

Model Based Architecting and Construction of Embedded Systems (ACES-MB 2010)

Stefan Van Baelen¹, Iulian Ober², Huascar Espinoza³,
Thomas Weigert⁴, Ileana Ober⁵, and Sébastien Gérard⁶

¹ K.U. Leuven - DistriNet, Belgium

`Stefan.VanBaelen@cs.kuleuven.be`

² University of Toulouse - IRIT, France

`Iulian.Ober@irit.fr`

³ Tecnalía, Spain

`Huascar.Espinoza@tecnalia.com`

⁴ Missouri University of Science and Technology, USA

`weigert@mst.edu`

⁵ University of Toulouse - IRIT, France

`Ileana.Ober@irit.fr`

⁶ CEA - LIST, France

`Sebastien.Gerard@cea.fr`

Abstract. The third ACES-MB workshop brought together researchers and practitioners interested in model-based software engineering for real-time embedded systems, with a particular focus on the use of models for architecture description and domain-specific design, and for capturing non-functional constraints. Twelve presenters proposed contributions on metaheuristic search techniques for UML, modelling languages and mappings, model based verification and validation, software synthesis, and embedded systems product lines. In addition, a lively group discussion tackled these issues in further detail. This report presents an overview of the presentations and fruitful discussions that took place during the ACES-MB 2010 workshop.

1 Introduction

The development of embedded systems with real-time and other critical constraints raises distinctive problems. In particular, development teams have to make very specific architectural choices and handle key non-functional constraints related to, for example, real-time deadlines and to platform parameters like energy consumption or memory footprint. The last few years have seen an increased interest in using model-based engineering (MBE) techniques to capture dedicated architectural and non-functional information in precise (and even formal) domain-specific models in a layered construction of systems.

MBE techniques are interesting and promising because they allow to capture dedicated architectural and non-functional information in precise (and even formal) domain-specific models, and they support a layered construction of systems,

in which the (platform independent) functional aspects are kept separate from architectural and non-functional (platform specific) aspects, where the final system is obtained by combining these aspects later using model transformations.

The Third Workshop on *Model Based Architecting and Construction of Embedded Systems* (ACES-MB 2010) brought together researchers and practitioners interested in all aspects of model-based software engineering for real-time embedded systems. The participants discussed this subject at different levels, from requirements specifications, model specification languages and analysis techniques, embedded systems product lines, model synthesis, to model based verification and validation.

2 Workshop Contributions

The keynote [2] was given by Prof. Lionel C. Briand from the University of Oslo and the Simula Research Laboratory, Norway, who discussed the use of metaheuristic search for the analysis and verification of UML models.

There is a growing research activity around the use of metaheuristic search techniques (e.g., genetic algorithms) in software engineering, for example to support test case generation, often referred to as search-based software engineering (SBSE). Several years of research have focused on using metaheuristic search to support the analysis and verification of UML models and its extensions such as MARTE and OCL. Examples include the analysis of real-time deadlines (schedulability analysis), concurrency problems, and constraint solving, for example for supporting model-based test case generation. Results suggest that applying metaheuristic approaches to these problems lead to practical and scalable solutions that rely solely on UML and extensions, and does not require translations into other languages and formalisms.

6 full papers and 5 short papers had been accepted for the workshop, see [1]. A synopsis of each presentation is given below. Extended versions of articles [5] and [6] are included in this workshop reader.

[3] presents a MARTE to AADL mapping that is valuable for MARTE users in order to enable the use AADL analysis tools on MARTE models. For example, CAT, the Consumption Analysis Toolbox, allows for system-level power and energy consumption estimation for AADL models.

[4] addresses the problem that real-time embedded software today is commonly built using programming abstractions with little or no temporal semantics. The paper discusses the use of an extension to the Ptolemy II framework as a coordination language for the design of distributed real-time embedded systems. Specifically, the paper shows how to use modal models in the context of the PTIDES extension of Ptolemy II.

[5] uses UML Interaction Overview Diagrams as the basis for a user-friendly, intuitive, modelling notation that is well-suited for the design of complex, heterogeneous, embedded systems developed by domain experts with little background on modelling software based systems. To allow designers to precisely analyse models written with this notation, a part of it is provided with a formal

semantics based on temporal logic, upon which a fully automated, tool supported verification technique is built.

[6] argues that system development and integration with a sufficient maturity at entry into service is a competitive challenge in the aerospace sector, and can only be achieved using efficient model-based techniques for system design as well as for system testing. Building on the general idea of model-based systems engineering, an integrated virtual verification environment for modelling systems, requirements, and test cases is proposed, so that system designs can be simulated and verified against the requirements in the early stages of system development. The paper exemplifies its application in a ModelicaML modelling environment.

[7] addresses the early validation of automobile electronic systems by providing a transformation of EAST-ADL models to SystemC at different layers of abstraction. This allows specific analysis with hardware-software co-simulation iteratively in the development process. The proposed approach is realized in a tool chain and demonstrated by an automotive use case, showing the potential of an early validation of system and software designs based on architecture models.

[8] argues that modelling tools should become development environments and support a methodologically guided development in which milestones are indicated and warnings are generated to inform the user about issues that are to be solved to reach these milestones. The paper indicates model maturity levels that correspond to an underlying development method and shows in the model maturity view which elements or parts of the model do not yet reach a certain level and why.

[9] proposes to abstract away from architectural platforms and their induced architectural styles to more abstract representation of applications. Architecture-independent application models, developed using modern model-based development techniques, can be mapped to application architectures in a variety of architectural styles. Architectural mappings therefore play an important role in synthesis of software implementations from abstract application models.

