Using Model Driven Engineering and UML/MARTE for HW/SW partitioning

Yessine Hadj Kacem, Adel Mahfoudhi, Walid Karamti, Mohamed Abid

***Abstract*—In most Hardware Software co-design methodologies, the distribution of a parallel application onto a heterogeneous computing resource represents a major challenge due to the complexity of the Embedded Real Time Systems (ERTS). Hence, powerful tools that can handle both scheduling and performance analyses are required. Most of the research in this area focuses on high abstraction level methods to decrease the design convolution. In particular, the Unified Modeling Language (UML) profiles and the Model Driven Engineering (MDE) aim at being an adequate solution to support the whole life cycle co-design of complex ERTS with their real time constraints and performance issues. Unfortunately, partitioning problems which present a key item of a multi processor system design are not well tackled using techniques founded on this high level abstraction. In this paper, an MDE integrated approach relying on the recent profile Modeling and Analysis of Real-Time and Embedded systems (MARTE) for design space exploration is proposed. Following refinement strategy, transformation rules allow the finding of a feasible schedule that satisfies timing constraints and defining where tasks will be implemented. A major impetus behind both the early and the later analyses of schedulability and performance is the separation of concerns. Indeed, a fast and guided partitioning strategy starting from high level design have been integrated.**

***Index Terms*—UML, MARTE profile, MDE, ERTS, design space exploration, scheduling analysis, performance analysis.**

I. Introduction

The design and the implementation of real time embedded systems is a difficult engineering task. It requires the verification of system properties, particularly, real time and precedence constraints. The goal of the design phase is to map the given system specification to hardware and software architecture, which has always been a challenge. Thus, standards to facilitate the asignement of a parallel application onto a heterogeneous architecture and the checking of system properties at a preliminary stage are progressing well, based on different abstraction layers.

During the last decade, the modeling and simulation of such systems have been tackled with model driven engineering (MDE) approach [28]. In this context, the Unified Modeling Language (UML) represents a viable solution to decrease the complexity of Embedded Real Time System (ERTS) design via UML profiles. Its objective is to become an adequate solution to support the whole life cycle co-design of complex ERTS with their real time constraints and performance issues. They have been adopted for representing different system views with their functional and non-functional properties. In this context, the recent profile Modeling and Analysis of Real-Time and Embedded systems (MARTE) [13] comes to upgrade the profile Schedulability, Performance, and Time (SPT) [10] and the profile Quality of Service and Fault Tolerance (QoS&FT) [12]. It provides not only a rich set of terminology for specifying and analyzing ERTS but also a vast notation for modeling the distribution of a parallel application onto a parallel architecture.

However, one of the major challenges in the multi-processor system design is to guarantee a high performance and a low cost of the final implementation. Thus, the assignment phase of system parts to mixed implementation units as processors and memories aims at performing a design implementation that satisfies all functional and non-functional requirements at a minimum cost.

Unfortunately, partitioning problems [33] which present a key item of multi processor system design are not well tackled using techniques founded on high level abstraction. Traditionally, the design of these systems is limited to the characterization of the architecture and application. Indeed, the designer relies on his own knowledge or on the mapping to external simulation tools in order to define where tasks are implemented and run. Moreover, it is mandatory to schedule the system's tasks to meet any timing constraints and estimate their execution time of tasks. It is also obligatory to evaluate the quality of a given solution and to verify whether design constraints are met.

Since the main goals of co-design are not included in the usual practice of UML profiles, automated or guided partitioning methods are required. In order to overcome this gap, the following open issues need to be addressed:

- to support a rich and pre-characterized framework of software and hardware components
- to support separation of concerns in order to allow application specification, abstract architecture description and possible tasks implementation
- to support the property-preserving transformations during refinement from requirement specification to concrete deployment
- to support a fast and guided Design Space Exploration (DSE) starting from high level design that can be integrated in MDE process
- to support system determinism, i.e., whether or not it will meet its performance and schedulability requirements.

This paper presents an early design space exploration method for real time systems. The proposed approach adopts the MDE concept based on UML/MARTE models and rule trans-

Yessine Hadj kacem, Adel Mahfoudhi, Walid Karamti and Mohamed Abid are with the Computer and Embedded System (ces) Laboratory, National Engineering School of Sfax,B.P.:w 3038, Sfax Tunisia e-mail: (see http://www.ceslab.org).

formation in order to find an optimal task assignment that satisfies imposed constraints. The remainder of the present paper is organized as follows. Section 2 provides a brief outlook on related work. Then, Section 3 analyzes in MARTE expressivities for ERS design space exploration. Section 4 introduces the underlying approach of the MDE Hardware Software partitioning.Section 5 presents the exploration strategy. The translation, explained in Section 6, yields the main transformations of our proposal. To conclude, in section 7, the proposed method is briefly outlined and future work are given.

II. Related Work

Several co-design methodologies shown in literature prove MDE to be well appropriate to embedded systems design. As partitioning phase is done manually, the mapping for classic Hardware Software (HW/SW) partitioning is sometimes made in order to find a design implementation that fulfils all the specification requirements at a minimum cost. Other methodologies are based on designer experience while allocating tasks to execution platforms. In the rest of this section we locate our research in the context of two areas of related work: Co-design methodologies based on MDE and research that deals with scheduling problems and HW/SW partitioning at high abstraction levels. In particular, the present study focuses on MDE-based methods for test schedulability or HW/SW partitioning. Therefore, neither these two basic co-design concepts nor UML profiles will be surveyed in this paper. More details about them can be found in [33], [29] and [3].

The Accord|UML methodology [9] was initially based on the SPT profile exploitation but is currently relying on MARTE profile. It provides several automatic model transformation cycles that ensure analysis, prototyping and testing. The UML model refinement encompasses the generation of executable models for simulation. Principally, some views defined in the MARTE profile are mapped on schedulability and performance evaluation platforms. Nevertheless, translation process does not guide the designer to choose the suitable architecture among the different and possible implementations.

Gaspard [8] is a model-based methodology dedicated to the development of parallel and distributed applications implemented on SoC (System on Chip). It is limited to data flow applications and does not tackle data control application; so real time scheduling problems are ignored. The distribution of computations to processing elements and data to memories are performed through model transformation from a UML design model platform independent of a UML design model specific platform. The transformation process ends with the generation of optimized SystemC Code for repetitive structure architecture.

