MARTE for time modeling and verification of real-time embedded system

Marie-Agnès Peraldi-Frati, Frédéric Mallet, Julien Deantoni,
I3S Laboratory
CNRS, University of Nice Sophia-Antipolis, INRIA
Sophia-Antipolis, France
map@unice.fr, frederic.mallet@unice.fr, julien.deantoni@sophia.inria.fr,

Abstract— This paper presents the different steps of a methodology for modeling real-time requirements and ensuring their validation and their traceability over a design flow for embedded system design. A specific meta-model for temporal constraints modeling is proposed that covers the needs for traceability and timing analysis issues. The timing architecture modeling, based on the conjoint use of the EAST-ADL2 and MARTE allows a precise semantics of timing annotations and establishes the link with validation models.

Keywords-component: Timing requirement, Time modeling, MARTE, Validation, Embedded application;

I. INTRODUCTION

Timing requirement modeling and analysis is a key issue in a design flow for electronic embedded systems. Industrials from IT have proposed and developed standards and engineering tools which partially cover these needs as they focus on functional requirements. Moreover the link between the initial expression of requirements and their impact on solution models is not fully ensured. In particular, the traceability of timing requirements raises some others specific issues such as the way to express and distinguish a timing requirement in a set of requirements; the ability for models to express timing properties, the link with validation tools for the test or the timing analysis of models and finally, the feedback of the analysis results on the design flow.

This paper presents a methodology based on MARTE [1] and EAST-ADL2 [2] that proposes a solution for the modeling of temporal features from the initial requirements in a specification document down to the model and the final product. We discuss and illustrate some methodological aspects on using MARTE at the different steps of a modelling approach that integrates both functional and non-functional modelling. Another contribution is to show how it is possible to exploit these models by extracting the temporal information and the implementation characteristics in order to provide a schedulability simulation or analysis. Traceability links interconnect these elements and the associated tools for their verification and their validation.

The paper is structured as follows: in section II, we introduce the main issues for modeling non functional requirements and timing constraints in real-time embedded systems. Section III presents a case study used for illustrating the methodology. Section IV describes the modeling and the classification of timing requirements and their link with the solution model. Section V presents the modeling of temporal behavior using EAST-ADL2 and MARTE, and Section VI illustrates the validation of the requirements and the temporal constraints by using formal approaches based on scheduling analysis.

II. TIME MODELING IN EMBEDDED SYSTEMS

The modeling of non functional and timing requirements remains a challenging problem in embedded system design approaches such as those used in system on chip design, automotive or avionic applications. In such domains, design process should comply with safety standards (CMMI level 5 and the ISO26262) which impose the full management and traceability of requirements.

In the set of non functional requirements, timing constraints are due to complex interactions between different functions that communicate within a mono-processor or multi-processors or through embedded networks. Processors evolutions are triggered on a specific internal clock, the network speed is also based on the communication controller clock. On each processor, functions are implemented by OS tasks which execution duration may vary depending of the OS scheduling policy. These different parameters induce an important issue for verifying the functional and non functional timing behavior of the overall system. Such system cannot be analyzable without a clear abstraction and modeling of the timing behavior of the system.

Last but not least, embedded system design for SOC and automotive covers multi-disciplinary fields composed of analogical/digital software and electronic hardware. By the way, the design flow includes heterogeneous models coming from various tools and languages.

For example in the SOC design domain, operational tools use IPXact [3] standard for expressing the structural modeling of SOC and development platform such as Synopsys Innovator® platform for interconnecting intellectual properties (IP) and generating SystemC and RTL synthesis code.

