International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:10, 2009

A model driven based method for scheduling analysis and HW/SW partitioning

Yessine Hadj Kacem, Adel Mahfoudhi, Hedi Tmar , Mohamed Abid

Abstract—Unified Modeling Language (UML) extensions for real time embedded systems (RTES) co-design, are taking a growing interest by a great number of industrial and research communities. The extension mechanism is provided by UML profiles for RTES. It aims at improving an easily-understood method of system design for non-experts. On the other hand, one of the key items of the co-design methods is the Hardware/Software partitioning and scheduling tasks. Indeed, it is mandatory to define where and when tasks are implemented and run. Unfortunately the main goals of co-design are not included in the usual practice of UML profiles. So, there exists a need for mapping used models to an execution platform for both schedulability test and HW/SW partitioning.
In the present work, test schedulability and design space exploration are performed at an early stage. The proposed approach adopts Model Driven Engineering MDE. It starts from UML specification annotated with the recent profile for the Modeling and Analysis of Real Time Embedded systems MARTE. Following refinement strategy, transformation rules allow to find a feasible schedule that satisfies timing constraints and to define where tasks will be implemented. The overall approach is experimented for the design of a football player robot application.

Keywords—MDE, UML profile, scheduling analysis, HW/SW partitioning.

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 also to map the given system specification to hardware and software architecture. Indeed, it has always been a challenge. Thus, standards to facilitate 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 are tackled with model driven engineering (MDE) approach. In this context, UML profiles aim at being an adequate solution to support the whole life cycle co-design of complex Embedded Real Time System 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. It is true that system design with UML profiles has become an improved and an easily-understood method for non-experts, but two key items of co-design method which are the schedulability test and the HW/SW (Hardware/Software) partitioning are not totally covered with UML extensions.

Y.Hadj kacem, A. Mahfoudhi, H.Tmar and M.Abid are with the Computer and Embedded (ces) Laboratory, National Engineering School of Sfax,B.P.:w 3038, Sfax Tunisia e-mail: (see http://www.ceslab.org).

Since the main goals of co-design are not included in the usual practice of UML profiles, there exists a need of mapping used models to the model execution platform for both schedulability test and HW/SW partitioning. From UML views, the automatic extraction of tools input for these validation platforms are probably the best studied model simulation paradigms. Among all of them, we bet on Real Time Design Trotter (RTDT) [19] tool due to its special sufficiency to support the Quality of Services QoS, its probabilistic approach for schedulability analysis of fixed priority, and its generic design space exploration.
The main contribution of this work consists in proposing an MDE approach to the derivation of both HW/SW partitioning models and scheduling analysis models from UML/MARTE [10] models. Indeed, the main goals are the simulation of UML models via the automatic generation of the RTDT input tool. For the suggested models, the real time application is specified through a class diagram which includes the stereotypes related to scheduling aspects; Hardware and Software implementation. Then, architecture and application models are extracted automatically. Using transformation issues, RTDT can check if given tasks are schedulable or not and can generate an optimal architecture on which their tasks will be executed.
The remainder of this paper is organized as follows. Section 2 provides a discussion about related work. The proposed design methodology is described in section three, in which the used Meta Models and the transformation process are introduced. Section 4 is devoted to the illustration of the applicability of the proposed approach through a case study. Finally, a summary and future works are given.

II. Related Work

UML refinement by means of extensions, stereotypes and tagged values are used to reduce the complexity of system development. Thus, UML profiles represent a viable solution to increase their complexity. Besides, a great deal of reported research and contributions to the paradigm UML for ERTS have been proposed in the literature. In this work, our study is particularly focused on methods based on MDE 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 [21], [17] and [2].
Bocchio et al. present a model driven co-design flow [16] that starts with a visual UML model for system specification and generate full implementations of the SW and HW components as well as their communication. In the 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.

Another work [7] exploits transformation rules to derive real time schedulability models from UML profile for schedulability, performance and time SPT by proposing a Meta model for schedulability analysis technique. Nevertheless, the target Meta model does not have the ability to calculate the response time and utilization factor of the processor. The target Meta model does not decide on the feasibility of task scheduling.

There are also works based on Petri formalism as target models, like that of [14] translating a subset of UML diagrams (for example Statecharts and Activity diagrams) into stochastic Petri nets to carry out a performance model representing the whole system via transformation rules. Likewise the authors of [1] and [3] propose a mapping of the UML models into the performance model to formalize and analyze quantitative aspects of the system. However, these works remain a solution for performance and timing analysis and cannot confirm the schedulability of system but a dedicated Petri Net or its extension to cover this gap as in [20].

