

MODEL DRIVEN DEVELOPMENT OF PROCESS CONTROL SYSTEMS USING UML PROFILES MARTE AND SYSML

M.Sc. I. Antonova, Prof. M.Sc. I. Batchkova

Dept. of Industrial Automation, University of Chemical Technology and Metallurgy
Bul. Kl. Ohridski 8, Sofia, Bulgaria

Abstract: Development of effective software application for real-time and embedded systems requires new advanced methods and techniques. This paper presents a model driven approach for development of process control systems based on the UML profile for system engineering – SysML and MARTE profile for analysis and modeling of real-time systems. The suggested approach is illustrated with a simple example for development of tank level feedback control system. Finally some conclusions are made.

Keywords: UML, SysML, MARTE, MDD, process control

1. Introduction

The considerably growth in the size and complexity of the software in control systems as well the increased need of their dependability and quality require a growing penetration and adoption of the approaches, methods and tools of software engineering in the field of automation and control systems. Some of the most promising and challenging approaches are these of Model Driven Engineering (MDD) and Model Driven Engineering (MDE) [1], where the systems are presented as models that conform to meta-models, and the model transformations is used to manipulate the various representations. MDD and MDE are generic terms which are used interchangeably. Using these approaches is an important step towards achieving the significant features of modern control systems, such as integrity, security, reliability, interoperability and portability.

Model Driven Architecture (MDA) [2] is a remarkable MDD initiative of Object Management Group (OMG), consisting in transformation of different platform independent models towards executable applications. In the core of MDA are the open standards, UML, MOF, XMI, etc. UML [3] does not specify a methodology for model driven software or system development but aims to provide an integrated modeling framework, covering structural, functional and behaviour descriptions. The UML notations support the development of various diagrams that reflect different aspects of the system in order to capture the full complexity in the phases of detailed analysis and system design. In the last years there has been an increasing striving to use UML in control, automation, and industrial enterprise engineering. There are many working groups whose research activities are directed in filling the gap between state of the art in software engineering and state of the practice in the control domain. They use different approaches and techniques in order to extend the object-oriented software development and especially UML for development of real time systems [4]. The main drawbacks of all these extensions are connected with the difficulties in modeling and analysis of the closed loop systems. In order to overcome these drawbacks, in this work, the UML profiles for System Engineering (SysML) [5] and Modeling and Analysis of Embedded Real Time Systems (MARTE) are used [6].

The paper is organized in 4 parts. After the introduction, in part 2 a short overview of the UML profiles for development of control systems is given as one of the most effective and useful extensions of UML for modeling of hard real time systems. In part 3 the suggested approach for development of process control systems is described. The last part of the paper illustrates the approach with modeling of simple process control system for water level feedback control. Finally some conclusions are made.

2. Short overview of UML profiles supporting control system design

In recent years we have witnessed an increased interest in the use of object-oriented approaches in the field of software development, which is associated with a reduction of its complexity

and achieving modularity, portability, interoperability and reusability, as well as opportunities for modifications and extensions [7, 8].

The main challenges towards real-time UML are connected with the creation of different mechanisms to handle real-time features such as: models of physical time, timing specifications, timing facilities, modeling and management of physical resources and concurrency. Another important issue is the development or use of means for early verification and validation of the designed systems in respect not only to their functionality, but also in respect to the non-functionality requirements.

There are many different proposals for extending UML to support the design and analysis of control systems. Short overview and analysis of the frequently used is made in [4]. One of the most popular approaches for this purpose is based on the development of different profiles UML. A profile is a restricted form of a meta-model that must always extend some reference meta-model created from MOF, such as UML. Creating UML profiles (standard and specific) on the base of stereotypes, constraints and tagged values. The 3 built-in extension mechanisms can be used separately or together. The objective of UML profiles is to package specific terminology and substructures for a particular application domain. One of the first attempt to provide RT capabilities of UML in this direction is the OMG initiative for creating of the profile for Schedulability, Performance, and Time Specification (SPT-profile) that is proposing a framework to model quality of service, resource, time and concurrency concepts in order to support predictive quantitative analysis of the UML1.4 models. This profile supports two well-established forms of time based model analysis: schedulability analysis based on schedulability theory and performance analysis based on queuing theory or Stochastic Petri Nets. The SPT profile is used as a basis for some other UML1.4 based profiles such as RT-UML, MAST-RT, embedded UML, HIDOORS UML profile, Graf-Ober profile etc., as shown in Fig.1. With the advent of the new UML2.x, started the development and use of some new UML profiles for real-time, among them with the highest prevalence is MARTE profile that aims to completely replace the SPT profile.

