Experiencing the UML profile for MARTE in the generation of schedulability analysis models for MAST

Julio L. Medina and Álvaro García Cuesta

Departamento de Electrónica y Computadores, Universidad de Cantabria, 39005-Santander, SPAIN
{ julio.medina , alvaro.garciacuesta }@unican.es

Abstract

This short paper presents the design, implementation, and initial results of a tool that extracts from UML models the schedulability analysis data that are necessary for the application of the MAST set of tools on them. The input analysis models represent concurrent, distributed real-time systems that are formalized in UML annotated with the extensions proposed by the UML Profile for MARTE; the OMG standard profile for the modelling and analysis of real-time and embedded systems. The experiences performed with the modelling elements that have been selected to support the underlying analysis methodology have led to the rising of concrete issues, which have been sent to the OMG for the improvement of the standard, and are also briefly described here.

1. Introduction[1]

Model-based software development is one of the most promising software engineering approaches, since using reusable, configurable, and composable models may help significantly in the separation of concerns, increasing the efficiency, but also the quality of software.

In the case of applications with real-time requirements, a model-based methodology can help by simplifying the process of building their temporal behaviour analysis models. These models constitute the basis of the real-time design and the schedulability analysis validation processes. With that purpose, the designer of a real-time system must generate, in synchrony with the models used to generate the system's code, an additional parameterizable model, suitable for its timing/schedulability validation. In the approach that we assume here, these analysis models are to be derived from the high level design models annotated with a minimum set of real-time features taken from the requirements of the application in which they are to be used. In analogy to the generation of the application's code, the analyst, or application designer, have also to define the set of real-time sub-models, and build the complete real-time analysis model of the application.

The objective of this paper is the identification of a practical set of constructs in the Schedulability Analysis Model of MARTE in order to construct and extract the input model necessary to perform schedulability analysis with the MAST set of tools [2].

A discussion of the process followed for the modeling and the organization of the timing characteristics of a distributed, concurrent, reactive real-time system in a UML based methodology may by found in [1]. This short paper concentrates on (Section 3.) the modeling elements equivalencies for the generation of the output MAST analysis models, (Section 4.) the description of the technologies used for it, (Section 5.) the issues found in MARTE for the representation of such systems, and finally some conclusions and future work.

2. Related work

Some other tools have been realized in this direction. The closest in style and modeling capabilities is the RSA plugin to perform schedulability analysis by means of RapidRMA [5]. The version of the tool that is available has some limitations: it supports only scheduling analysis for mono-processors, with periodic and sporadic events (through sporadic servers). It does not provide support for multi-processors and distributed systems. RapidRMA and IBM RSA are not integrated thought the GUI. Moreover, there is no automatic launch of RapidRMA after the input files are generated. It requires a manual operation. The current implementation does not offer any feedback capabilities from RapidRMA to the UML modeling tool. All the analysis results can be exploited within the tool only. Similar limitations plus a lack in modeling guidance is provided by the tool [6] for representing Cheddar models with MARTE. The explanatory document shows how MARTE concepts can be matched to those used in Cheddar in order to do analysis on models and proposes model transformation solutions using ATL.

1. This work has been funded by the European Union under contracts, FP7/NoE/214373 (ArtistDesign), and FP7/CSA/224330 (ADAMS); and by the Spanish Government under grants TSI-020400-2009-108 (ITEA2-EVOLVE), and TIN2008-06766-C03-03 (RT-MODEL). This work reflects only the author's views; the EU is not liable for any use that may be made of the information contained herein.