In [11], the authors provide a scheduling analysis tool based on MARTE profile. With a palette extension, user can create easily scheduling analysis views. In spite of their valuable contributions, there is a need for a tool that gives predictions or verifications concerning the timing behaviour. 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 [13] automatically. In this context, further research is in progress. Each one aims at transforming UML/MARTE model into scheduling analysis tool such as [12], [6] and [18]. Another contribution consists in mapping UML models to a tool for HW/SW partitioning like [4].

Our approach differs from the aforementioned one in the sense that it integrates in the same environment the specification of an application with recent UML profile MARTE adopted by the Object Management Group OMG, the schedulability test of given tasks and where they will be implemented. In addition, it includes their dependency, QoS metrics, power cost, area cost and probabilistic assumptions, so that scheduling analysis and HW/SW partitioning become feasible in an easier and more flexible way. The applicability of this new proposal is based on the model driven engineering concept.

III. PROPOSED APPROACH

Our work based on model driven paradigm, proposes an automated process where models are expressed as first class artefact of the development flow. Non- functional properties such as real time constraints are analysed at an early stage and then translated into specefic output Meta Model. Mapping between models is technically challenging and key to the whole automated approach. In the rest of this section, we start by giving an overview of our approach. After that, we describe the input and the output Meta models. Then, we present the transformation process and the environment that supports our proposal.

A. Overview

The design methodology comprises three stages: system modeling, automatic extraction of architecture and application models annotated with MARTE profile and finally scheduling analysis and HW/SW partitioning and our concern will be only on the second one. The purpose of our design flow presented by figure 1 is to map a specific UML model annotated with MARTE profile for RTES to a model execution platform. In fact, the designer starts by specifying the requirements of the application and the architecture including various constraints such as dependency, power consumption, area cost and particularly real time constraints. Then, the transformation extracts automatically the input models of RTDT from initial specification. In the final stage, RTDT should produce schedulability test and an architecture to which the application is mapped. In the rest of this section the used models are reviewed since transformations between models are the key elements of MDE. Figure 1 illustrates the main concepts of our methodology.

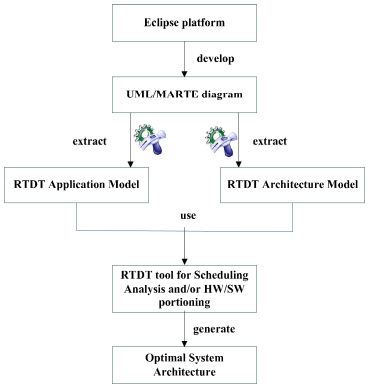

Fig. 1. Proposed Design Flow

B. Input Meta Model

The recent MARTE profile comes to upgrade the profile Schedulability, Performance, and Time (SPT) [8] and the profile Quality of Service and Fault Tolerance (QoS&FT) [9]. It consists of three major packages. Foundations 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 two other packages are refined from the foundation one. The second package named MARTE Design Model is dedicated for 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

take into account timing constraints and execution platform characteristics. Consequently, we assume the input design model while the transformation process is expressed in UML annotated with MARTE profile in this paper.

Considering the requirements of RTDT input files, the system static view is identified. Using MARTE and UML dynamic views, RTES behaviour can be performed; in our study we are limited to a presentation through a class diagram since RTDT can analyse system properties without using the great number of package offered by MARTE profile. Figure 2 shows a class diagram of necessary information required by RTDT. It describes tasks and where they can be implemented, the resources and communication supports. Furthermore, some other entities can be added to the source Meta model such as mutual exclusion resource; but they are not supported by the analysis tool.