The paper in[5] proposes a methodology using MDE and UML TURTLE [1] profile for the specification and validation of ERTS design. It defines a flow, parallel to the development flow, which focuses on formal verification and validation. The ArchMDE proposed method provides a set of architectural independent platform meta-models and a set of transformation rules capable of generating specific models. The used meta-models are annotated with a profile that is not considered as standard. The allocation of tasks is also done manually.

As a complete environment for RTES co-design, the MOPCOM methodology [19] [34] [2] is worth mentioning. MOPCOM is based on three design levels. The fist one presents abstract models for system behaviour and requirements. The focus on the performance analysis of system requirements and functions comes in the second level. The last modeling level allows code generation for system simulation. DSE still remains to be based on designer experience.

The integration of model driven development concept and Design Space Exploration has been the focus of other research initiatives: Olivera et al. in [25] dealt with the exploration space using UML-SPT as a modeling language and the H-Spex (High-level design Space Exploration) tool. Above all, system properties are not well analyzed at design level; the automatic multi-objective design space exploration is handled with simulation models. Schedulability analysis and performance evaluation are also performed through simulation platforms.

Marcello presented in [22] an approach aiming at the combination of ERTS design by means of SysML [32], MARTE and DSE tool. The motivation behind this approach was mainly to help designers in evaluating the HW/SW partitioning solutions provided by a DSE tool. The design space selection points correspond only to allocations satisfying temporal constraints. The authors extend their work by joining the benefits of HW/SW Co-design and MDE in another proposal [23]. Their effort consists in proposing a semi-automated co-design frame work that integrates DSE, schedulability analysis and estimation techniques.

Bocchio et al. present a model driven co-design flow [27] that starts with a visual UML model for system specification and generates full implementations of the SW and HW components as well as their communication. In the authors' proposal, a generic hardware platform for the hardware architecture is selected and then the mapping of the software components on the given physical platform is carried out. However, this neither guarantees an optimal architecture nor determines the predictability of the system.

In [14], the authors provide a scheduling analysis tool based on MARTE profile. With a palette extension, users can easily create scheduling analysis views. In spite of the mentioned authors' valuable contributions, there is a need for a tool that gives predictions or verifications concerning the timing behavior. That is why they are developing an eclipse plug-in which translates information from the scheduling analysis view into the Meta model of SymTA/S [16] automatically.

In this context, further research is in progress. Each effort aims at transforming UML/MARTE model into scheduling analysis tool such as [15], [7] and [30]. Another contribution consists in mapping UML models to a tool for HW/SW partitioning like [4].

Hence, the proposed work is integrated in an organized infrastructure, which provides a model driven support for the design of embedded systems and allows the interaction between the exploration strategy and the user. In addition, in our model driven approach, the system models have sufficient details to enable the generation of a full system implementation from the models themselves. Indeed, they support not only the analysis

but also the translation to model-based estimation views.
Our approach differs from the aforementioned ones. In fact, in the same environment, it includes the specification of an application with the standardized UML profile MARTE adopted by the Object Management Group OMG, the schedulability test of given tasks and their implementation. Indeed, application and execution platforms are defined separately and the efficient partitioning is performed in a guided way. Our study based on model driven engineering concepts and particularly transformation rules, could well reduce the cost of mapping to co-design other tools and designer errors risk.

III. MARTE NOTATIONS FOR SW/HW PARTITIONING

As an introduction to the technical sections of the paper, this section overviews the UML/MARTE profile on which the work is based outlining the concerns related to design space exploration problems in co-design and scheduling analysis.
As already mentioned, a UML profile is the extension or the restriction of UML views for specific domains. In real time context, it represents a viable solution to decrease the complexity of ERTS which depends on the architecture deployment and requires runtime guarantees. From current times, the OMG has voted for a new standard for model driven development and analysis of real time systems. The adopted MARTE profile provides mechanisms to model appropriate specification in order to perform specific analysis. For that reason, we assume the models to be expressed in the new OMG standard MARTE in our work.
In fact, MARTE consists of three major packages. Foundation Package represents the foundational concepts for RTES design. It allows the specification of basic real time concepts such as non-functional properties (NFPs), time constraints and useful resources. The other two packages are refined from the first one. In fact, the second package named MARTE Design Model is dedicated for a detailed hardware and software description. As for the third package, MARTE Analysis Model package offers annotations for generic basis of quantitative performance and schedulability analysis. According to this structure, MARTE takes into account timing constraints and execution platform characteristics. We, next, present the profile concepts that are useful to our research.
In the MARTE Design Model package, the Software Resource Modeling (SRM) and the Hardware Resource Modeling (HRM) sub profiles present a specialization of the Generic Resource Modeling (GRM). SRM also intends to describe software multi tasking software platform such as Real Time Operating System (RTOS). To express real time and embedded features modeling, high level modeling concepts are delivered by means of the High Level Application Modeling (HLAM) package.
Thus, MARTE includes the Generic Quantitative Analysis Modeling (GQAM) package that supports early performance and schedulability analysis of system specifications as well as power, reliability, etc. The core of the GQAM is the Non-Functional Properties (NFP) annotations framework which provides a rich terminology to describe all system aspects such as response times, deadline, resource utilizations and power consumption. It contains two sub-packages for modeling Performance Analysis (PAM) and Schedulability Analysis (SAM). The first sub-package allows an early analysis of a design model. Timing behavior mistakes can be prevented or detected while analyzing the system under several configurations with different parameter values for many scenarios. The early analysis can be followed by a later temporal evaluation or modification on the scheduling policy. At this level, a set of scheduling analysis techniques and strategies has led to simulation or test feasibility approaches such as the Rate Monotonic Analysis (RMA) policy [20]. While SAM aims at predicting whether a set of software tasks meets its timing constraints, the PAM sub-package deals with the evaluation of ERTS. It offers means for determining how the system behavior uses system resources.
Before achieving the MARTE review, we should bear in mind that MARTE comprises the Allocation package which presents an association between an application and an execution platform. The set of all the allocations of functional application elements defines the mapping of the available resources.
According to the wealthy features offered by MARTE, the design space exploration parameters and the system constraints /requirements can simply be specified, whereas it does not include a method that guides the designer to the mapping process. In the next section, we will show through the proposed appraoch how to use the MARTE annotations for our purpose. The system refinement based on model transformations can lead to significant guidance improvements. The transformation of alternative solutions for evaluation process will be performed automatically.