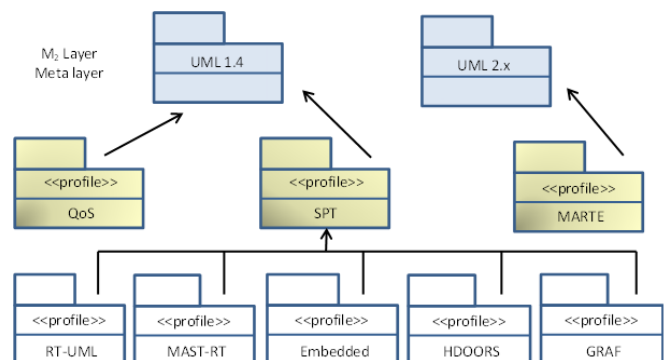


Fig.1: UML profiles for real time

2.1. Short overview of SysML

SysML is a general-purpose modeling language for system engineering that reuses a subset of the last UML2.x versions and provides additional extensions through stereotypes, diagram extensions and model library in order to model a wide range of system engineering problems as for example specifying requirements, structure, behaviour, allocations and constraints on system properties to support engineering analysis. The reusable subset of UML, known as UML4SysML includes Interactions, State machines, Use Cases and Profiles. In Fig.2 the set of SysML diagrams in respect to their modeling aspects is summarized. The system structure design is supported by four types of diagrams: Block Definition Diagram (BDD), Internal Block Diagram (IBD) reinforced by Parametric Diagram (ParD), and Packages Diagrams (PacD). The Behaviour Diagrams incorporate four diagrams too, namely: Activity Diagram (AD), Sequence Diagram (SD), State Machine Diagram (SMD), and Use Case Diagram (UCD). The Requirements Diagrams (RD), which can be presented in graphical, tabular or tree structure format, are used to specify different constructs for system requirements and to cover the relationships between them. In SysML two kinds of requirements are used – functional and performance, as they specify the capabilities or the conditions which must be performed or satisfied by the system.

Other modeling capabilities of SysML not shown in fig.2 are the cross-cutting constructs, such as allocations for connecting of different views, and Profiles & Model libraries allowing further customizing and extending of SysML to specific applications. SysML also includes extensions supporting the causal analysis, the verification and testing processes and the decision tree development.

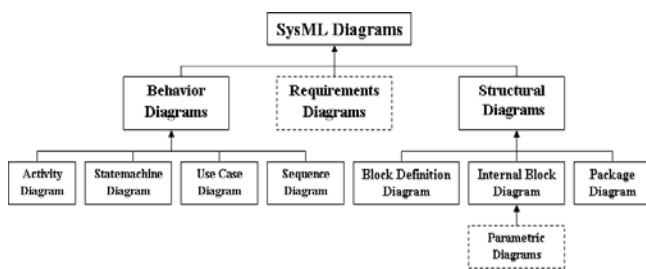


Fig.2. SysML Diagrams [5]

2.2. Short overview of MARTE profile

MARTE is a specification of OMG and is based on the SPT (Schedulability, Performance and Time) profile using the standard notations and semantics of UML [6, 9]. This UML profile is an independent framework, offering a compatible set of standard notations and semantics to design custom hardware and software applications. MARTE profile consists of three packages named “MARTE Foundation”, “MARTE Design Model” and “MARTE Analysis Model”, shown in fig.3.

“MARTE Foundation” package defines all basic foundational concepts required for design and analysis of real-time and embedded system. It provides model developers with constructs for modelling of non-functional properties (NFPs), time modelling, generic resource modelling (GRM), generic component model (GCM) and allocation modelling.

“MARTE Design Model” package addresses model-based design, starting from requirement capture to specification, design and implementation. It provides high level concepts for modelling both, quantitative and qualitative features of real-time systems/protocols. Further, it also provides means for detailed description of software and hardware resources used for execution of an application.