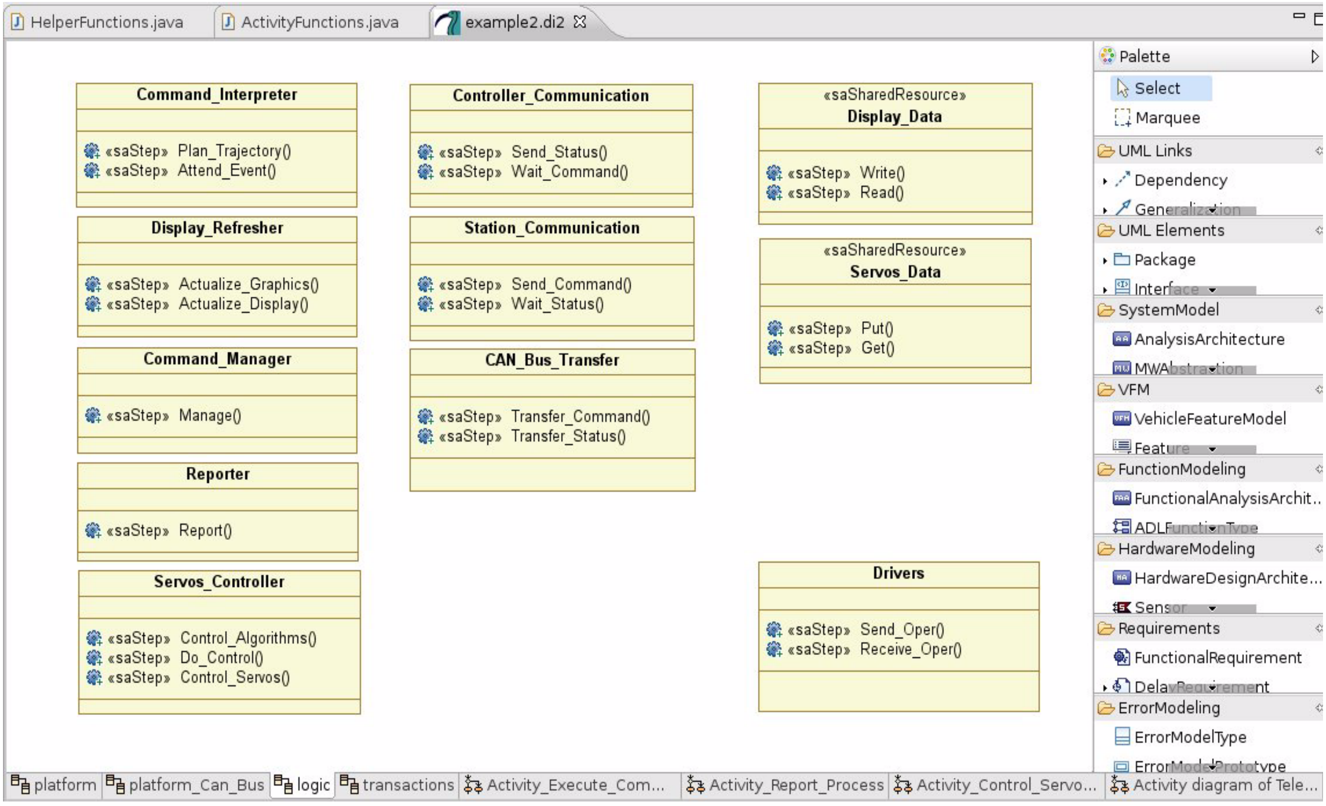

Fig. 1. View of the tool dealing with structural modeling for schedulability analysis

Other approach for timing verification is the use of specific clocks and timing annotations in MARTE and analyse them using Time Square [7]. Timing verification is pursued by using the equivalence of the behavioural models to a set of instants analized in a time-driven scheduled approach. In [4], Marcos et al. deal with modeling for verification using automated test generation and simulation tools.

3. Analysis models

The UML Profile for MARTE [3] brings a large number of modeling constructs and concepts that may be used for realizing schedulability analysis in a variety of ways. The tool here described uses a subset of those modeling constructs which is sufficient and adequate for enabling early V&V and the iterative use of the analysis models created.

In order to cope with complexity, to manage the risks associated to the research and the development of tools efforts, and also to make better use of the modeling resources offered by MARTE, we assume that the complete design/specification/analysis problem is divided in two challenging but achievable steps. One comprises the definition and manipulation of what we will denominate the "analysis models". The other one is the specification and automation of implementation oriented "design models".

This paper deals with the tool that exploits the analysis capabilities in MARTE: essentially the SAM, GQAM, and GRM chapters. In a model-driven approach, these analysis model are meant to be generated in a (semi or fully) automated way by the use of model transformations, which take the elements in the High Level Application Modeling chapter: RtUnit and PpUnit combined with the real-time features there described.

