Schedulability analysis and optimization in a model-based integrated tool-chain.

Synthetic MARTE models for optimizing real-time design with MAST and TEMPO

Laurent Rioux, Rafik Henia, and Nicolas Sordon
Thales Research & Technology
1 avenue Augustin Fresnel
F-91767 Palaiseau, France
firstname.lastname@thalesgroup.com

Michael Gonzalez Harbour, José Javier Gutiérrez, Juan M. Rivas, César Cuevas, José María Drake, and Julio Medina
Universidad de Cantabria.
Av Los Castros s/n, 39005 Santander, Spain
{mgh,gutierjj,rivasjm,cuevasce,drakej,medinajl}@unican.es

***Abstract*—This demonstration presents a model-based tool chain for the design of real-time applications. It uses as first step a synthetic version of the UML Profile for MARTE standard as a design model. This is transformed into an intermediate model called TEMPO-Analysis that condenses the instance view of the concrete scenarios to be analyzed. The TEMPO-Analysis model is finally transformed into the tool specific schedulability analysis input models used to obtain the resulting response times of the application scenarios of interest. The analysis set of tools used in this effort is MAST but other tools may be used also in the final analysis step if needed. The automation that these models and model-transformations enables, is used to automatically iterate the analysis in the search of feasible/optimum scheduling parameters and tasks-to-processors allocation. This short abstract presents the basic building blocks and functionality of the tool chain used in the demonstration and describes the main visible stages in the model transformations involved.**

Keywords—schedulability analysis; response time analysis; models; real-time; optimization; design space exploration.

I. Introduction

The complexity of real-time applications, added to the industrial constraints on quality and time-to-market, post significant challenges in the design of real-time embedded systems. The traditional usage of the "V"-cycle for real-time systems does not fit well the needed performance verification and validation activities, since they start only after the development and integration phases are complete. At that point, performance issues are extremely difficult and expensive to solve. Thus, a reliable performance prediction model assessed at early design stages would be helpful to guarantee timing predictability before time and resources are invested in a potentially wrong system implementation.

The tool chain that we propose and show in this demonstration is a result of the fruitful collaboration of Thales Research & Technology, and the University of Cantabria, industry and academia. It represents a contribution to the industrial exploitation of model-driven technologies, response time schedulability analysis, and optimization techniques in the design of real-time systems in a variety of application domains.

This work has been funded in part by the Spanish Government and FEDER funds under grants TIN2011-28567-C03-02 (HI-PARTES) and TIN2014-56158-C4-2-P (M2C2). This work reflects only the authors' views; the funding organisms are not liable for any use that may be made of the information contained herein.

Fig. 1 shows a simplified block view of the main models, tools and transformations used in this integrated tool chain.

Using the MARTE modelling standard [1] as a basis for the synthetic design view (Tempo-MARTE) helps to address all the different formalisms that this integrated tool is meant for. Having also a variation of MARTE-SAM (Tempo-Analysis) as an intermediate model for the schedulability analysis tools helps to generate inputs for those tools that might be not directly MARTE compatible. This is useful in particular for those analysis tools that do not support design oriented model elements like SaSteps applied on structural elements, or schedulable resources like threads for example.

The meta-models have been formalized on Ecore, and all transformations work under the Eclipse Modeling Framework. Some are implemented in Java, others in ATL. No UML graphical tool is used for the automation of the analysis.

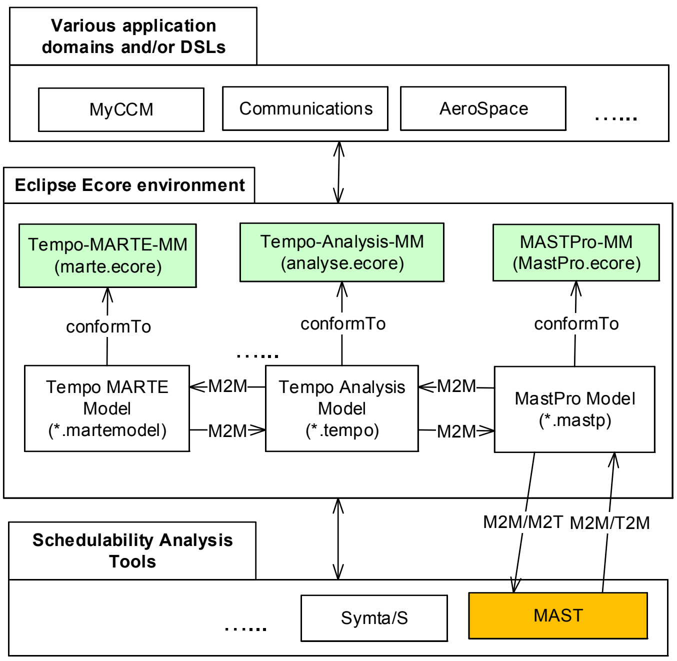

Fig. 1. A simplified view of the models, tools and transformations used.