The package “MARTE Analysis Model” offers specific abstractions and relevant annotations that could be read by analysis tools. MARTE analysis is intended to provide trustworthy and accurate evaluations using formal quantitative analysis based on sound mathematical models. This package is sub-divided into three other

packages, namely “Generic Quantitative Analysis Modeling” (GQAM), “Schedulability Analysis Modeling” (SAM) and “Performance Analysis Modeling” (PAM).

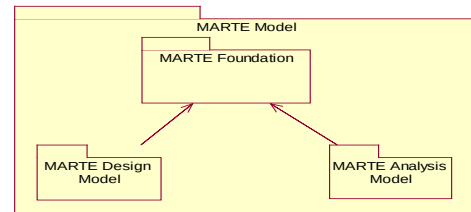


Fig.3: MARTE profile

By defining new stereotypes for elements in structural diagrams, activity diagrams and statechart diagrams UML MARTE profile ensures concepts for modelling and analysis of time characteristic and constraints such as clocks, time and delay. Additional stereotypes used to model the time are <<Clocks>> and its instances as: <<ClockType>>, <<ClockConstraint>>, <<TimedEvent>> and <<TimedProcessing>>. The properties of the types of clocks used, the timer functions as a resolution, maximum value and act are defined by stereotype <<ClockType>>. <<ClockConstraint>> stereotype is used to present the constraints of the clock by determining the dependence on the time structure in a time domain [10]. The specification of time constraints is presented declaratively using OCL language [11]. By stereotype <<TimedEvent>> a particular clock is specified. In cases of repeated event, the frequency and period of repetition are defined. Stereotype <<TimedProcessing>> expand meta-classes of elements representing behaviour, messages, and actions [12].

MARTE profile extensions are implemented by adding new model library stereotypes. Elements from library models can be used to model different levels of meta-models, profiles and applications.

Like SysML, the concept “allocation” is introduced. The allocation in MARTE profile means the choice of platform on which the developed application will be realized. The allocation covers both the spatial distributions and temporal aspects of time scheduling in order to map different executable algorithms to existing computing and communication resources and services. The time scheduling is necessary in cases in which multiple applications are distributed on the same platform, or when synchronization is required between the various elements.

3. Short description of suggested approach

In this section the suggested approach for development of process control system based on the combined use of UML profiles MARTE and SysML is proposed. This approach offers a software development life cycle model using combination of modified Harmony-SE methodology [13] and MADES approach [14].

The Harmony-SE based methodology uses service-request-driven modeling approach based on SysML structure diagrams using blocks as basic structural elements. The requirements analysis phase starts with the analysis of process inputs. Customer requirements are translated into a set of requirements that define what the system must do and how well it must perform. For the purposes of this first stage, the SysML requirements diagrams (RD) to create taxonomy of the captured requirements and System Use Cases (SUC) are used. The system functional analysis phase is presented through transformation of the identified functional requirements into coherent system functions. Use Case Diagram (UCD) presents the system functional phase. For each use case an analysis is performed. BDD and IBD are applied to present the composite system structure. Different Black-Box Activity (BB-AD) and Sequence diagrams (BB-SD) are used to capture the high level system functionality. The Architecture design is the third stage in the proposed methodology and includes the design of subsystem structures and behaviour based on White-Box (WB) variants of the

diagrams used in the previous stage (WB-UCD, WB-AD, WB-SD). Other important tasks are the ports and interfaces definitions and the description of subsystem state-based behaviour using the SysML statecharts (SCD).

The last stage is the Hardware/Software design specification and is closely connected with the system implementation. All methodology stages include verification and validation tasks, based on the model developed in the previous stages and the defined requirements [15].

The MADES approach aims to provide effective framework which combines the both profiles SysML and MARTE. The design phases include some abstract stages. The first stage of the methodology carries out system design at high level of abstraction. This stage includes a few steps for system requirements and use case scenarios definition. For the purposes of this stage the SysML requirements diagrams and use case diagram are used. The High Level Specification and Refined High Level Specification are the next two stages in which the composite and detailed structure of the application is modeled. In Refined High Level Specification stage the parallel between SysML and MARTE profiles is ensured. The relationship between these profiles is used in order to model the hardware and software aspects of the system. The hardware and software modeling are the next stages in MADES in which modeling is combined with MARTE Non Functional Properties and Timed Modeling package to express aspects such as throughput, temporal constraints, etc. The connection between high level and detailed high level specifications is carried out by Allocation diagram.