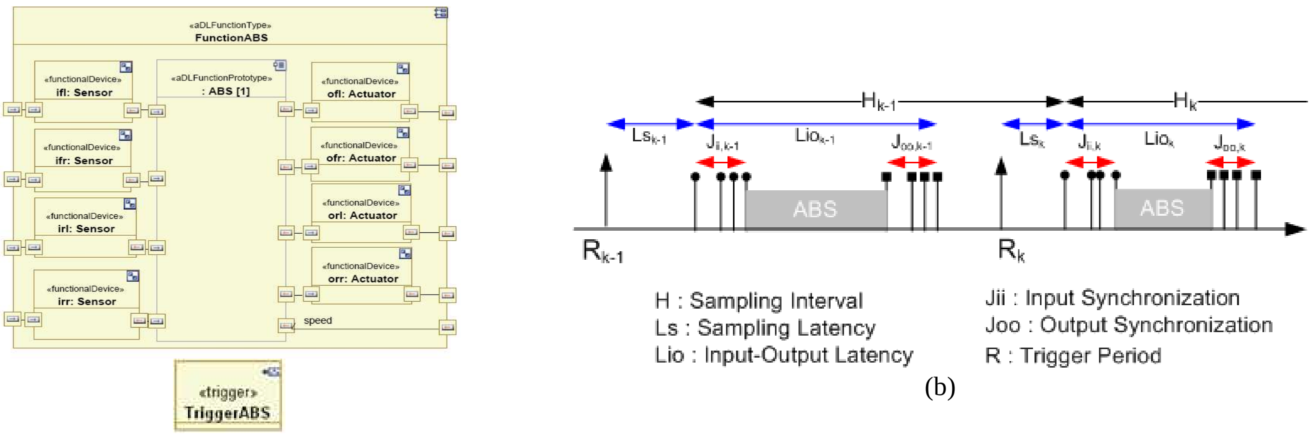
In the automotive domain, operational teams use Telelogic DOORS® and Geensys Reqtify® industrial tools for supporting the traceability. The architecture description language EAST-ADL and AUTOSAR [4] are used for the structural and behavioral modeling of systems dealing with the traceability aspects. Additional tools are used such for MathWorks Matlab/Simulink® for analogical system modeling and simulation.

Some of these languages are UML–centric. Other tools dedicated to the temporal analysis of systems are based on well founded theory widely used in the domain of real-time control systems. The highest challenge for a new method and tool in the electronic system engineering domain is to tackle all these aspects i.e. the heterogeneity at the different abstraction levels i.e. from high level abstraction model to effective implementation, and to consider automatic transition/transformation of models between these abstraction levels. The analysis of validation aspects

The recent OMG standardization of the MARTE profile (Modelling and Analysis of Real-Time and Embedded systems) is an important step for modelling non-functional characteristics of real-time embedded systems. An important contribution of MARTE lies in its time model, centred on the notion of multiform time (logical or chronometric), that enriches UML with explicit time model elements (clocks, clocks type …). This is a real advance in regard to the SPT [5] (Scheduling Performance and Time) profile which considers time as an implicit notion closed to the physical time, modelled by a simple annotation onto UML models. In addition, MARTE provides two other models for describing execution platforms, and the allocation of application functions onto the resources of the platform.

III. CASE STUDY

We illustrate the methodology on an Anti Blocking System (ABS). The ABS architecture consists of four sensors, four actuators and an indicator of the vehicle speed. The sensors measure the rotation speed of the vehicle wheels. The actuators indicate the brake pressure to be applied on the 4 wheels. Precise timing requirements are imposed on the ABS function such as the latency of sensor sampling (Ls) and the latency of the function execution (Lio), A trigger period for the function is define (R) and an interval delay for inputs and outputs must be respected (Jii Input Synchronization, Joo Output Synchronization). All these timing requirements are modeled at the different step of the methodology and at the different level of the design flow.



(a) (b)

Figure 1: The Anti-Blocking System example

The functional architecture, presented on Figure 1.a, is composed of an `ADLFunctionType` for the functional part of the ABS and `functionalDevice` for sensors and actuators modeling. An `ADLOutFlowPort` provides the vehicle speed. A `trigger` models the periodic triggering of the ABS. The dynamic execution of the ABS is presented on Figure 1.b. The ABS is triggered by the `R`. parameter and the `Ls` value measures the latency of sensor sampling. The values of the four sensors involved in the ABS must arrive on the input `ADLFlowPorts` within delay `Jii` (InputSynchronization). A similar OuputSynchronization delay `Joo` is represented on the output interface side. The `Lio` represents the delay from the first event on the input set of the ABS until the last event occurrence on the output set. The sampling interval of the sensor is given by parameter `H`. These parameters are modeled by timing requirements characterized by timing values or intervals with jitters.