IV. MODELING APPROACH

In this section, a model driven based method for HW/SW partitioning is proposed. Hence, we start by giving an overview of our proposed methodology. We, then, propose a method that accounts for these concerns at a high level abstraction layer. Finally, we detail each step of our process, highlighting the expressivity of MARTE notations for ERTS modeling for each abstraction level.

A. Our model driven approach for HW/SW partitioning overview

As most DSE approaches, we follow the Y-chart [18] to perform the optimal solutions among all possible combinations after mapping application to architectural specifications. Generally, three views are proposed to represent the system specification: Application, Platform, and Allocation in each level.
Figure 1 illustrates the refinement from high level concerns to detailed aspects. The first step of the process deals with the Abstract System Modeling (ASM), with a view to model system behavior. Once the system specification is described at high level abstraction, the user will be able to refine the physical platform to generic architecture components. This is the purpose of the second step that tackles Partitioning Activities. Our major contributions are addressed here: the user starts by defining how tasks are combined with the architecture

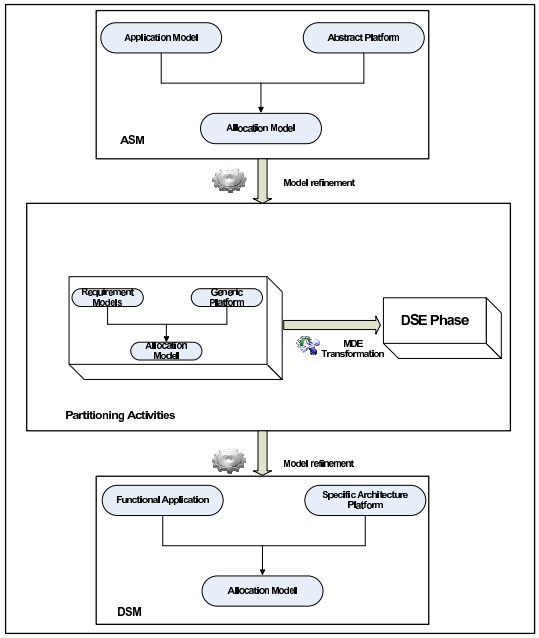

Fig. 1. Proposed Approach

and then translates models to the DSE activities. In step 3, Detailed System Modeling (DSM) is performed in the last two phases.

Since we tend to build our approach on standard MDE tooling, we base our implementation on the OpenEmbeDD platform which is an Eclipse Modelling Framework (EMF) dedicated to ERTS and relying on MDE. The selected UML modeller is Topcased for handling Meta Models instances. The transformation process must act on a transformation language. In our work, we choose Kermeta [6] which makes it possible to define models according to Meta Object Facility (MOF) Meta Model [26] in a textual form. In the rest of this paper, the overall approach is illustrated by a football player robot application [24]in which video tasks for object detection, wireless communications for message exchanging with other devices, motors controls, sensor acquisition, image processing and decision computation are included. The design of the robot system is of manageable complexity; thus, various HW with different granularities, real time constraints and SW with coprocessor implementations are considered for the set of tasks.

B. Abstract System Modeling

The ASM step is intended to cover real time scenarios via use cases UML diagram. A concrete scenario is identified for each use case. Figure 2 shows a scenario that exposes a global view of 4-task interactions. The used sequence diagram contains the stereotype "RtUnit" which is similar to the UML active object. It owns one or more schedulable resources. Several behaviors are related to an "RtUnit". For each one, a message queue for storing incoming messages is defined. An "RtUnit" Schedulable resource is used by a "PpUnit" which is considered as a passive object for modeling concurrency and shared information, so that the "PpUnit" stereotype can be used at this level. The "RteConnector" is also applied in order to ensure communication between two "RtUnit" entities. To capture system requirements, the sequence diagram is annotated with the Value Specification Language (VSL). VSL is an improvement of the Object Constraint Language (OCL) [11]. It comes to define complex expressions such as time duration, periodic behaviors, deadlines, etc. As shown in Figure 2, the period, deadline and dynamicity are attached to each task.

It should be noted that the cited stereotypes during ASM phase

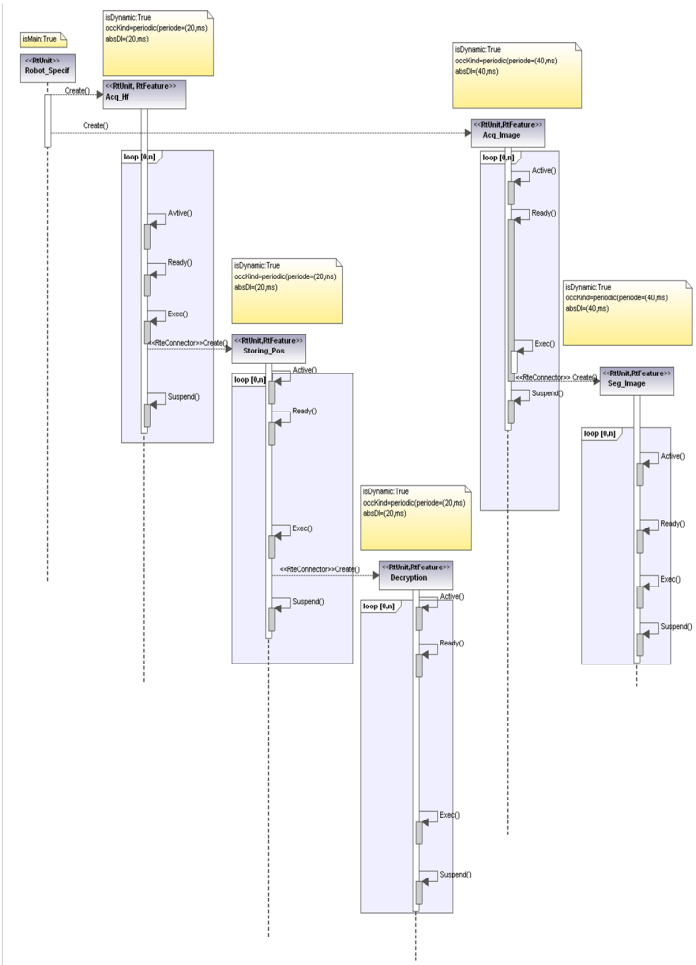

Fig. 2. Abstract application view

are included in the HLAM MARTE package. As for the platform and allocation, they are considered as the highest level of abstraction since the application is free of any architecture consideration.