The proposed development process includes the following basic life cycle phases: Requirements Analysis; System functional analysis; Architecture design; Hardware/Software design specification based on a combined subset of MARTE and SysML. The first two stages - Requirements Analysis and System functional analysis are modeled by SysML profile using Requirements Diagrams, Use Case Diagram, BDD and IBD. The last stage Hardware/Software design specification can be separated in two substages namely Hardware design specification and Software design specification and use BDD and IBD with MARTE stereotypes.

4. Case Study

In this section, an application for water level feedback control system using PID controller as case study is presented. The physical system is shown in fig.4 and consists of a cylindrical tank, filled with water until specified level that is registered with a level sensor and is controlled by changing the input valve position. The core of control system is a feedback controller, which controls the level in the tank according to the PID principle. The discussed example is divided in two parts – modeling of physical system, named “Physical SubSystem” and modeling of control system called “ControlSubSystem”.

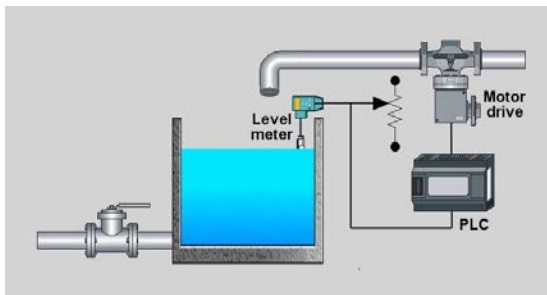


Fig.4: Physical system

For the development of case study, Modelio tool [16] is used. The Modelio is a development of Softeam group and provides an open source tool for development and maintain of MDA for UML through the profile technique. Modelio supports several UML

diagrams, as well as additional diagrams such as Business Process, Requirement, SysML or Enterprise Architecture diagrams.

The Requirements Analysis and System Functional Analysis are developed according Harmony-SE based methodology and the diagrams supporting these phases are Requirements diagram and Use Case Diagram which are described in details in [4].

In Fig.5 the BDD of “ControlSystem” is presented, which is composed by different types of equipments such as input valve, output valve, tank and sensor. For each part the appropriate MARTE stereotypes defining hardware and software resource are assigned. For “ControlSubSystem” part the `<<ComputingResource>>` is assigned, while “TankSubSystem” part by `<<ProcessingResource>>` is defined. The communication bus presented by part “BUS” is defined through stereotype `<<CommunicationMedia>>`.

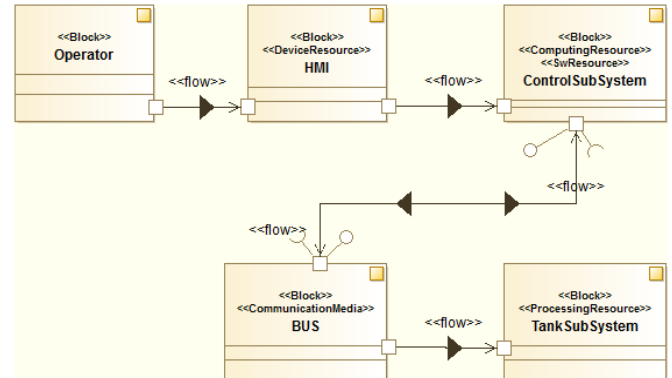


Fig.5: BDD for ControlSystem

The static composite static structure of the controller and namely “ControlSubSystem” and internal structure are modeled using respectively BDD (fig. 6) and IBD (fig.7). The components of “ControlSubSystem” are “Derivative”, “PID_Calc” and “Integral” modeled as blocks. The Internal Block Diagram contains three main parts: “PID_Calc”, “Derivative” and “Integral” with stereotypes `<<ComputingResource>>`, which are interconnected via ports.

