHERIF. *A High-level Methodology for Automatically Generating Dynamic Partially Reconfigurable Systems using IP-XACT and the UML MARTE Profile.*, in "Design Automation for Embedded Systems", December 2012, http://hal.inria.fr/hal-00745377

[17] I. R. QUADRI, A. GAMATIÉ, P. BOULET, S. MEFTALI, J.-L. DEKEYSER. *Expressing embedded systems configurations at high abstraction levels with UML MARTE profile: advantages, limitations and alternatives*, in "Journal of Systems Architecture", 2012 [*DOI :* 10.1016/J.SYSARC.2012.01.001], http://hal.inria.fr/hal-00666014

[18] V. RUSU. *Embedding domain-specific modeling languages into Maude specifications*, in "Software and Systems Modeling", 2012, To appear [*DOI :* 10.1007/S10270-012-0232-5], http://hal.inria.fr/hal-00660104

International Conferences with Proceedings

[19] A. GAMATIÉ. *Design of Streaming Applications on MPSoCs using Abstract Clocks*, in "Design, Automation and Test in Europe Conference (DATE'2012)", Dresden, Allemagne, 2012, http://hal.inria.fr/hal-00647480/en/

[20] S. KUMAR RETHINAGIRI, R. BEN ATITALLAH, J.-L. DEKEYSER. *Virtual Platform for Embedded System Power Estimation*, in "DATE-2012", Dresden, Germany, Rainer Leupers, March 2012, http://hal.inria.fr/hal-00673911

[21] S. KUMAR RETHINAGIRI, R. BEN ATITALLAH, J.-L. DEKEYSER. *Virtual Platform for Embedded System Power Estimation*, in "QVVP'12 (conjunction with DATE)", Dresden, Germany, Rainer Leupers, March 2012, http://hal.inria.fr/hal-00673917

[22] S. K. RETHINAGIRI, R. BEN ATITALLAH, J.-L. DEKEYSER, S. NIAR, E. SENN. *An Efficient Power Estimation Methodology for Complex RISC Processor-based Platforms*, in "GLSVLSI 2012", Salt lake city, UTAH, United States, J. CAVALLARO (editor), ACM, May 2012, http://hal.inria.fr/hal-00675469

[23] W. RODRIGUES, L. CHEVALIER, Y. LE MENACH, F. GUYOMARCH. *Test Harness on a Preconditioned Conjugate Gradient Solver on GPUs: An Efficiency Analysis*, in "CEFC - 2012", Oita, Japon, November 2012, http://hal.inria.fr/hal-00759476

[24] V. RUSU, D. LUCANU. *A K-Based Formal Framework for Domain-Specific Modelling Languages*, in "Formal Verification of Object-Oriented Systems", Torino, Italy, Lecture Notes in Computer Science, Springer Verlag, 2012, vol. 7421, pp. 214-231 [*DOI :* 10.1007/978-3-642-31762-0], http://hal.inria.fr/inria-00637099

[25] A. TINZEFTE, Y. LE MENACH, F. GUYOMARCH. *A New Preconditionner Based on F.I.T Applied To Solve F.E.M Problem*, in "CEFC - 2012", Iota, Japon, November 2012, http://hal.inria.fr/hal-00759484

[26] P. WATTEBLED, J.-P. DIGUET, J.-L. DEKEYSER. *Membrane-based design and management methodology for parallel dynamically reconfigurable embedded systems*, in "RecoSoc 2012", YORK, United Kingdom, July 2012, http://hal.inria.fr/hal-00745150

Scientific Books (or Scientific Book chapters)

[27] J.-L. DEKEYSER, A. GAMATIÉ, S. MEFTALI, I. R. QUADRI. *Models for Co-Design of Heterogeneous Dynamically Reconfigurable SoCs*, in "Heterogeneous Embedded Systems - Design Theory and Practice", G. NICOLESCU, I. O'CONNOR, C. PIGUET (editors), Springer, 2012, 26 p. p. , http://hal.inria.fr/inria-00525023

Research Reports

[28] A. ARUSOAIE, D. LUCANU, V. RUSU. , *A Generic Approach to Symbolic Execution*, Inria, December 2012, n^{o} RR-8189, 21 p. , http://hal.inria.fr/hal-00766220

[29] D. LUCANU, V. RUSU. , *Program Equivalence by Circular Reasoning*, Inria, October 2012, n^{o} RR-8116, 26 p. , http://hal.inria.fr/hal-00744374

Other Publications

[30] G. AFONSO, R. BEN ATITALLAH, J.-L. DEKEYSER. , *Software Implementation vs. Hardware Implementation: The Avionic Test System Case-Study*, March 2012, http://hal.inria.fr/hal-00665162

[31] H. KRICHENE, M. BAKLOUTI, M. ABID, P. MARQUET, J.-L. DEKEYSER. , *Broadcast with mask on a Massively Parallel Processing on a Chip*, 2012, DRNoC, http://hal.inria.fr/hal-00688309

[32] H. KRICHENE, M. BAKLOUTI, M. ABID, P. MARQUET, J.-L. DEKEYSER. , *Broadcast with mask on a Massively Parallel Processing on a Chip*, 2012, workshop drnoc2012, http://hal.inria.fr/hal-00688418

[33] C. TRABELSI, S. MEFTALI, J.-L. DEKEYSER. , *Distributed control for reconfigurable FPGA systems: a high-level design approach*, June 2012, Workshop Recosoc 2012, http://hal.inria.fr/hal-00709755

[34] C. TRABELSI, S. MEFTALI, J.-L. DEKEYSER. , *Semi-distributed control for FPGA-based reconfigurable systems*, September 2012, Conférence DSD Turquie, http://hal.inria.fr/hal-00703093