IV. A DESIGN FLOW FOR TIME MODELING AND ANALYSIS.

We propose a methodology that integrates in common framework, domain specific languages such as EAST-ADL2, AUTOSAR, the UML profiles MARTE and SysML[6] and the heterogeneous analysis tools such as Timesquare[7] or SynDEX[8] for scheduling simulation and analysis. We explore the traceability aspect of the methodology to propose a seamless flow for timing requirement and constraints management.

An overview of the design flow, illustrated on Figure 2., handles the full traceability link from the initial customer requests (i.e. the requirement specification documents) down to their modeling and their implementation.

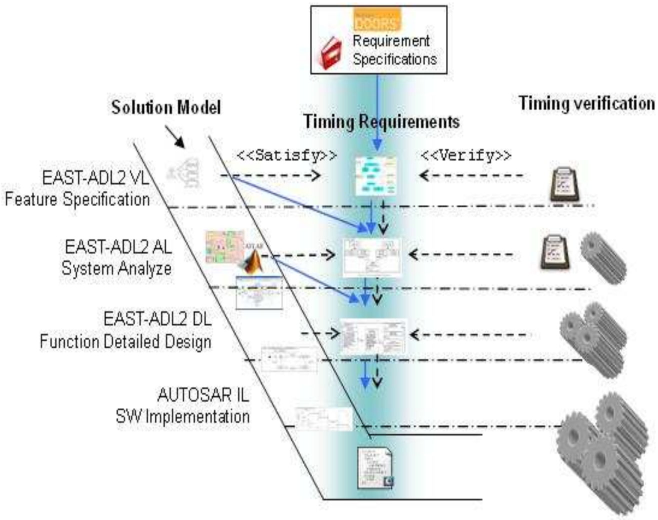
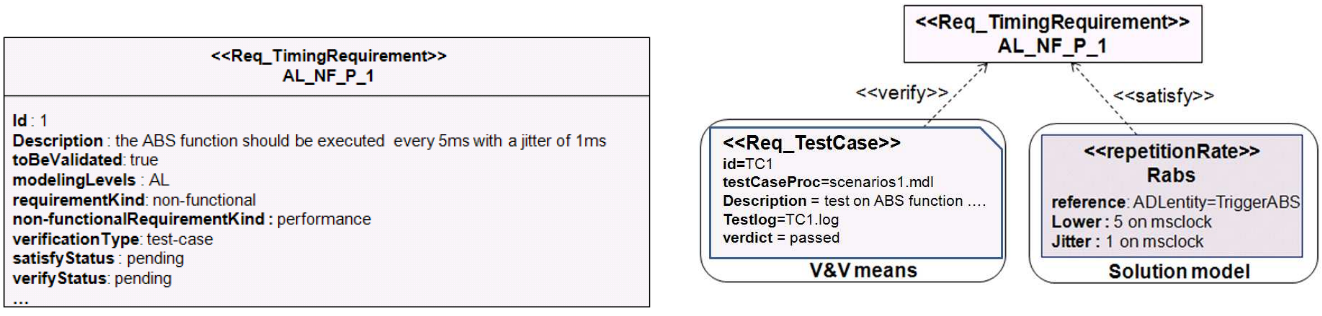


Figure 2 Overview of the proposed timing approach

This work ensures at each level of abstraction of the system, from functional analysis down to software implementation, a relation between requirements and the design models. This method targets to run backward and forward analysis during a change request, either on requirement or in solution (code or design specifications). The tool chain ensures and identifies which elements of the solution is impacted and needed to be updated or verified again. ISO26262 recommends the full requirement traceability, meaning from request to solution and test. The methodology answers to this request and propose a complementary usage of such tool for industrials using massively product line approach.