C. Partitioning Activities

After an abstract system description, the designer has to clarify and further to refine the application characteristics more. So, three models are made:

- the platform: it denotes a virtual hardware architecture based on high level generic entities. The stereotypes "StorageResource", "ComputingResource", "CommunicationMedia" provided by the sub MARTE profile General Resource Modeling (GRM) are applied to highlight

the computing resources and their communication and the storage entites in a Platform Independent Model (PIM) of a Model Driven Architecture (MDA) paradigm [21].
- the application domains: they are described independently of computational model. Such domains can be described using the "SchedulableResource" stereotype included in the GRM package in order to define concurrent resources.
- allocation of the schedulable elements are presented via the "allocate"' stereotype included in MARTE Allocation sub profile.

After achieving the three models, an automatic mapping for the entry points of the DSE phase presented by the DSE formalization is completed in order to guide the designer to select the suitable execution platform for his application. The HW/SW partitioning that includes schedulability and performance tests will be explained more in the next section. The assignment of the tasks on the hardware architecture can be modified with reference to the feedback of the designer analysis.

D. Detailed System Modeling

At this point, the application, platform and allocation are performed from the previous DSE formalism and designer decision. Indeed, a generic hardware platform is generated. Figure 3 shows a UML deployment diagram of an architecture built around a processor, a Digital Signal Processor (DSP) and three accelerators. The whole of the computing resources are communicating through a standard bus. Fortunately, MARTE provides the HRM sub profile to describe such architecture efficiently. The stereotypes "HwProcessor", "HwPld", and "HwBus" are applied to the deployment view.

Besides hardware modeling, MARTE allows software model-

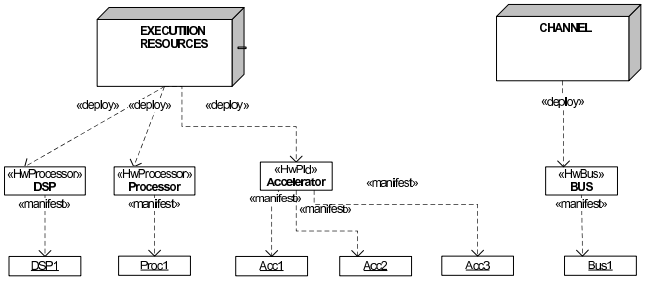

Fig. 3. Detailed Deployment View

ing via its sub profile SRM. In particular, we apply the stereotypes "SwSchedulableResource" and "EntryPoint" to model tasks and their dependencies, respectively. "SwSchedulableResource" is refined from "'RtUnit"' entities

The allocation of the software elements to the hardware platform is completed using "allocate" relationship since MARTE gives the notion of allocation that emphasizes distribution and temporal scheduling concepts, between application elements and hardware resource elements.

V. EXPLORATION STRATEGY

The guided implementation flow for HW/SW partitioning based on MDE and UML/ MARTE is presented in Figure 4. First, the UML/MARTE models for the requirements, generic platform and allocation are developed. Next, the behavior of the application is described as state-machines and sequence diagrams. Besides, he architecture presented in independent platform model defines the computing resources, as well as the communication between them. In the allocation model, the application is mapped onto the abstract architecture.

After explaining the mentioned three models, all different mapping parameters are ready to be translated to the design phase. At this level, possible implementation needs to be computed and the solutions that could not meet the schedulability requirements must be isolated. Thus, the previous three models are translated into a formalization of the DSE considerations. As a result, a primary solution that assigns all tasks to Software components is generated. Then, the transformation engine is invoked to perform the analysis of models. At this step, the schedulability annotations of the candidate solution represented by means of MARTE Schedulability Analysis Modeling diagrams are attached to the models.

The early analysis of the UML/MARTE models can lead

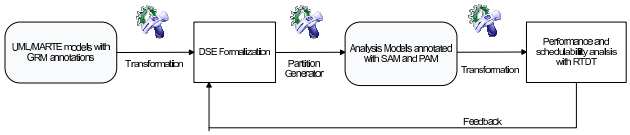

Fig. 4. Exploration strategy

to significant guidance improvements. However, analysis tools based on a lower abstraction level including simulations and consequentiality have the ability to test the system determinism. Many analysis tools were proposed in the literature; here, analysis tool can be addressed to performance or scheduling analysis tool and also partitioning framework. Among them, we bet on Real Time Design Trotter (RTDT) tool [31] due to its special sufficiency to support the Quality of Services QoS, its significant approach for schedulability and performance analysis and its generic design space exploration.

At this stage, the task that does not respect the required real time constraints is indicated to the implementation generator in order to implement it on a new hardware component. It is true that a hardware implementation will increase the cost, but it has positive influence on timing constraints. After selecting the partition corresponding to allocation options satisfying imposed constraints (e.g., deadlines, power consumption), the designer has to calculate the implementation cost in terms of energy consumption, area and material constraints and to compare it with previous cost. According to the DSE feedbacks, the assignment of the tasks on the hardware architecture can be updated. Consequently, a new automatic generation of candidate solution will take place.

A. DSE Formalization

The model driven exploration method starts with the formalization of the DSE with a tuple structure DSE. The properties of DSE structure are directly obtained from instances of the Meta models. In fact, information extracted from UML application structure/behavior models is annotated with

MARTE stereotypes. The DSE is the union of three tuples: Application, architecture and implementation.
Let us start with the application which is structured as follows: App=$< T, C_t >$ where:

- T = {T_0, T_1, ..., T_n} is a finite set of Tasks with $n > 0$;
- C_t : $T \times T \mapsto N$ is the matrix that describes the communication between tasks, where:
  $\forall\ T_i, T_j \in T, \forall\ i, j \in [0, n]$ and $NData_{ij} > 0$
  $\begin{cases} C_t(T_i, T_j) = NData_{ij} \Leftrightarrow T_i \text{ is dependent on } T_j \ \wedge\ i \neq j \\ C_t(T_i, T_j) = 0 \Leftrightarrow T_i \textit{ is not dependent on } T_j \ \vee\ i = j \end{cases}$
  $C_t(T_i, T_j) = NData_{ij}$ means that T_i is the task producer and T_j is the task consumer. The number $NData_{ij}$ is the data produced by T_i to the consumed task T_j during a period.