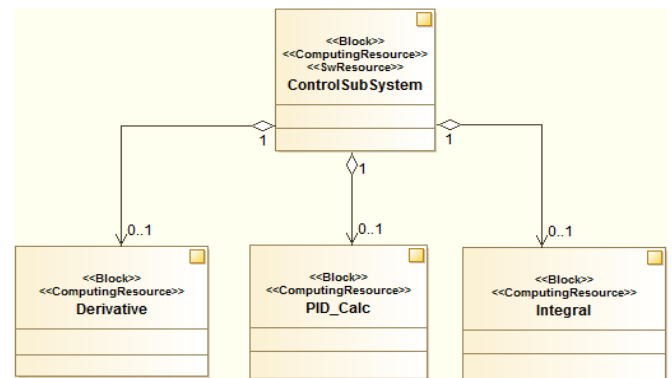


Fig.6: BDD for ControlSubSystem

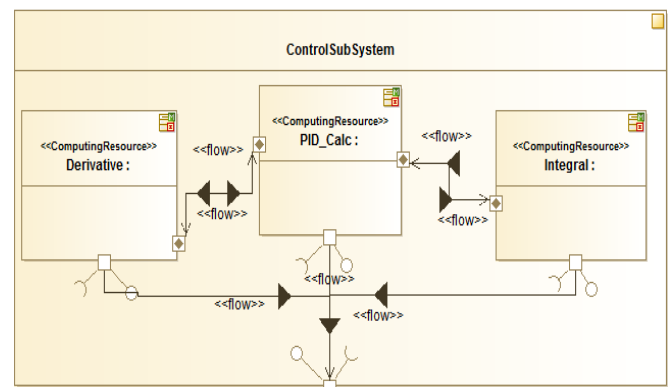


Fig.7: IBD for ControlSubSystem

The static composite structure of the “TankSubSystem” is presented with BDD (fig.8) and shows the physical structure in terms of installed equipment. The “TankSubSystem” is composed by different types of equipment’s such as input valve, output valve, tank and sensor. In Fig.9 the IBD of “TankSubSystem” is presented. For the blocks “Valve”, “Valve_in” and “Valve_out” the stereotype <<HwActuator>> is assigned, for sensors the stereotype is <<HwI_O>> and for tank <<HwProcessingResource>>. The connections between parts are modeled as ports type “flow”.

Once the hardware and software specifications have been created, the allocations between hardware/detailed hardware, software/detailed software and hardware/software have to be defined using allocation diagram. An allocation diagram representing the allocation between software and detailed software specification is shown in fig.10.