References in notes

[35] , *The Eclipse Project*, 2003, http://www.eclipse.org

[36] , *EMF Eclipse Modeling Framework*, 2007, http://www.eclipse.org/modeling/emf

[37] , *Acceleo*, 2009, http://www.acceleo.org

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

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

[40] , *SystemC*, 2002, http://www.systemc.org/

[41] , *OpenMP Application Programme Interface*, May 2005, http://www.openmp.org/drupal/mp-documents/spec25.pdf

[42] A. AGRAWAL. *Graph Rewriting And Transformation (GReAT): A Solution For The Model Integrated Computing Bottleneck*, in "18th IEEE International Conference on Automated Software Engineering (ASE'03)", 2003, pp. 364-368

[43] V. ARANEGA, A. ETIEN, J.-L. DEKEYSER. *Using an Alternative Trace for QVT*, in "Workshop on Multi-Paradigm Modeling", Norvège Olso, Oct 2010, http://hal.inria.fr/inria-00524153/en

[44] V. ARANEGA, J.-M. MOTTU, A. ETIEN, J.-L. DEKEYSER. *Traceability Mechanism for Error Localization in Model Transformation*, in "ICSOFT", Bulgaria, July 2009

[45] V. ARANEGA, J.-M. MOTTU, A. ETIEN, J.-L. DEKEYSER. *Using Traceability to Enhance Mutation Analysis Dedicated to Model Transformation*, in "Workshop on Model driven Engineering Verification and Validation", Norvège Olso, Oct 2010, http://hal.inria.fr/inria-00524150/en

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

[47] P. BOULET. , *Array-OL Revisited, Multidimensional Intensive Signal Processing Specification*, Inria, February 2007, n^{o} RR-6113, 24 p. , http://hal.inria.fr/inria-00128840/en

[48] 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, pp. 105–112

[49] T. BUCHMANN, A. DOTOR, S. UHRIG, B. WESTFECHTEL. *Model-Driven Software Development with Graph Transformations: A Comparative Case Study*, in "Applications of Graph Transformations with Industrial Relevance, Third International Symposium (AGTIVE'07)", 2007, pp. 345-360

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

[51] G. CSERTÁN, G. HUSZERL, I. MAJZIK, Z. PAP, A. PATARICZA, D. VARRÓ. *VIATRA - Visual Automated Transformations for Formal Verification and Validation of UML Models*, in "17th IEEE International Conference on Automated Software Engineering (ASE'02)", 2002, pp. 267-270

[52] 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

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

[54] M. EGEA, V. RUSU. *Formal executable semantics for conformance in the MDE framework*, in "Innivations in Software and Systems Engineering", 2010, http://hal.inria.fr/inria-00527502/en

[55] M. ELHAJI, P. BOULET, S. MEFTALI, A. ZITOUNI, J.-L. DEKEYSER, R. TOURKI. *An MDE approach for modeling network on chip topologies*, in "Design and Technology of Integrated Systems in Nanoscale Era (DTIS), 2010 5th International Conference on", Tunisie Hammamet, 2010, http://hal.inria.fr/inria-00526629/en

[56] A. ETIEN, A. MULLER, T. LEGRAND, X. BLANC. *Combining Independent Model Transformations*, in "ACM Symposium On Applied Computing (SAC)", Suisse Sierre, Mar 2010, http://hal.inria.fr/inria-00516708/en

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

[58] I. GALVÃO, A. GOKNIL. *Survey of Traceability Approaches in Model-Driven Engineering*, in "IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)", October 2007, pp. 313-326

[59] F. JOUAULT. *Loosely Coupled Traceability for ATL*, in "European Conference on Model Driven Architecture (ECMDA) workshop on traceability", 2005, pp. 29–37

[60] S. LE BEUX, P. MARQUET, J.-L. DEKEYSER. , *Model Driven Engineering Benefits for High Level Synthesis*, Inria, 2008, n^{o} 6615, http://hal.inria.fr/inria-00311300/en/

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

[62] J.-M. MOTTU, B. BAUDRY, Y. LE TRAON. *Mutation Analysis Testing for Model Transformations*, in "proceedings of the European Conference on Model Driven Architecture (ECMDA 06)", Bilbao, Spain, July 2006

[63] OBJECT MANAGEMENT GROUP, INC.. , *Meta Object Facility (MOF) Core Specification, Version 2.0*, January 2006, http://www.omg.org

[64] OBJECT MANAGEMENT GROUP, INC.. , *MOF Query / Views / Transformations*, July 2007, OMG paper, http://www.omg.org

[65] J. RIVERA, E. GUERRA, J. DE LARA, A. VALLECILLO.. *Analyzing Rule-Based Behavioral Semantics of Visual Modeling Languages with Maude*, in "Proc. 1st International Conference on Software Language Engineering (SLE'08)", LNCS, 2008, vol. 5452, pp. 54-73

[66] V. RUSU, L. GONNORD, B. COMBEMALE. , *Formally Tracing Executions From an Analysis Tool Back to a Domain Specific Modeling Language's Operational Semantics*, Inria, Oct 2010, n^{o} RR-7423, http://hal.inria.fr/inria-00526561/en

[67] V. RUSU. *Embedding Domain-Specific Modelling Languages in Maude Specifications*, in "ACM SIGSOFT Software Engineering Notes", 2010, http://hal.inria.fr/inria-00527859/en

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

[69] G. TAENTZER. *AGG: A Graph Transformation Environment for Modeling and Validation of Software*, in "Applications of Graph Transformations with Industrial Relevance, Second International Workshop (AGTIVE'03)", 2003, pp. 446-453

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

[71] Y. VELEGRAKIS, R. J. MILLER, J. MYLOPOULOS. *Representing and Quering Data transformations*, in "International conference on Data Engineering (ICDE)", April 2005, pp. 81-92