The first and more relevant problem has been the definition/selection of which elements in MARTE are to be used in the creation of the schedulability analysis models. These elements are the basis for the tool that has been developed for the generation of MAST analysis models taken from UML+MARTE annotated analysis models. Following previous research efforts [1], MARTE provides concepts to structure the analysis models using three main categories: The platform resources (a), the elements describing the logical behaviour of the system constituent parts (b), and finally the real-time situations to be analysed (c). The platform elements are modelled as a set of structural elements with stereotypes annotated on them. Figure 1 shows an example of the usage of these elements in the modeling of a tele-operated robot distributed platform. In a similar way, the logical components are modelled by means of regular

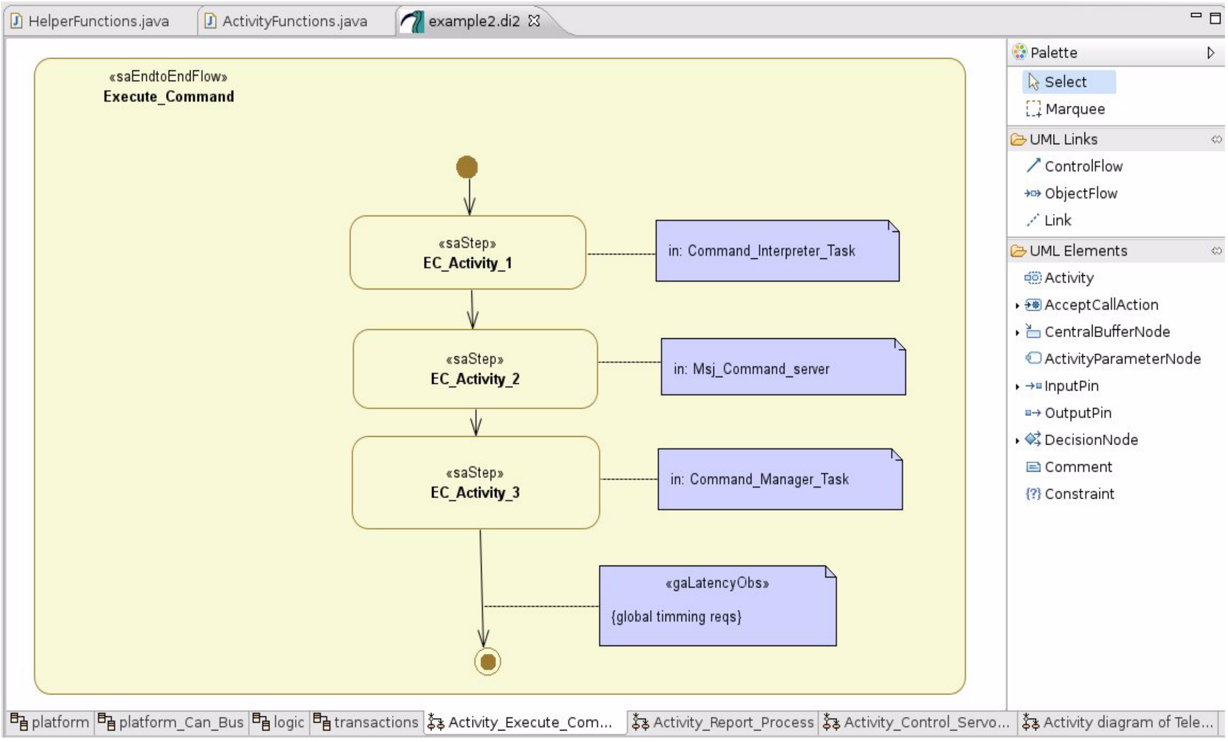

Fig. 2. Views of the tool dealing with behavioral modeling for schedulability analysis

classes and operations stereotyped as SaStep. Finally, the Real-time situation is modelled as a SaAnalysisContext. As shown in Figure 2, the end-to-end flows that described scenarios are modelled using activity diagrams.

Though the precise mapping from MARTE to MAST elements is described in the documentation accompanying the tool, here we summarize a condensed view of the MARTE elements proposed for their use in each of these three main categories.

Table 1

Platform Resources	**Behavioral Models**	**Real-Time Situations**
GaResourcesPlatform SaExecHost * SaCommHost * SaSharedResource * SchedulableResource *	GaWorkloadBehavior GaScenario SaStep * SaCommStep	SaAnalysisContext * GaWorkloadEvent * Allocate Allocated Assign SaEndToEndFlow * SaSchedObs GaLatencyObs *
* Elements used in the extraction tool in current version.		