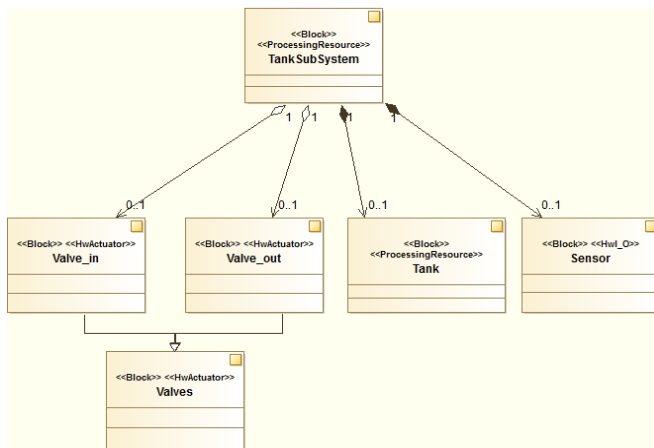


Fig.8: BDD for TankSubSystem

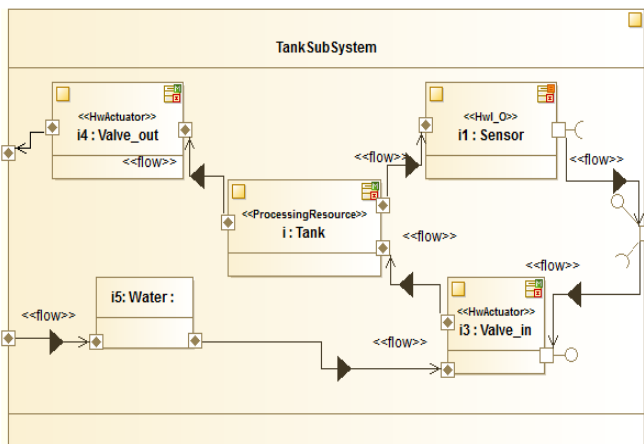


Fig.9: IBD for TankSubSystem

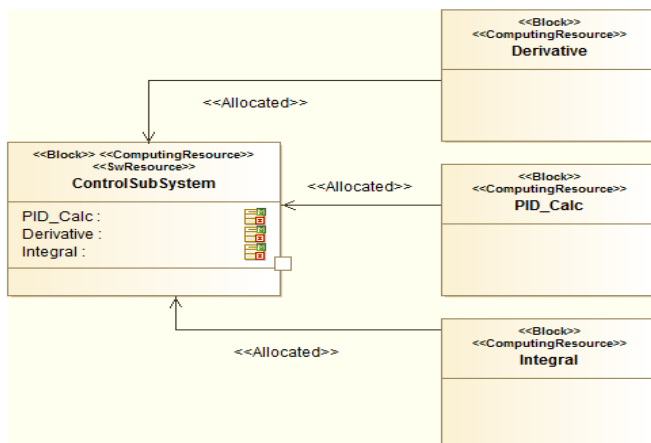


Fig.10: Allocation between software/detailed software components

5. Conclusions

The suggested methodology for development of software control system based on the combined use of the both UML profiles SysML and MARTE is suitable for development of open, interoperable, re-configurable, and distributed and process control systems. This methodology allows describing the whole live-cycle of the control system. One of the most essential features of this approach is that control engineers are able to model the closed loop control system and to apply the different type of analysis techniques in order to determine whether these models meet their performance and schedulability requirements, without requiring a deep understanding of the inner working of those techniques. Another main advantage of suggested approach is the possibility for analysis of the designed system and detailed design of the hardware and software platform of the modeled application provided by using of UML/MARTE profile.

6. References

1. Kent, S. (2002). Model Driven Engineering. In Proceedings of IFM2002, LNCS 2335, Springer.
2. OMG-MDA (2003). MDA Guide version 1.0.1. OMG document omg/2003-06-01, 2003
3. OMG-UML (2010). OMG. <http://www.omg.org/>.
4. Batchkova I., Antonova I. (2011), Improving the Software Development Life Cycle in Process Control using UML/SysML, Preprints of the 18th IFAC World Congress, pp. 14133-14138, August 28 - September 2, Milano, Italy.
5. OMG-SysML (2006). The OMG Systems Modeling Language, <http://omgsysml.org/index.htm>, May.
6. OMG MARTE (2007). Profile for Modeling and Analysis of Real-Time and Embedded (MARTE) systems, Beta 1, 2007.
7. Boggs W., Boggs M. (2002). Sybex Mastering UML with Rational Rose 2002, Sybex, 2002
8. Nierstrasz O. (1989). Object-Oriented Concepts, Databases and Applications, ed. W. Kim and F. Lochovsky, pp. 3-21, ACM Press and Addison- Wesley, 1989.
9. Mallet F., Simone R. (2008). MARTE: A Profile for RT/E Systems Modeling, Analysis —and Simulation?, 1st Int. Conf. on Simulation Tools and Techniques for communications, Networks and Systems SIMUTools'08, pp. 1-8.
10. Peraldi-Frati M., Sorel Y., From high-level modeling of time in MARTE to real time scheduling analysis, In MoDELS'08 W. on Model Based Architecting and Construction of Embedded Systems on ACESMB, pp. 129-143, Toulouse, France, September 2008
11. OMG, Object Constraint Language, version 2.0, May 2006, OMG document number: formal/06-05-01.
12. Andre C., Mallet F., Simone R., Modeling time(s). In Engels, G., Opdyke, B., Schmidt, D.C., Weil, F., eds.: MoDELS. Volume 4735 of Lecture Notes in Computer Science., Springer (2007) 559_573
13. Estefan, A. J. (2008). Survey of Model-Based Systems Engineering (MBSE) Methodologies, INCOSE MBSE Focus Group.
14. Quadri I., Sadovykh A., Indrusiak L., MADES: A SysML/MARTE high level methodology for real-time and embedded systems, 2012 7th International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), ISBN (Print) 978-1-4673-2572-1
15. Hoffman H.- P., Harmony-SE/SysML Deskbook Model-based Systems Engineering with Rhapsody, Rev.1.51, Telelogic/I-Logix whitepaper, Telelogic AB, May 24, 2006.
16. Modelio tool, <http://www.modeliosoft.com/>