In particular, we are restricted to using the Hardware Re-

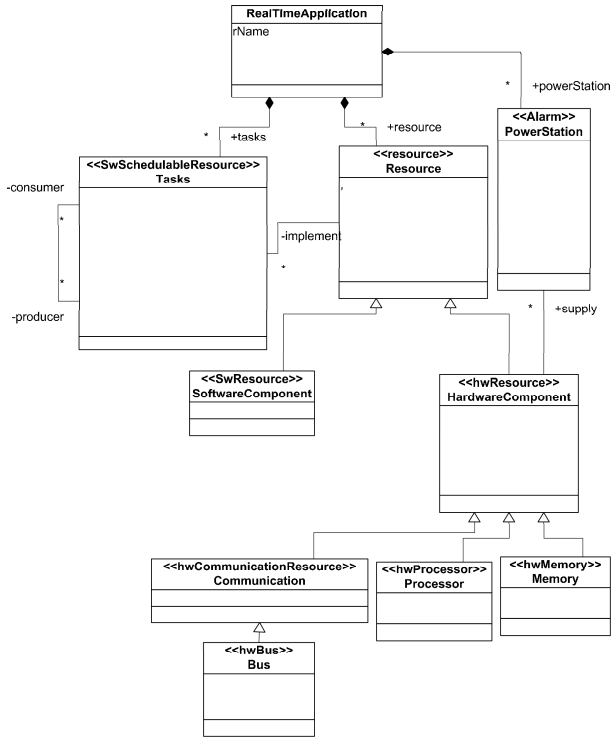

Fig. 2. Source Meta model annotated with MARTE profile

source Modeling HRM and Software Resource Modeling SRM. The HRM package provides modelers with a set of entities covering hardware RTES execution platform. HRM logical and physical views present functional properties and physical characteristics respectively. The SRM is composed of four views. The first view presents a general resource related to a specific domain. The second one provides a concurrent support whose interaction between its components is detailed in the third package. The last package represents the software resource brokers.

Using SRM and HRM packages, system architecture and application can be represented at high level view. Class stereotypes are used to represent the different software and hardware components, i.e., tasks, precedence, possibilities of implementations and resources. The adopted representation allows designer to instantiate the proposed Meta model in a simple and fast way thanks to the stereotypes provided by MARTE. The main selected stereotypes are:

- SchedulableResource : it presents the resources that execute concurrently to other concurrent resource
- Alarm resource: it determines executing context to a routine
- SwResource: it models the structure of software entities provided to the user via execution supports
- HwProcessor: it is responsible for scheduling and executing threads
- HwMemory : it represents a storage component for data and executable code
- HwBus : it ensures communication among other execution platform components

C. Application and Architecture Meta Models

RTDT tackles the design space exploration problem. It processes the application and the architecture models which are inputs and are described according to XML format. The application model gives some information about processing tasks, like time behaviour (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.

In spite of 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. A second step of the transformation process is the definition of a Meta model for each RTDT input. Based on Extensible Markup Language (XML) file, we propose the application Meta Model illustrated by Figure 3. The classes in this first target model match various real time application concepts. Now we describe these 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 hast to be met
- AQoS: it represents the possible periods attributed to a task. It depends on power consumption

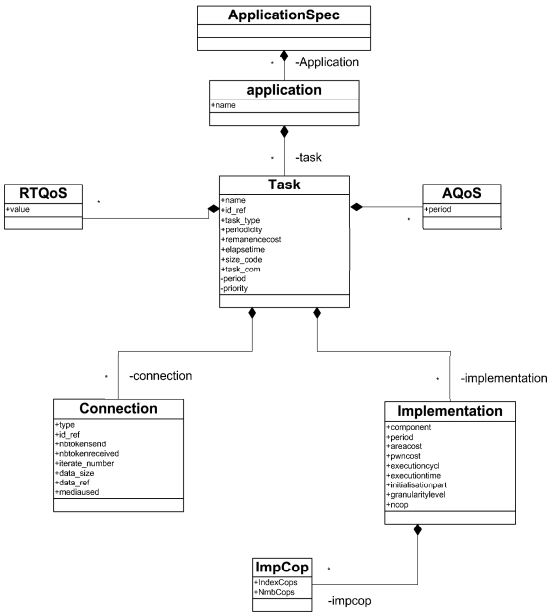

Fig. 3. RTDT Application Meta model

In the same manner as application Meta Model is performed, the Meta Model related to the target architecture is built. As shown in figure 4, the main semantic 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

D. A Meta model based transformation

The 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. The model transformation is based on Kermeta [5]. It makes it possible to define models according to Meta Object Facility (MOF) meta model in a textual form and it can be used as a transformation language. The transformed source model corresponds to the diagram of class presented in Figure 2. The code corresponding to XML based XMI offers a tree structure to our model by presenting the classes and the attributes in textual format.

The transformation process is the core of our complete top down systematic technique. It is carried out according to a set of rules. Here we consider a rule as a mapping between