4. Technologies used

This effort has been realized using the technologies provided by PapyrusUML as graphical tool, the UML2 plugin as model repository, and the Acceleo plugin for the extraction of text from the UML2 models plus a significant number of Java functions. For the recuperation of results the classical XML to Java technologies have been used. The code used as well as the scripts created are shared as open source. An initial version with support for activity diagrams and composition of independently characterized timed behaviours is available from http://mast.unican.es/umlmast/marte2mast.

5. Enhancements to MARTE

The issues that are in their way to be sent to the OMG are basically related to the support for hierarchical scheduling (GRM) and the distinction between the typical context switch and the one used for servicing interrupt routines.(SAM)

From the tooling perspective the modelling element that has been used for the specific purposes of the invocation of the tool is the ContextParams attribute of the SaAnalysisContext stereotype.

This field has multiplicity * and is hold in an NFP_String type. The name it has in the Papyrus implementation of the profile is “context”, which is smaller and will be suggested for enhancement. The concrete parameters defined in it are:

- “invoke”: (true or false) defines whether the automatic invocation of the MAST tool shall be done or not.

- recoverResults: (true or false) defines whether the results obtained from MAST in its results file should be inserted back into the UML model or not.
- overwriteResults: (true or false) defines whether the response times annotated in the Steps should be replaced by the new ones or appended with the “mode” of the recently executed conversion for analysis.
- ,overwriteOutputModel: (true or false) indicates whether the UML output model should be overwritten or a new one need to be created to hold the results from the previous analysis.
- modeID: (string) indicates the name that wants to be used for the “mode” attribute that will identify the values resulting from the analysis to be performed after the execution of the tool. This id is going to be used in the name of the resulting UML model file also. By default a combination of date and time of the tool execution is used as the mode of the analysis context.

These parameters will be discussed in the OMG forum for its inclusion in the next versions of the standard.

6. Conclusions and future work.

Considering the prospects of the OMG´s UML Profile for MARTE as a modelling standard for analysis tools inter operability, it seems reasonable to look for model based strategies that link it with modeling intensive activities. And a clear semantics for the High level application modeling is the basis for automating the process of having timing analysis results quickly in the development life cycle. The issues that are in their way to be sent to the OMG are basically related to the support for hierarchical scheduling (GRM) and the distinction between the typical context switch and the one used for servicing interrupt routines.(SAM)

The extraction of MAST analysis models from the UML+MARTE schedulability analysis specific models is a first demonstrable step in the direction pointed out by this effort. The tool is shared as open source with GPL license: http://mast.unican.es/umlmast/marte2mast

From the real-time and embedded systems research community perspective, this effort constitutes a step to get the effective exploitation of the capabilities of the available analysis and verification techniques, which despite the efforts in dissemination, have not yet reached an audience large enough to reward the many years of work in the field.

The next effort that need to be addressed is the required fixing in the semantics and models of computation of the RtUnits and PpUnits and the definition of the transformations in order to automate the construction of the schedulability analysis models from the restricted design models.

References

[1] J.L.Medina, M.González Harbour and J.M. Drake, Mast Real-Time View: A Graphic UML Tool for Modeling Object-Oriented Real-Time Systems, in Proc of the *22nd IEEE Real-Time System Symposium (RTSS 2001)*,pp 245-256, 2001.

[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] Object Management Group, UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE) version 1.0, OMG doc. formal/2009-11-02, 2009

[4] M. Marcos, E. Estévez, N. Iriondo, and D. Orive, Analysis and Validation of IEC 61131-3 Applications using a MDE Approach. In Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on, September 2010, Bilbao, Spain.

[5] S.Demathieu, and L. Rioux. MARTE to RapidRMA. Thales Report/Technical Document number 61565273 305 6. http://www.omgwiki.org/marte/node/31.

[6] Eric Maes and Nicolas Vienne. MARTE to Cheddar Transformation using ATL. Thales Report/Technical Documents number 61565546-179 and 61565546 108. http://beru.univ-brest.fr/~singhoff/cheddar/contribs/examples_of_use/thales_rt/MARTE2CheddarTransformationRules.pdf

[7] Équipe-projet AOSTE. TimeSquare. http://www-sop.inria.fr/aoste/dev/time_square/