The Architecture tuple is structured as follows:
Arch=$< CR, C_{CR}, BUS, C_{in}, C_{out} >$ where:

- CR = {CR_1, CR_2, ..., CR_m} is a finite set of computing resources with $m > 0$. The communications between the different units of the CR set can be done with end-to-end communication or with a bus.
- C_{CR} : $CR \times CR \rightarrow \begin{cases} 0 \\ 1 \end{cases}$ is a matrix which presents an end-to-end communication between computing resources.
  $\forall\, CR_i,\ CR_j \in CR \wedge CR_i \neq CR_j,\ C_{CR}(CR_i, CR_j) = 1$
  $\Leftrightarrow$ $CR_j$is the sender and CR_i is the receiver
- BUS = {BUS_1, BUS_2, ..., BUS_k}is a finite set of bus communication with $0 < k < m$
- C_{in} : $CR \times BUS \rightarrow \begin{cases} 0 \\ 1 \end{cases}$
- C_{out} : $CR \times BUS \rightarrow \begin{cases} 0 \\ 1 \end{cases}$

Moreover, to represent the implementation, we define for each Task T_i :

- The vector implementations [Imp_{i1}, Imp_{i2}, ..., Imp_{im}], where Imp_{ij} is the estimated implementation j of the task i
- The vector number of cycle execution [$NCyc_{1i}$, $NCyc_{2i}$, ..., $NCyc_{mi}$], where $NCyc_{ji}$ is the number of execution cycles related to the implementation j of the task i
- The vector implementation costs of the possible implementation :[$Cost_{i1}$, $Cost_{i2}$, ..., $Cost_{im}$], where $Cost_{ij}$ is the cost of the implementation j of the task i. It is calculated with the same manner invoked by [31].

B. Scheduling Analysis

Scheduling analysis models are derived from the DSE formalisms. They present a set of UML models annotated with MARTE stereotypes. There are usually different analysis diagrams for evaluating different properties of the system properties such as deadline and time execution. Hence, the scheduling analysis models are performed automatically from a generated HW/SW partitioning. The imposed system requirements (delays, power consumption, hardware resources, real-time, and embedding constraints) are then checked through models execution.

In order to do that, the SAM sub-profile provides a set of elements to evaluate time constraints and guarantee system determinism. Figure 5 depicts the allocation of an application onto a given deployment partition using object diagram. The "SaEexecutionHost", "SaCommunicationHost" and "SaExecStep" stereotypes are applied to a specific analysis context provided by the GQAM stereotype "GaAnalysisContext". The first stereotype includes schedulability metrics and takes into account scheduling processing resources. The second one is used to relate schedulable resource entities and to control communication between them. The third stereotype is applied to a method to model a sequential computation on a "ProcessingHost". In addition to the structural view given by the object diagram, the sequence is adopted to represent the different scenarios of the system behaviors.

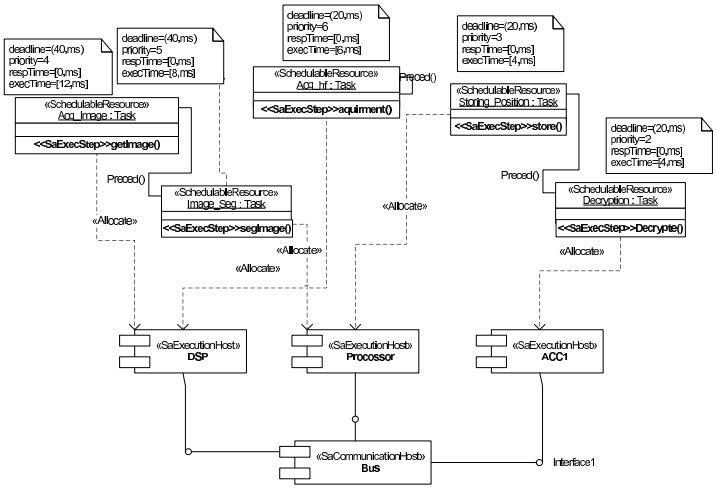

Fig. 5. Scheduling analysis of a given partition using Object Diagram

C. Performance Analysis

Our approach integrates two different steps of performance engineering repeatedly at two different stages of partitioning activities. An early performance feedback is carried out from the DSE formulization. It is followed by a seconded analysis provided by RTDT. Here, we are restricted to the illustration of the analysis provided by MARTE capabilities and used to measure metrics such as the delay or the probability of missing a target response.

Performance analysis provided by PAM sub profile integrates the description of a BehaviorScenario that corresponds to statecharts behavior diagram in our study. Figure 6 illustrates a sequence of actions annotated with "PStep" stereotype which aims at representing the probability of event success as well as the delay risk. Furthermore, the exchanged messages between objects on different nodes are defined through the "PCommunicationStep" stereotype as mentioned in Figure 6.
Figure 6 shows a state machine that provides a view of three tasks: acquisition, storing position and decryption. The main task is named Acquisition and has three states: Reception, Interpretation and classification. The nominal acquisition state is the Reception state. It enters in the Interpretation state for classifying the frequency. A sensor detects a position and then allocates its value in a buffer. After storing it, the decryption state is entered.