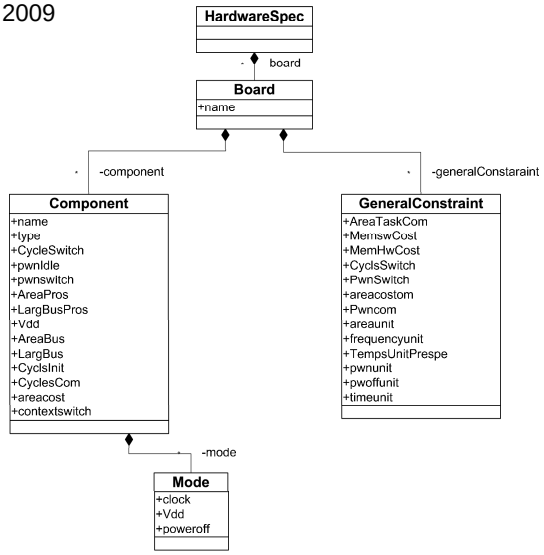

Fig. 4. RTDT Architecture Meta model

models not as a rule transformation since Kermeta is not based on transformation rules. Each rule depends on the applied stereotype to the UML class mapping. For example, any element that is stereotyped by SRM::SwSchedulableResource in the source model is mapped to RTDT task. Attributes are referenced by the used stereotypes to be translated to the attributes of RTDT Meta model. For instance, the attribute named period, referenced as a periodElements in the SwSchedulableResource stereotype is mapped to the attribute Period related to the RTDT Task class. Table I just gives a brief illustration about the translating mechanism because of the multiplicity of stereotypes, tagged values.

Typically, the application and the architecture target models are obtained simultaneously through the same process. To perfom the application Meta Model, we create in a first step all tasks of the given application. Then for each task, the following entities are defined:

- connection type: due to the reflexive relation producer consumer in the source Meta Model, the connection type in or out for the application Meta Model is easily carried out.
- the possible implementation: here we have to distinguish between Hardware and Software Resource. Two cases must be taken into account: If the resource type is HardwareComponent type then the attribute granularityLevel should be added. For the coprocessor resource, it is necessary to create the ImpCop class which contains the specific coordinates of each coprocessor

The architecture Meta Model is determined in a uniform way

E. Environment supporting our approach

Recently, the RTDT tool is being 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, our tool contributes in an Eclipse view to the platform accessible

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

MARTE Concepts	RTDT Concepts
addresSpace (SRM ::SwConcurrentResource)	Task ::TaskareaCostCom
PriorityElements (SRM ::SwConcurentResource)	Task ::Priority
baseUnit (NFP ::Unit)	Resource ::TimeUnit
area (HRM ::Hw_Resource)	Processor ::areaPros
Frequency (HRM ::Hw_Resource)	HardwareComponent ::freqUnit

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.

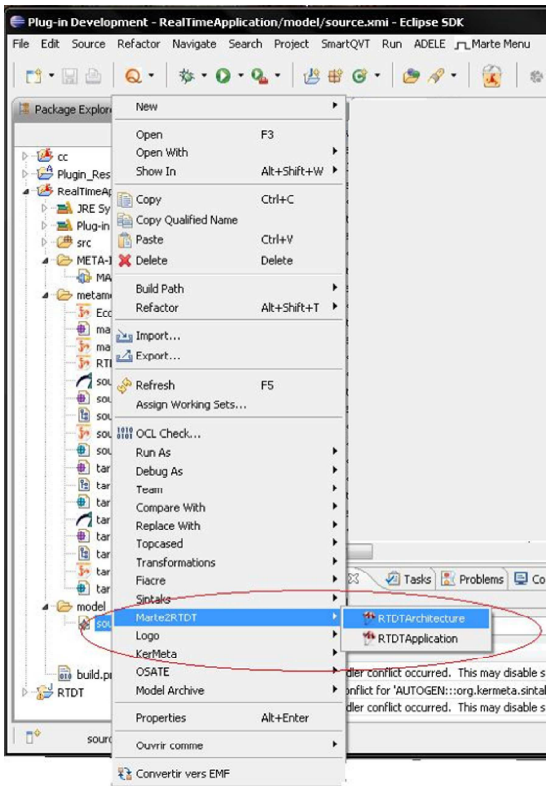

Fig. 5. Eclipse plug-in for the extraction of RTDT models from UML/MARTE class diagram

Figure 5 shows how transformation is divided into two model transformations that are executed independently. The first one processes the source Meta model and generates a set of tasks with the different possibilities of implementation, dependency, real time constraints and QoS requirements. The second one generates a set of Hardware components with their types, area cost, etc.

IV. EXPERIMENTAL RESULTS

The case study [15] presented in previous work [19] is taken again. It presents a football player robot application where 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, SW and SW with coprocessor implementations are considered for the set of tasks.

Figure 6 shows a view of the tasks as well as their dependencies and the variation of execution time. To prove the effectiveness of our model driven approach, it has been evaluated it through a comparison with classic RTDT. Hence, the same specification of the experiment illustrated by Figure 6 was provided to a designer that has limited knowledge about UML, scheduling analysis and HW/SW partitioning.

