TEMPO: Integrating Scheduling Analysis in the Industrial Design Practices

Rafik HENIA, Laurent RIOUX, Nicolas SORDON
Thales Research & Technology
1 Avenue Augustin Fresnel, 91767, Palaiseau Cedex, France
{Rafik.Henia, Laurent.Rioux, Nicolas.Sordon}@Thalesgroup.com

Usually, the industrial practices rely on the subjective judgment of experienced software architects and developers to predict how design decisions may impact the system timing behavior. This is however risky since eventual timing errors are only detected after implementation and integration, when the software execution can be tested on system level, under realistic conditions. At this stage, timing errors may be very costly and time consuming to correct. Therefore, to overcome this problem we need an efficient, reliable and automated timing estimation method applicable already at early design stages and continuing throughout the whole development cycle. Scheduling analysis appears to be the adequate candidate for this purpose. However, its use in the industry is conditioned by a seamless integration in the software development process. This is not always an easy task due to the semantic mismatches that usually exist between the design and the scheduling analysis models. At Thales Research & Technology, we have developed a timing framework called TEMPO that solves the semantic issues through appropriate model transformation rules, thus allowing the integration of scheduling analysis in the development process of real-time embedded software. In this demonstration paper, we present the basic building blocks and functionalities of the TEMPO framework and describe the main visible stages in the model transformations involved.

Keywords—timing verification; scheduling analysis; model-based design; model transformation

I. Introduction

It has always been a challenge to introduce scheduling analysis into the industrial development process as the inputs required for the analysis, in particular the worst-case execution time and the system behavior description, are moving target all across the different development process phases. Thanks to the introduction of model based methods (in particular viewpoints for non-functional properties) in the industrial development process, this goal seems to be reachable. Starting from very high level system architecture and rough timing allocations, the scheduling analysis has to be refined at each step of the project (architectural design, detailed design, coding, unit test and software validation phases) down to concrete timing measurements on the final system. A major problem however persists: scheduling analysis is often not directly applicable to conceptual design due to the semantic gaps between their respective models. Solving this issue is essential to break the remaining walls separating the scheduling analysis from the development process of real-time embedded systems, and to enable its use in the industry.

At Thales Research & Technology, we have therefore developed a timing framework called TEMPO allowing adapting design models to the semantic of the scheduling analysis timing models through a set of transformation rules. The transformation preserves the timing behavior modeled in the conceptual design. After performing scheduling analysis, the obtained results are, in turn, adapted back to the semantic of the design model.

In this demonstration, we present an integrated tool chain from a design modeling tool to a scheduling analysis tool via the timing framework TEMPO and show how the issue of the semantic gaps between design and scheduling analysis is solved.

II. TEMPO Framework Structure

The TEMPO timing framework that we present in this demonstration represents a contribution to the industrial exploitation of model-driven technologies and response time scheduling analysis in the design of real-time systems in a variety of application domains. The TEMPO framework structure is illustrated in Figure 1. It is composed of two building blocks (the TEMPO Design and the TEMPO Analysis pivot models) as well as a set of transformation rules between them.

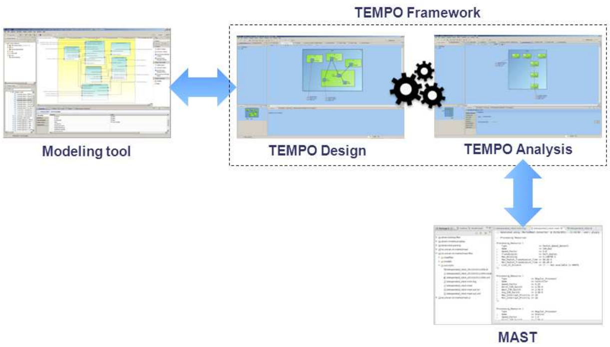

Figure 1: Tool chain including the TEMPO framework

A. *TEMPO Design Pivot Model*

The TEMPO design building block uses a subset of the UML Profile for MARTE standard [1] as a basis to represent a synthetic view of the system design model that captures all elements, data and properties that impact the system timing behavior and that are required to perform the scheduling analysis (e.g. tasks mapping on processors, communication links, execution times, scheduling parameters, etc.). TEMPO Design is not limited to the use of a particular design modeling tool and environment. It can be connected to various environments such as UML, SysML, AADL or any other proprietary environment. This was imposed by the fact that THALES divisions are using various modeling tools, languages and methodologies to design their systems.