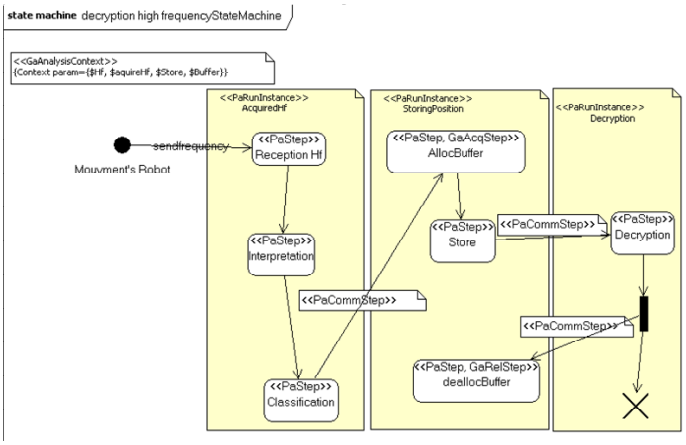

Fig. 6. Performance analysis of task acquisition using UML statecharts

D. RTDT supported by Model Driven Engineering

RTDT tackles the design space exploration problem. It processes the application and the architecture models which are inputs and they are described according to (Extensible Markup Language) XML format. The application model gives some information about processing tasks, like time behavior (e.g. execution time, period, etc.), time constraints, dynamic and static power cost, area cost, and so on. The architecture model represents the execution platform built around one processor that offers some opportunities for adding any processing accelerators, like hardware accelerators and/or co-processors, acceded through communication links. It includes some information about these processing elements and communication links that are related to area and power. RTDT also considers RTOS overhead, resource sharing between tasks, pre-emptive scheduling and multirate task graph. It attempts to find optimal alternative for system implementation, in terms of area and power cost according to the architecture mode, using simulated annealing heuristic. This alternative has to satisfy time constraints according to the run-time scheduler and input considerations. The analysis is performed according to probability concepts. However, a huge number of application types (hard and soft real time) are taken into consideration and treated in a uniform way.
Despite the interactivity of this platform, the definition of RTDT input files cannot be automated, as the specification of the real time constraints and QoS parameters are manual steps that have to be performed by the experts of RTES domain. An important step before the transformation process is the definition of a Meta Model for each RTDT input. Based on XML file, we propose the application Meta Model illustrated in Figure 7. The classes in the first target model match various real time application concepts. Now we describe the entities of each application component.

- Task: it is the central application component. It maintains a great deal of information such as elapse time, period. Data dependency between tasks is presented through Connection entity
- Implementation: It describes where a task can be implemented so that it has many characteristics such as the period and the execution time
- ImpCop: RTDT makes a difference between co-processors and other execution Hardware components. Tasks with the highest priority are implemented on co-processors
- RTQoS: it means the deadline ratio that has to be met
- AQoS: it represents the possible periods attributed to a task and depends on power consumption

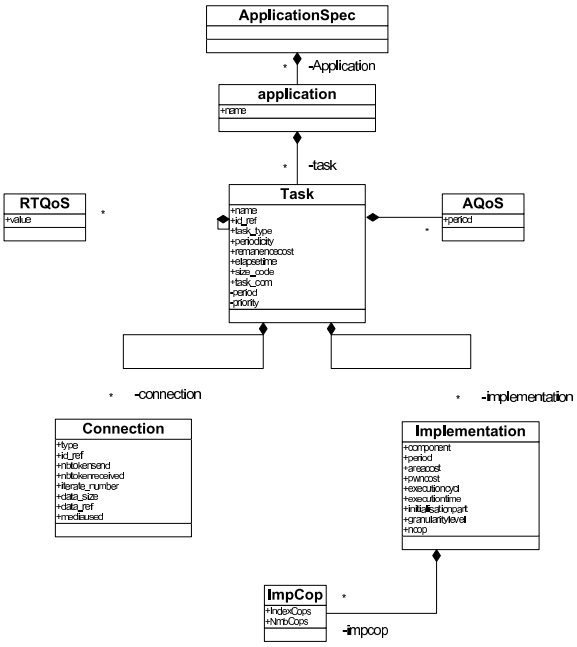

Fig. 7. RTDT Application Meta Model

In the same manner as application Meta Model is performed, the Meta Model related to the target architecture is built. As shown in Figure 8, the main semantics of elements are:

- Component: it is a processing unit similar to the class Component of RTDT application Meta Model
- GeneralConstraint: represents the different related constraints such as area cost and power consumption

Recently, the RTDT tool has been included into the Eclipse development environment. Thus, the translation has been implemented as an eclipse plug-in by embedding our Kermeta application in eclipse user interface. In fact, transformation from UML/MARTE models to RTDT contributes in an Eclipse view to the platform accessible through the reflective Ecore Model Editor. After defining a model conformed to the source Meta Model, RTDT input models are carried out automatically. Test schedulability and HW/SW partitioning are not immediately run after models extraction; it is triggered by a user intervention. It should be noted that our plug-in is interoperable and easy to integrate with other tools used within the model driven engineering process.

VI. TRANSFORMATION PROCESS

In this section, we give the key lines of the mapping of UML models into DSE formalization and RTDT tool. The

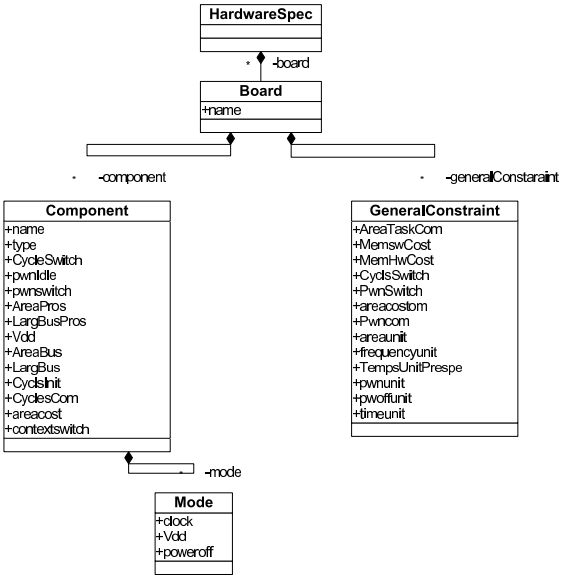

Fig. 8. RTDT Architecture Meta Model

objective of this step consists in transforming an XMI (XML Metadata Interchange) source model obtained automatically from a UML source model to an XMI target model.

A. From UML/MARTE to DSE Formalization and vice versa

The partitioning activities start with two major transformations. The first mapping deals with information from UML/MARTE models to DSE formalism while the second one aims at performing MARTE analysis models from the DSE. The transformation process based on Kermeta is the core of our complete top down systematic technique. It is conducted according to a set of rules, each of which depends on the applied stereotype to the UML class mapping. For example, any element that is stereotyped by GRM:: SchedulableResource in the source model is mapped to T vector of the DSE. Attributes are referenced by the used stereotypes to be translated into algebra DSE Meta Model. Table I gives a brief illustration about the translating mechanism because of the multiplicity of stereotypes and tagged values.
In the same manner as UML/MARTE models are mapped to DSE, the reverse transformation is built. As shown in Table II, the main correspondence between analysis models and DSE Meta Models are illustrated.

B. Mapping MARTE analysis models to RTDT

The transformation process starts within a mapping of MARTE execution platform to RTDT architecture described in the previous section. Every stereotype associated with "manifest" relation is mapped to an RTDT component that has a local variable "type" corresponding to a hardware component. For example, a "HwProcessor" manifested with "DSP" is mapped to an RTDT component having DSP type.