[10] proposes to integrate multiple partially overlapping models from different tools, since each tool and associated modelling language have different strengths and weaknesses. It is crucial that relevant dependencies between models and related timing properties are explicitly captured, allowing the analysis of the impact of changes on the timing properties and timing requirements. The paper proposes to use the concept of megamodels as a solution for the support of those dependencies relevant for timing properties, so that no violation may remain undetected.

[11] presents a model of an evolutionary product line process based on architecture transformations. The model attempts to give an accurate description of how real architects actually work. Key elements of the approach are how the transformations interact with consistency constraints and with feasibility in terms of resource limitations.

[12] proposes an approach for the identification of features supported by class models annotated with stereotypes. The models are automatically reverse engineered by a tool called Rejasp/Dmasp where attributes and methods are

stereotyped if they have some relation with candidate features. The approach consists of four guidelines and focuses on identifying features in embedded systems for ground vehicles.

[13] states that much meaning can be given to a model using a domain specific language (DSL), and the code generation rate can be increased. Model-based product line development is possible using code generation to realize variability. The paper presents a case study where a high rate of code generation was achieved by using two DSLs, the characteristics of which supplement each other. Structure is described by a highly general DSL and behaviour by a specialised DSL. Various kinds of products have been developed from a product line efficiently by using code generation from DSLs to realize variability.

3 Summary of the Workshop Discussions

The workshop was divided into 3 sessions: modelling languages and mappings, verification and validation, and a position statement session. After each session, a group discussion was held on issues raised during the session presentations. The following integrates and summarizes the conclusions of the discussions.

Mappings between modelling languages

An important issue for mappings between modelling formalisms concerns the level of detail that can be expressed in each formalism and the ability to transform information from one modelling formalism into another. One viewpoint is to consider a mapping between modelling formalisms as a dedicated transformation for a specific purpose, e.g., model analysis in the case of the MARTE to AADL mapping. This means that a mapping can abstract away certain details from the source model that are useless for the transformation purpose. In addition, specific information can be refined or added to the target model by enlarging the target model or by using an in-between Platform-Specific Model (PSM) profile on top of the source model, e.g., an AADL profile for MARTE. OCL constraints can be added to the PSM profile in order to validate the correctness of the added information. In order to bridge the semantic gap between the two modelling levels, abstraction patterns can be introduced, e.g., by creating user-defined extensions for AADL.

Constructing architectural models

The construction of architectural models raises a number of issues, such as dealing with a multitude of inter-model dependencies and coping with model changes and the ripple effects they can cause. In addition, a product line approach can be beneficial but even further complicates the architectural modelling phase, since one should focus on an architecture for the whole product line instead of for a single product. When generative techniques are used, there was a common agreement on never touching the generated models or code. If changes seem necessary, either the source model or the generator itself should be changed.

Using a megamodel approach

Although the use of megamodels was recognised as a need in order to support different modelling formalisms, it was not clear if one should stay in the same technological space or not. It can be difficult to arrive at a combined megamodel. Therefore, it is sometimes better to use transformations or filters to combine different models rather than try to squeeze them into a single megamodel.

Acknowledgements

This workshop was supported by the IST-004527 ARTIST2 Network of Excellence on Embedded Systems Design (<http://www.artist-embedded.org>), the research project EUREKA-ITEA EVOLVE (<http://www.evolve-itea.org>), the research project EUREKA-ITEA VERDE (<http://www.itea-verde.org>), the research project EUREKA-ITEA OPEES (<http://www.opees.org>), and the research project ICT FP7-INTERESTED (<http://www.interested-ip.eu>).

References

1. Van Baelen, S., Ober, I., Espinoza, H., Weigert, T., Ober, I., Gérard, S. (eds.): Third International Workshop on Model Based Architecting and Construction of Embedded Systems. CEUR Workshop Proceedings Vol.644, CEUR, Aachen, Germany (2010)
2. Briand, L.C.: Using metaheuristic search for the analysis and verification of UML models. In: [1], p. 9
3. Turki, S., Senn, E., Blouin, D.: Mapping the MARTE UML profile to AADL. In: [1], pp. 11–20
4. Eidson, J.C., Lee, E.A., Matic, S., Seshia, S.A., Zou, J.: A time-centric model for cyber-physical applications. In: [1], pp. 21–35
5. Baresi, L., Morzenti, A., Motta, A., Rossi, M.: From interaction overview diagrams to temporal logic. In: [1], pp. 37–51
6. Schamai, W., Helle, P., Fritzsion, P., Paredis, C.J.J.: Virtual verification of system designs against system requirements. In: [1], pp. 53–67
7. Weiss, G., Zeller, M., Eilers, D., Knorr, R.: Approach for iterative validation of automotive embedded systems. In: [1], pp. 65–83
8. Grosse-Rhode, M.: Model maturity levels for embedded systems development, or working with warnings. In: [1], pp. 85–99
9. Bagheri, H., Sullivan, K.: Towards a systematic approach for software synthesis. In: [1], pp. 101–105 (2010)
10. Neumann, S., Seibel, A.: Toward mega models for maintaining timing properties of automotive systems. In: [1], pp. 107–111
11. Axelsson, J.: A transformation-based model of evolutionary architecting for embedded system product lines. In: [1], pp. 113–117
12. Durelli, R.S., Conrado, D.B.F., Ramos, R.A., Pastor, O.L., de Camargo, V.V., Penteadó, R.A.D.: Identifying features for ground vehicles software product lines by means of annotated models. In: [1], pp. 119–123
13. Tokumoto, S.: Product line development using multiple domain specific languages in embedded systems. In: [1], pp. 125–129