B. TEMPO Transformation Rules

Scheduling analysis is very often not directly applicable to the conceptual design models in general and to TEMPO Design models in particular due to the semantic mismatch between the latter and the variety of scheduling analysis models known from the classical real time systems research and represented by academic [2] [3] and commercial tools [4]. For instance, in the common scheduling analysis models, a standard assumption is that a task writes its output data at the end of its execution. This is not always the case in design models. Very often in design models, operation calls are either synchronous (blocking) or asynchronous (non-blocking). As a consequence, the task, to which the caller operation is mapped, may write data into the input of a connected task, to which the called operation is mapped, at any instant during its execution and not necessarily at the end. In order to overcome the semantic mismatch between design and scheduling analysis, we have defined a set of rules transforming the TEMPO Design model into a corresponding TEMPO Analysis model, while preserving the initial modeled timing behavior.

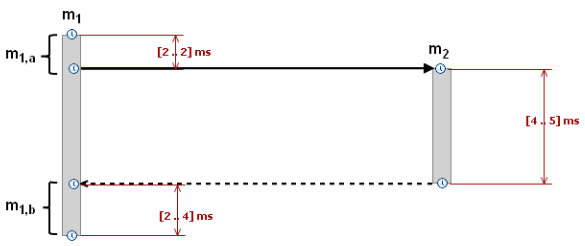

Figure 2: Synchronous call between two operations in the TEMPO Design pivot model

In the following, we present an example of a transformation rule. Figure 2 illustrates an example of a synchronous call between two operations (m_1 and m_2) in TEMPO design. Let us assume that operation m1 (composed of two operation fragments $m_{1,a}$ and $m_{1,b}$) is mapped to a task called T_1, while the operation m_2 is mapped to a task called T_2. Let us assume static priority preemptive scheduling for the tasks. Regardless of the priority assignment for the tasks, the execution order of the operations will always be the following: after its activation, task T_1 will first execute the operation fragment $m_{1,a}$. Then, it calls task T_2. Since the call is blocking, task T_1 is suspended until task T_2 finishes executing the operation m_2 and sends data back. Then, task T_1 executes the operation fragment $m_{1,b}$.

In order to keep the synchronous call behavior of the operations and tasks while being compliant with the scheduling analysis model semantic, we split the operation m_1 in two distinct operations corresponding to the operation fragments $m_{1,a}$ and $m_{1,b}$ as illustrated in Figure 3. We also split task T_1 in two tasks $T_{1,a}$ and $T_{1,b}$ that inherit its priority. Then, we to map the operations $m_{1,a}$ and $m_{1,b}$ respectively to the tasks $T_{1,a}$ and $T_{1,b}$. Obviously, this transformation preserves the same execution order and thus, the synchronous call behavior of the original operations and tasks in the system design model. In addition, it is compliant with the above mentioned timing analysis standard assumption, since task $T_{1,a}$ calls task T_2 at the end of its execution and not before as task T_1 does in TEMPO Design.

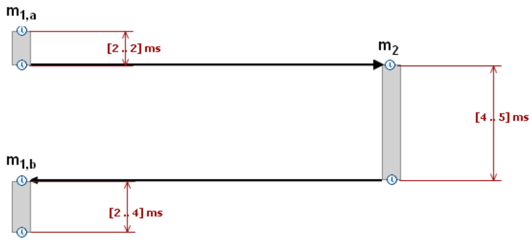

Figure 3: Transformed synchronous call between two operations in the TEMPO Analysis pivot model

C. TEMPO Analysis Pivot Model

The TEMPO Analysis pivot model is based on generic modeling concepts known from the classical real time systems research, such as tasks, processors, busses, scheduling parameters (priorities, time slots, deadlines, etc.). TEMPO Analysis models preserve the timing behavior modeled in the corresponding TEMPO Design models, while ensuring the compatibility with the variety of existing scheduling analysis tools. As for TEMPO Design, TEMPO Analysis is not limited to a specific scheduling analysis tool. This ensures a minimum of independence from the analysis tools specificities and allows hiding its complexity to the designer. If required, the used analysis tool can be easily replaced by another. After analysis in the selected scheduling analysis tools, the results are injected in TEMPO Analysis. Then, they are translated to be compliant with the original design model and injected in TEMPO Design

III. Demonstration

Several practical use cases are available as hands-on demonstration of the quality of the TEMPO framework. 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].

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] PyCPA: Compositional Performance Analysis in Python ; https://code.google.com/p/pycpa/

[4] SymTA/S: Symbolic Timing Analysis for Systems; https://www.symtavision.com/symtas.html

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