Beside defining their Meta Models, the conversion of schedulability analysis models to RTDT entries follows these main steps:

1) Tasks' transformation: each element stereotyped "SchedulableRessource" is mapped to the class "Task"
2) Timing constraints transformation: This act depends on the scenario of execution on a physical resource. For example, the period and the deadline are carried out from the method "execute()" stereotyped "SaExecStep "
3) Tasks dependencies transformation: If an object A stereotyped "SchedulableRessource" precedes an object B with the same type, a projection of a connection must be done. So, A.precede() is mapped to Task.connection()
4) Transformation of the partition to analyze: Each execution resource associated with an entity stereotyped "SchedulableRessource" is mapped to its corresponding implementation. Since a task is affected to a computing resource with "allocate" stereotype, we map the "allocate" stereotype to the RTDT "implantation" entity.

RTDT supports analysis context provided by MARTE profile as well as "GaWorkloadEvent" and "GaWorkloadBehavior " . Thus each state stereotyped "PaStep" is considered as a task state. So, it is transformed to the RTDT task entity. Moreover, each event is translated into RTDT connection. When the specificities of the RTDT tool and tranformation process to its input Meta Models are examined, there is a number of questions about transformation sequences and their order. Important is the question why application, platform and allocation are not mapped directly to RTDT. The solution of this question was presented in previous effort [17]. The automatic extraction of scheduling and performance models from UML/MARTE models can lead to an optimal tasks' distribution. Nevertheless, the user interaction and the separation of concerns are isolated. Another improvement in the present solution lies in the use of the DSE formalization that guarantees an easy mapping for any co-design tools in spite of their dissimilar input Meta Models.

VII. Conclusion and future work

This paper presents the issue of HW/SW partitioning co-design using Model driven engineering methodology and UML/MARTE profile. Adopting various UML view annotated with real time stereotypes in iterative design life cycles and implementation phase, our proposed methodology raises the credibility of partitioning decisions that are very significant mainly in the perspectives of managing schedule and risk.
The design flow comprises three stages: the first one describes a system modeling at a high level abstraction. The key item of our approach presents the second phase which includes the mapping to DSE formalization and the scheduling and performance analyses. The last step is the issue of the refinement of the two previous phases.
The keystone of the proposal is to recognize that model driven approaches, supported by adequate, compositional, mathematics formalization and tools, are needed to cost-effectively develop and evolve real time systems while guaranteeing their completeness and correctness. Besides

TABLE I
EXAMPLES OF UML/MARTE AND DSE META MODEL MAPPING

MARTE Concepts	DSE(algebra) Concepts
MARTE::GRM::Scheduling:: SchedulableResource	T
MARTE::GRM::Scheduling:: SchedulableResource	C_T
MARTE:GRM::ResourceTypes:: ComputingResource	C_R
MARTE::GRM::ResourceTypes:: CommunicationResource	C_{CR}
MARTE::GRM::ResourceTypes:: CommunicationMedia	BUS
GRM::ResourceTypes:: CommunicationEndPoint	C_{In}(relative)
GRM::ResourceTypes:: CommunicationEndPoint	C_{Out}(relative)

TABLE II
EXAMPLES OF CORRESPONDENCE BETWEEN DSE AND MARTE MODELS

DSE	UML/MARTE Concepts for design	UML/MARTE Concepts for analysis)	
		SAM	PAM
T	Scheduling Resources from GRM		PaRunTInstance + PaStep
C_T	Link between Scheduling Resources	SaCommStep	PaCommStep
C_R	Component	SaExecHost	
C_{CR}	Link between components	SaEndToEndFlow	
BUS	component	SaCommHost	
C_{In}	interface		

providing high level reuse for system development, our HW/SW partitioning approach supports the interaction between the exploration strategy and the user. For instance, the exploration strategy is able to guide the designer to select the suitable execution platform for his application. Indeed, models have sufficient details to enable the refinement and generation of a full system implementation from the models themselves. Besides the early analysis of schedulability and performance, the RTDT tool provides later NFPs evaluation or modification on the scheduling policy.

As future work, more Meta Models and rules transformation complementary to our MDE methodology are required to refine and complement the process of the automatic code generation such as VHSIC Hardware Description Language (VHDL) or C. Moreover, supplementary research on the formal verification is necessary to increase the success rate of the approach and decrease its risk factors.

REFERENCES

[1] Turtle: A real-time uml profile supported by a formal validation toolkit. *IEEE Trans. Softw. Eng.*, 30(7):473–487, 2004.

[2] Denis Aulagnier, Ali Koudri, Stéphane Lecomte, Philippe Soulard, Joël Champeau, Jorgiano Vidal, Gilles Perrouin, and Pierre Leray. Soc/sopc development using mdd and marte profile. In *Model Driven Engineering for Distributed Real-time Embedded Systems*. Hermes, 2009.

[3] Fateh Boutekkouk, Mohammed Benmohammed, Sebastien Bilavarn, and Michel Auguin. Uml2.0 profiles for embedded systems and systems on a chip (socs). *Journal of Object Technology*, 8(1):135–157, January-February, 2009.

[4] Moy Christophe, Raulet Mickael, Urban Fabrice, Nezan Jean-Francois, and Deforges Olivier. Syndex executive kernels for fast developments of applications over heterogeneous architectures. In *EUSIPCO'05, Antalaya (Turquey)*, September 2005.

[5] Nourchène Elleuch, Adel Khalfallah, and Samir Ben Ahmed. Archmde approach for the development of embedded real time systems. In *Ada-Europe*, pages 142–154, 2007.

[6] Jean-Rémy Falleri, Marianne Huchard, and Clémentine Nebut. Towards a traceability framework for model transformations in kermeta. In *In: ECMDA-TW Workshop*, 2006.

[7] Elena Fersman and Wang Yi. A generic approach to schedulability analysis of real-time tasks. *Nordic Journal of Computing*, 11(2):129–147, 2004.

[8] Abdoulaye Gamatié, Sébastien Le Beux, Éric Piel, Anne Etien, Rabie Ben Atitallah, Philippe Marquet, and Jean-Luc Dekeyser. A Model Driven Design Framework for High Performance Embedded Systems. Research Report RR-6614, INRIA, 2008.

[9] Sebastien Gerard, François Terrier, and Yann Tanguy. Using the model paradigm for real-time systems development: Accord/uml. In *OOIS '02: Proceedings of the Workshops on Advances in Object-Oriented Information Systems*, pages 260–269, London, UK, 2002. Springer-Verlag.