A. Importing initial timing requirement

In the approach, the initial requirements are expressed by using the EAST-ADL2 requirement package extended with some features for classification and traceability [9]. By this way, in a set of requirements, a non functional requirement can be easily spotted because of the use of the specific stereotype showed on **Figure 3**.a.



(a) **(b)**

Figure 3. Example of a timing requirement

The description part is expressed in a natural language (description field). Verification and analysis procedures of the requirement are defined in the verificationType field (code inspection, test-case, OCL expressions checking or real-time analysis). The result of such verification applied on the solution model or on the final product impacts the properties value of the initial requirement (satisfyStatus and verifyStatus). This reporting becomes possible because of the traceability features.

B. Modeling timing requirements

The EAST-ADL2 language defines the concept of `timingRequirement` that allows a formalization of the requirements across the solution model. A `timingRequirement` can express either a duration constraint (`delayRequirement`) between input(s) and. output(s) port(s) or a period (`repetitionRate`*)* on a port or on a trigger. It can be also an `input` *(*resp. `output`*)* `Synchronization` between a set of `input` (resp. `Output`) ports. Nevertheless, these timing values are not linked to explicit clocks in the system. This is a real problem for automotive system design because several clocks can be considered in the same function. In the ABS example, the wheel speed value is expresses in RPM (round per minutes) whereas the sensors acquisition period value is in millisecond. In our approach, we use the time concepts of MARTE conjointly to the EAST-ADL2 modeling elements leading to `timingRequirements` elements linked to logical or chronometric clock units. The figure 3.b gives the example of timing requirement (`repetitionRate`) with duration values linked to the clock `msclock`. The expression `5 on msclock` is not a simple textual annotation but a MARTE `timedValueSpecification` on which timing analysis became possible.

C. Traceability with SysML requirement profile

We explore the SysML traceability links (*hierarchy*, *derive*, *satisfy*, *trace*) between the requirements and the solution model to interconnect the modeling elements of the requirement model, the solution model with the temporal information, and the validation and verification procedure and tools. These relations are useful to identify that the requirement `AL_NF_P_1` (cf. **Figure 3**.a) is ensured by a modeling element in the system and consequently what part of the system is affected by changes on it.

The *verify* relation links the requirements to the validation and verification elements (V&V means) for example on. Figure.3.b, a test-Case scenario is used to validate `AL_NF_P_1`.

V. FUNCTIONNAL AND TIME MODELING

We adopt the recent UML-MARTE profile and particularly its timing model to capture the various forms of repetitive event that trigger the functional parts of the ABS, and to model them at a high level of abstraction. As it is shown in section IV.B, the ABS temporal behaviour is linked to multiple clocks `RPM`, `ms`. In the initial requirements values for the period and the deadline are expressed with these different units. The semantic attached to these multiform time (clocks, period, deadline, jitters…) makes it possible to transform this high level model of time to the real time.

UML MARTE for expressing time is used conjointly with EAST-ADL2 which deals in particular with the structural modeling.

A. *Structural modeling*

EAST-ADL2 allows a structural and hierarchical decomposition of functionality with `ADLFunctionType/prototype` and `ADLFlowPorts` that carry the data and the control between these functions. Temporal information is attached to these elements such as the trigger period and the execution time of a function or the period of an `ADLFlowPort`. Figure 3.b gives an example of structural modeling elements with a `repetitionRate` modeling element that provides the constraints of the trigger for the ABS. This element of the solution model satisfies a temporal requirement given on Figure 3.a. Because we apply the same MARTE time concepts we have used for requirements modeling; the timing properties of the model element can be compared with the timing value of requirements. Equation (1) reads that the period attribute value of the TriggerABS in the solution model should be within the interval given by the `lower` and `jitter` attributes values of the requirements `Rabs`.

$$\text{TriggerABS.}\mathit{period} \in [\{\text{Rabs.lower- Rabs.jitter}\}, \{\text{Rabs.lower+Rabs.jitter}\}] \qquad (1)$$

This equation that uses CVSL (Clock Value Specification Language) of MARTE can be checked by an evaluation tools.

B. *Behavioral modeling*

The behavior of a structural `ADLFunctionType` can be described with a UML2 behavioral (UML2 Native) or a domain specific modeling language (External Behavioral).