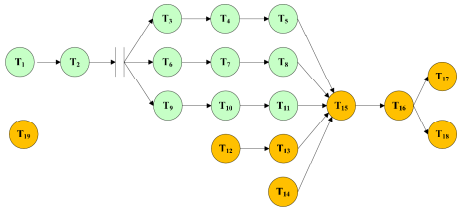

Fig. 6. Football player robot application

With classic RTDT, the modeler had to understand the Document Data Type DTD of the two input XML files related to real time design turtle tool. Many difficulties were found while specifying application requirements and constraints especially timing constraints and task's implementation. A significant difference between the previous tool and the model driven approach was noted. The same designer has just to fill in an XMI file conformed to the input Meta Model presented in Figure 6. The result of the second test was more successful satisfying the case study. As a result, modelers do not need to be familiarized with co-design methods in general and with RTDT in particular. Nevertheless, one limitation of our approach is that it can only handle models conform to the source Meta model.

The evaluation experience demonstrates that scheduling analysis and HW/SW partitioning based on model driven engineering concept can be a benefit for non-expert ERTS designers. Since MDE is a very promising methodology to support high level reuse for system development, the designer just has to instantiate the input Meta model of our methodology. As a matter of fact, models can be reused and easily manipulated in order to take into account various kinds of real time systems. That is why we are restricted on the presentation of one case study while proving the usefulness of our method.

V. CONCLUSION

A model driven based method for the scheduling analysis and HW/SW portioning of real time embedded systems has

been proposed. It was shown that the translation of UML class models into RTDT as a model execution platform can cover all co-design phases, even in the presence of complicated system specification. This framework that supports QoS choices provides a schedulability test. An optimal design space exploration that takes into account many metrics such as area cost, power cost and energy constraints can also be obtained by selecting the suitable target architecture.

In the proposed methodology, the modeler takes advantage of the new standard UML profile for Modeling and Analysis of Real Time Embedded Systems developed by the UML/MDE community. He starts from UML/MARTE diagrams to specify the application especially tasks characteristics, their dependency as well as their implementation possibilities. Our approach aims at mapping UML class diagram annotated with MARTE stereotypes into RTDT input models. It simultaneously integrates scheduling analysis, design space exploration and model driven engineering. This approach could reduce the complexity of design phase by checking the feasibility of tasks scheduling at early stage and on high level abstraction. It also determines where and when task are implemented and run.

An experiment has been done on a football player robot application. This practical proof is one contribution compared to the UML mapping for the execution platform for RTES taken by other researchers. The presented case study confirms that our approach is beneficial for non- expert designer.

Further works still remain to be done; the source Meta model can be improved in order to support all tools for HW/SW portioning and scheduling analysis. Some real time constraints can be checked with formal methods such as the extension of Petri Net.

REFERENCES

[1] S. Bernardi and J. Merseguer. Performance evaluation of uml design with stochastic well-formed nets. *Journal of Systems and Software*, 80(11):1843–1865, 2007.

[2] 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.

[3] Javier Campos and José Merseguer. On the integration of uml and petri nets in software development. *Lecture Notes in Computer Science*, 4024:19–36, 2006. Invited paper.

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

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

[7] Abdelouahed Gherbi and Ferhat Khendek. From uml/spt models to schedulability analysis: a metamodel-based transformation. *Object-Oriented Real-Time Distributed Computing, IEEE International Symposium on*, 0:343–350, 2006.

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

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

[10] OMG Object Management Group. Uml profile for modeling and analysis of real-time and embedded systems (marte). In *RFP*, 2006.

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

[12] 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.

[13] 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.

[14] José Merseguer and Javier Campos. Software performance modeling using uml and petri nets. In *MASCOTS Tutorials*, pages 265–289, 2003.

[15] H.Kitano M.Veloso, E.Pagello. Robocup-99: Robot soccer world cup iii. In *Velsoso (Eds.)*.

[16] Elvinia Riccobene, Patrizia Scandurra, Sara Bocchio, and Alberto Rosti. A model-driven co-design flow for embedded systems. In *FDL*, pages 345–351, 2006.

[17] 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.

[18] Frank Singhoff, Jérôme Legrand, Laurent tchamnda Nana, and Lionel Marc. Cheddar : a flexible real time scheduling framework. *ACM Ada Letters journal, 24(4):1-8, ACM Press, ISSN :1094-3641*, November 2004.

[19] 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.

[20] W.M.P. van der Aalst. Petri net based scheduling. Computing Science Reports 95 /23. Eindhoven University of Technology, Eindhoven, 1995.

[21] 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.