[10] OMG Object Management Group. Uml profile for schedulability, performance and time. 2002.

[11] OMG Object Management Group. UML 2.0 OCL Specification. OMG Adopted Specification ptc/03-10-14. Object Management Group, October 2003.

[12] OMG Object Management Group. Uml profie for modeling quality of service and fault tolerance characteristics and mechanisms. 2004.

[13] OMG Object Management Group. A UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems, Beta 2, ptc/2008-06-09. Object Management Group, June 2008.

[14] Matthias Hagner and Michaela Huhn. Tool support for a scheduling analysis view. In *MARTE workshop at DATE'08*, pages 41–46., 2008.

[15] M. Gonzalez Harbour, J. J. Gutierrez Garciia, J. C. Palencia Gutierrez, and J. M. Drake Moyano. Mast: Modeling and analysis suite for real time applications. *Real-Time Systems, Euromicro Conference on*, 0:0125, 2001.

[16] Rafik Henia, Arne Hamann, Marek Jersak, Razvan Racu, Kai Richter, and Rolf Ernst. System level performance analysis - the symta/s approach. In *IEEE Proceedings Computers and Digital Techniques*, volume 152, pages 148–166., 2005.

[17] Yessine Hadj Kacem, Adel Mahfoudhi, Hedi Tmar, and Mohamed Abid. From uml/marte to rtdt: A model driven based method for scheduling analysis and hw/sw partitioning. In *Eight ACS/IEEE International Conference on Computer Systems and Applications AICCSA*, May 16-19, 2010 (to appear).

[18] Kurt Keutzer, Sharad Malik, Richard Newton, Jan Rabaey, and Alberto Sangiovanni-Vincentelli. System level design: Orthogonolization of concerns and platform-based design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(12), December 2000.

[19] Ali Koudri, Didier Vojtsiek, Philippe Soulard, Christophe Moy, Joel Champeau, Jorgiano Vidal, and Jean-christophe Le Lann. Using marte in the mopcom soc/sopc methodology. In *workshop MARTE*, March 2008.

[20] C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM*, 20(1):46–61, 1973.

[21] J. Miller and J. Mukerji. Mda guide version 1.0.1. Technical report, Object Management Group (OMG), 2003.

[22] Marcello Mura, Luis Gabriel Murillo, and Mauro Prevostini. Model-

based design space exploration for rtes with sysml and marte. In *FDL*, pages 203–208, 2008.
[23] Luis Gabriel Murillo, Marcello Mura, and Mauro Prevostini. Semi-automated hw/sw co-design for embedded systems: from marte models to systemc simulators. In *FDL*, 2009.
[24] H.Kitano M.Veloso, E.Pagello. Robocup-99: Robot soccer world cup iii. In *Velsoso (Eds.)*.
[25] Marcio F. S. Oliveira, Eduardo W. Bri ao, Francisco A. Nascimento, and Flávio R. Wagner. Model driven engineering for mpsoc design space exploration. In *SBCCI '07: Proceedings of the 20th annual conference on Integrated circuits and systems design*, pages 81–86, New York, NY, USA, 2007. ACM.
[26] OMG. *Meta Object Facility (MOF) 2.0 Core Specification*. Object Management Group, October 2003. ptc/03-10-04.
[27] Elvinia Riccobene, Patrizia Scandurra, Sara Bocchio, and Alberto Rosti. A model-driven co-design flow for embedded systems. In *FDL*, pages 345–351, 2006.
[28] Douglas C. Schmidt. Model-driven engineering. *IEEE Computer*, 39(2), February 2006.
[29] Lui Sha, Tarek Abdelzaher, Karl-Erik Arzen, Anton Cervin, Theodore Baker, Alan Burns, Giorgio Buttazzo, Marco Caccamo, John Lehoczky, and Aloysious K. Mok. Real time scheduling theory: A historical perspective. *Real-Time Systems Journal*, 28(2/3):101–155, 2004.
[30] Frank Singhoff, Jérôme Legrand, Laurent tchamnda Nana, and Lionel Marcé. Cheddar : a flexible real time scheduling framework. *ACM Ada Letters journal, 24(4):1-8, ACM Press, ISSN :1094-3641*, November 2004.
[31] Hedi Tmar, Jean-Philippe Diguet, Abdenour Azzedine, Mohamed Abid, and Jean Luc Philippe. Rtdt: A static qos manager, rt scheduling, hw/sw partitioning cad tool. *Microelectronics Journal*, 37(11):1208–1219, 2006.
[32] Yves Vanderperren and Wim Dehaene. The sysml profile for embedded system modelling. In *FDL*, pages 589–598. ECSI, 2005.
[33] G. Vanmeerbeeck, P. Schaumont, S. Vernalde, M. Engels, and I. Bolsens. Hardware / software partitioning of embedded system in ocapi-xl. *Hardware/Software Co-Design, International Workshop on*, 0:30, 2001.
[34] Jorgiano Vidal, Florent de Lamotte, Guy Gogniat, Philippe Soulard, and Jean-Philippe Diguet. A co-design approach for embedded system modeling and code generation with uml and marte. In *DATE*, pages 226–231, 2009.

Yessine Hadj Kacem received a Master degree in Computer science in 2007 from the University of Sfax, Tunisia where he is now completing his Ph.D. thesis. His fields of interest are Model Driven Engineering and formal methods for Real Time Embedded Systems.

Adel Mahfoudhi is currently Assistant Professor at the University of Sfax in Tunisia. He obtained a Diploma in computer engineering in 1992 from the University of Monastir in Tunisia and received his Ph.D. degree in Computer Engineering in 1997 from the University of Valenciennes in France. His current research interests are formal methods for Embedded Real Time System modeling and verification. He is author/co-author of several papers in international conferences and journals.

Walid Karamti was born in Tunisia in 1985. He is currently completing his Master degree in Computer science in the University of Sfax, Tunisia. His research interests are mainly focused on Model Driven Engineering and formal methods.

Mohamed Abid is currently Professor at Sfax University in Tunisia. He obtained a Diploma in Electrical Engineering in 1986 from the University of Sfax in Tunisia and received his Ph.D. degree in Computer Engineering in 1989 from the University of Toulouse in France. His current research interests are High level abstraction methods for Embedded Real Time System modeling and verification. He has authored/co-authored over 100 papers in international journals and conferences. He served on the technical program committees for several international conferences. He also served as a co-organizer of several international conferences.