Native behavior models can be either a UML2 activity diagram, a STATECHARTS, or a timing and sequence diagram. These diagrams are used to refine the structural model by giving a dynamic view of data and control flows. We extend these behavioral diagrams by applying a MARTE stereotype (`TimedProcessing`) on it. As shown on Figure 4, this stereotype allows temporal annotations conform to requirement and structural timing modeling. We associate to the ABS a temporal characteristic such as its period and a timing constraint such as a deadline. For the sake of visibility we have represented these constraints at the bottom parts of the actions and the activity. In an actual model using tools, these constraints are modelled with `timedValueSpecification` expressions which are elements of the design.

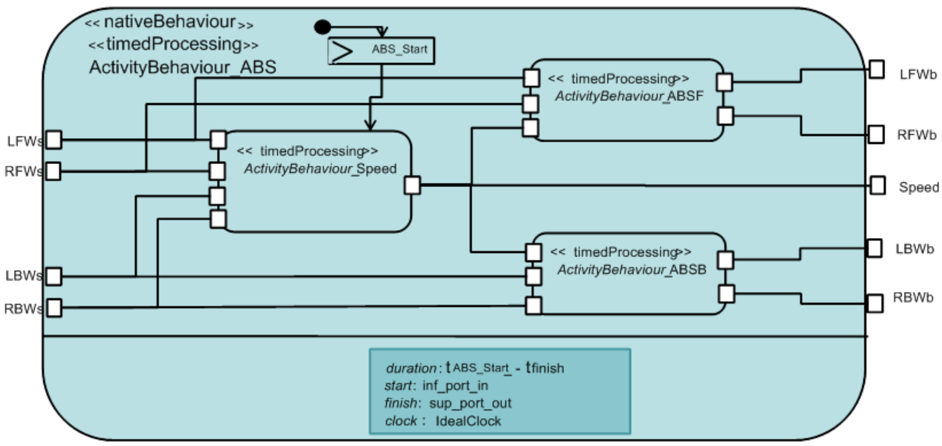


Figure 4: activity diagram for the ABS

External behavioral models are models out of the scope of UML. SIMULINK [10][11] can't be ignored as model for expressing the system behavior. The EAST-ADL2 `ExternalBehavior` stereotype gives it possible to associate heterogeneous formalisms with EAST-ADL2. In this context, ensuring the traceability of temporal requirements through heterogeneous models is a key issue. The skeleton of the SIMULINK structural part can be inherited from the EAST-ADL2 structural model. Some temporal parameters like the trigger timing parameter, the acquisition period of sensors and the period on ports can also be extracted and imported into the SIMULINK model.

VI. VALIDATION AND VERIFICATION PROCESS

The final step of the approach aims at validating and verifying the timing requirements with respect to the timing behavior. Starting from the requirement modeling phase presented in section IV.B, allowing the integration of non-functional properties in requirements and solution models, developers may use either analysis tools for supporting a systematic verification activity. Such verification can be applied at the different abstraction levels of the design for detecting unfeasible real-time architectures, i.e. a non conformance of the scheduling policies with respect to the period and the deadline of the different application tasks, or validate an implementation with respect to performance criteria. Two kinds of verification are applied: verification on the solution model and verification of the final product (the effective implementation).