II. SCHEDULABILITY ANALYSIS

As already mentioned, having a scenario based intermediate model like Tempo-Analysis is useful to address transformation into models suitable for a variety of schedulability analysis tools. In this demonstration the tool to use is MAST [2]. The automated link to MAST has been implemented using ATL to get the input models, and also to return the output timing results. A specific MAST tool launcher is invoked by the plugin so all its functionalities [3] can be used from the integrated tool chain. These include the selection of the various analysis techniques, the ceilings and blocking times calculation, as well as the native MAST sensibility analysis. More details about the generation of a Tempo-Analysis model from a Tempo-MARTE model can be found in [4].

III. OPTIMIZATION TOOL ARCHITECTURE

Besides the link to MAST as a tool for schedulability analysis, the integrated tool chain we show is able to optimize the assignment of priorities. Currently it is also being extended to support the automated search for feasible tasks to processors allocation variations in distributed systems.

In order to allow the specification of other parameters necessary for the priority assignment and the architecture optimization algorithms, the input analysis model has been complemented with an additional model. This model contains the constraints that define the values of the priorities available to each task and the tasks that belong to the same OS thread as well as the possible allocations. This way the iterations on the analysis model can be done either by successive invocation of the analysis tool or inside the tool if it can handle the optimization. Since MAST has already a native priority assignment tool, this approach scales quite well to the situation, allowing to use MAST or any other analysis tool for which the appropriate link exists.

We propose an architecture that have both the architecture and priority optimizations working with the Analysis Model, thus avoiding unnecessary model transformations (between Tempo-MARTE and Tempo-Analysis models), as shown in Fig. 2.

The MAST Priority Optimization tool is offered as a stand-alone tool working with the input Tempo-Analysis and Constraints models. It works with the selected analysis tool, and if desired, it will also integrate a shortcut to the MAST analysis tools in order to be more efficient.

The Real-Time Architecture Optimizer will work also with the Tempo-Analysis and Constraints models, and will use the MAST Priority Optimization tool over any new architecture proposed by the selected optimization algorithm.

IV. DEMONSTRATION

Several practical use cases are available as hands-on demonstration of the qualities of the tools. One in particular might be of interest to the attendees since it appears reported as an industrial challenge for the timing verification of a deployable real system in WATERS 2015 [5]. The results obtained for that use case using previous versions (less automated) of the tools have been reported in [6].

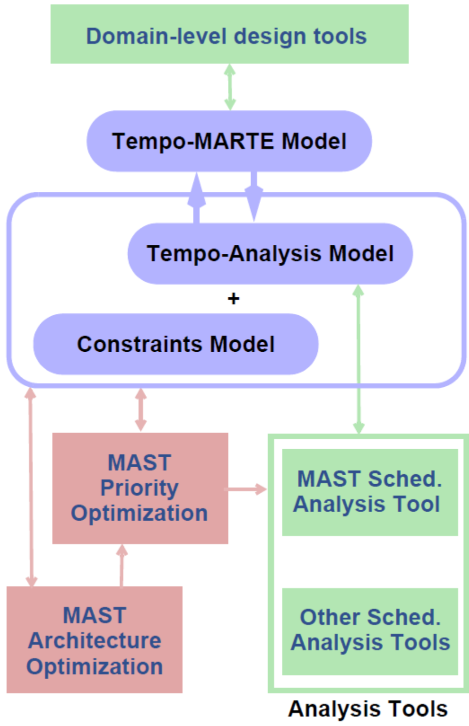

Fig. 2. Architectural design of the optimization tools

The tools we present here have been integrated in the eclipse development environment, but they may well run also as standalone transformations. This way the automations can be inserted in different software production chains and/or being adapted to different formalisms or application domains.

As might be expected the initial design intent models are produced by hand. The formalism to use for this initial step is to be the closest semantically to the target user application domain or modelling language.

REFERENCES

[1] Object Management Group, UML profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems, version 1.1, OMG document formal/2011-06-02, 2011.

[2] M. González Harbour, J.J. Gutiérrez, J.C.Palencia and J.M.Drake, MAST: Modeling and Analysis Suite for Real-Time Applications, in Proc. of the Euromicro Conference on Real-Time Systems, June 2001.

[3] MAST home page: http://mast.unican.es/

[4] Rafik Henia, Laurent Rioux, Nicolas Sordon, Gerald-Emmanuel Garcia, and Marco Panunzio: Integrating Formal Timing Analysis in the Real-Time Software Development Process, in WOSP '15 Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development Pages 35-40

[5] https://waters2015.inria.fr/files/2014/11/FMTV-2015-Challenge.pdf.

[6] Julio L. Medina, Juan M. Rivas, J. Javier Gutiérrez, and Michael González Harbour. Solving the 2015 FMTV Challenge Using Response Time Analysis with MAST. In Procceedings of The 6th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS'2015)