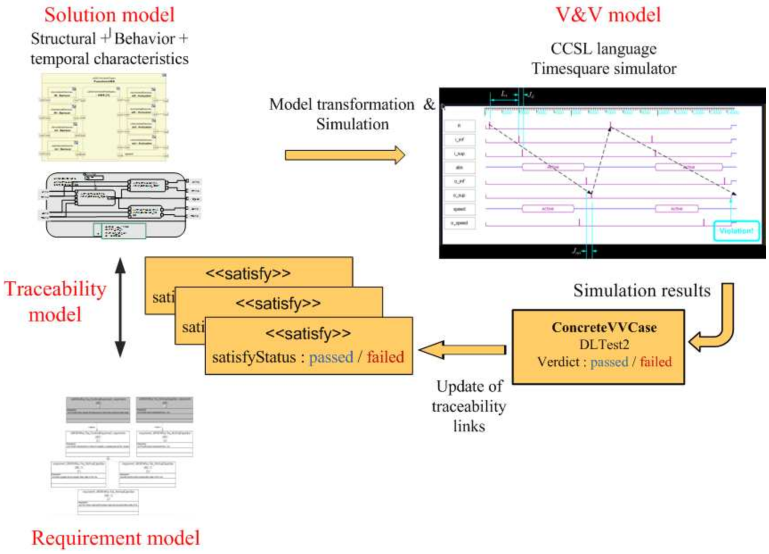


Figure 5: Validation and verification flow

Figure 5 shows the different elements used for this V&V process. The non-functional requirements model contains the timing objectives to be verified. These needs are taken into account by the solution model through the structural and behavioral modeling elements endowed with the MARTE timing features. Then, a model transformation from this solution model to a V&V model is performed. Details on the transformation rules are presented in [12]. The EAST-ADL2 and MARTE timing information are transformed into a TIMESQUARE simulation model. The result of such simulation impacts the verdict of an abstract V&V case targeting the system scheduling. In this case, the simulation result demonstrates that the trigger period is not compatible with the sampling period and the function latency values

The timesquare simulation result impacts the traceability links which are endowed with *status* that change after the completion of the verification procedures. Coming back to Figure.3.b we see that the test case TC1 should verify the solution model. This test-case become conclusive (verdict==passed), so the *status* of its *verify* link switches to "*passed*". If all V&V procedures linked to a requirement are conclusive, then the *status* of the satisfy link will be positioned to "verified". These different cases are listed in the methodology.

VII. CONCLUSION

This paper presents and illustrates a solution for timing requirement modeling and validation in an automotive design flow. The initial requirements are extracted from the stakeholders document. MARTE extensions for time modeling have been applied to structural and behavioral solution models for ensuring time flow consistency. The different links with the validation and verification means have been illustrated and the impact of these validations onto the traceability links has been explained. The overall method has been applied on of a simplified ABS. Future improvements can be done that concern the traceability over heterogeneous models in this design flow and their integration.

References

[1] The ProMARTE consortium, UML profile for MARTE, beta 2, ,June 2008, OMG document number : ptc/08-06-08

[2] EAST-ADL:The EAST-EEA Architecture Description Language, June 2004. ITEA Project Version 1.02.

[3] The spirit consortium : IPXact, http://www.spiritconsortium.org

[4] http://www.**autosar**.org/

[5] OMG. UML Profile for Schedulability, Performance, andTime Specification, January 2005. OMG document number: formal/05-01-02 (v1.1).

[6] OMG. *Systems Modeling Language (SysML) Specification*, April 2006. OMG document number: ad/2006-03-01

[7] J. DeAntoni, F. Mallet, C. Andre. TIMESQUARE : on the formal execution of UML and DSL models. April 2008, Tool session of the 4th model driven development for distributed real time systems. http://www.mdd4dres.info/_tools/1f3d0539532e6396ad1ecadc4d363a9a

[8] SYnDEx: www-roc.inria.fr/syndex.

[9] H. Dubois, M.-A. Peraldi-Frati, F. Lakhal, *A model for requirements traceability in an heterogeneous model-based design process* . In 15th IEEE Int. Conf. on Engineering of Complex Computer Systems ICECCS2010 Oxford UK, March 2010

[10] Simulink www.mathworks.com

[11] J. Shi, M. Törngren, et al.. Combined usage of UML and Simulink in the Design of Embedded Systems: Investigating Scenarios and Structural and Behavioral Mapping. In Proc. of OMER 4 workshop on Object-oriented modeling of embedded real-time systems, Oct. 30-31,

[12] F. Mallet, M.-A. Peraldi-Frati and C. Andre, "Marte CCSL to execute East-ADL Timing Requirements" IEEE Int. Conf. on Object/Component/Service-oriented Real-Time Distributed Computing, 17-*20 March, 09 Tokyo, Japan , pp249—253*