Model-Driven Engineering of Embedded Real-Time Systems

Federico Ciccozzi

Mälardalen University, Mälardalen Real-Time Research Center
federico.ciccozzi@mdh.se

1 Introduction

1.1 Research Topic

Model-Based Engineering (MDE) aims to improve productivity by increasing the return which companies can derive from previous software development effort [1]. The Composition with guarantees for High integrity Embedded Software componentS assembly (CHESS) project focuses on the area of design and analysis of extra-functional properties of diverse systems in MDE; the aim is building modeling languages for the specification of such properties based on a common platform-independent component model to be developed and which could be used in several different domains (space, telecom, railways, automotive). Development of tools for extra-functional properties evaluation and provision support for evaluation of software architectures and traceability of non-functional requirements have to be provided. A noteworthy goal of the project is to provide adaptation of component infrastructures for integration of real-time and dependable systems; it should be possible to integrate the specific design patterns associated to real-time and dependable systems to the software infrastructures, modeling tools and automatic software generators. Moreover validation of the entire approach through innovative dynamic component-based case studies has to be fulfilled. The MDH side contribution to CHESS will mainly involve the Telecom domain. Requirements for the domain have to be formalized in a uniform way and analysis methods have to be specified. Execution platform and rules for transforming from CHESS to platform-specific models have to be specified and knowledge in modeling languages and tools, domain-specific modeling, model transformations, transformation tools, model weaving will be required as well as remarkable research in both component-based and embedded real-time systems. Several code generators from the CHESS modeling language will be needed as well as transformers from analysis tools’ output format and the CHESS modeling language for round-trip engineering. Reuse and integration of existing tools will be also relevant. The project results have to be verified and validated through corresponding use cases.

1.2 Research Method

In order to reach the given objectives, several research activities have to be carried out. The component model to be developed has to address several different domains (space, telecom, railways, automotive), therefore a deep investigation of every domain’s needs
Research Planning Course 2009/2010

is a central issue to carry out the requirements and specification phase. Decisions must be taken regarding the development of the component model: reuse vs. development from scratch. The available component models have to be explored and compared to the project objectives in order to take such a decision. The component model will be followed to build platform-independent models of the system under development. Through a set of transformations and ad-hoc tools that will be developed or adopted, the system will allow the generation of analyzable code and furthermore analysis results. Evolution, co-evolution and synchronization of models must be handled in order to maintain traceability and consistency among the transformed and enriched models. Knowledge has to be accrued in the areas of model transformations, evolution and co-evolution, as well as model synchronization. The system must eventually provide analysis tools and methodologies for verification and validation; existing model-based analysis approaches and methodologies will be explored.

2 Field of Research

2.1 Model Driven Engineering

Model Driven Engineering (MDE) aims at facilitating the system development by creating, maintaining and manipulating models. A possible scenario in an MDE process is the OMG’s Model Driven Architecture (MDA), which is based on a distinction between platform independent models (PIM) and platform description models (PDM) that are realized through a set of middleware and programming languages into platform specific models (PSM). The MDE term was first introduced by Kent [1] and includes all modeling tasks needed for the entire software development process. A system is developed by refining models starting from higher and moving to lower levels of abstraction until code is generated; refinement is implemented by transformations over models. Maintaining traceability among models over transformations, consistency and platform independence is not trivial. In fact, the more transformations are applied to models, and higher is the possibility to lose initial information. The need to address several different platforms makes the problem even less trivial. In MoDELS’08 there has been a wide discussion over the most noteworthy challenges in MDE; many of them are highly relevant in the CHESS project and my research area in general and can be summarized as follows [2]:

- **Quality of developed models**: model quality metrics have to defined and guidelines to measure, predict, improve and predict quality have to be agreed. It is also important to assure balanced trade-offs between different and sometimes conflicting quality aspects.

- **Model at run-time**: In order to have a higher applicability in industry, the use of models has to be extended also to the phases beyond the design and implementation. Dynamic behavior has to be represented in some way and the derived run-time models be executed. A relationship between static and dynamic models has to be caught in order to maintain traceability and consistency. In this area a core aspect is the research in code generation techniques which, if well accomplished, could lead to a considerable reduction in the coding effort.
– **Requirements modelling**: the first phases of the software development process concern the requirements specification. In order to have a more complete model-based approach, requirements have also to be modelled and traceability between written informal specification and formal models has to be maintained.

– **Standards**: a set of standards has to be selected in order to facilitate interoperability between different tools used in the same development process. A way of sharing models, model transformations and case studies among them would be very useful.

– **Modelling languages**: development of models involves well-defined modelling languages. Languages, methods and tools have to be selected to fit the needs of the meta-model that has to be created as basis for the further modelling phases. Possibilities for dealing with multi-models and multi-modelling in the same process have to be explored.

– **Domain-specific modelling**: As stated before, the model-based development process consists in a series of model transformations. MDA sees this process as a transit from PIM to PSM models. In order to achieve it, domain-specific modelling languages must be designed and models specified in different languages should be related. Reusability across different domain-specific modelling languages should be increased and code generation facilitated. A certain balance of the trade-off between general-purpose modelling languages and tools and domain-specific ones could be necessary, depending on the specific environment.

– **Model verification and validation**: the latter phases of software development process regard verification and validation. A way to verify, validate and test models and generated code has to be found and automation of test cases generation is a core issue. Validation and verification should be addressed also to a multi-model development.

– **Industrial adoption**: the industrial adoption is not always cost effective. In cases of companies with a large legacy code base written in old technologies, the migration to MDE could imply an inconvenient effort in terms of cost and time.

**Model transformation**  Model-Driven Engineering concerns not only horizontal usage of models, but rather a horizontal/vertical combination is preferred. These multiple interrelated models have to be handled paying particular attention to their overall consistency through model synchronization. This is performed through a set of automated processes called model transformations. A model transformation is a particular process which takes, as input, a set of source models and, following specified transformation rules, derives a set of target models as output. The model transformation approaches can be divided into the following categories and subcategories [3]:

– **Model-To-Code**
  - Visitor-Based: it is the simplest approach. It writes code to a text stream by a visitor mechanism which explores the internal representation of the model to be transformed.
  - Template-Based: the approach is based on a template which uses target metadata to access information from the source and to select and interatively expand code.

– **Model-To-Model**
Direct-Manipulation: these approaches perform model transformation using an internal model representation and some provided APIs plus transformation rules developed usually from scratch by the developer.

Relational: approaches based on different mathematical relations. Query/Model/View (QVT) is a standard for relation model transformations introduced by OMG.

Graph-Transformation-Based: these approaches perform transformations by using particular graphs specifically designed to represent UML-like models.

Structure-Driven: they are characterized by a two-phases process that first creates the hierarchical structure of the target model and eventually sets its attributes and references.

Different approaches can be combined in hybrid approaches in order to fit particular needs.

Model consistency and co-evolution A MDE approach can imply several different developers, modelling languages and tools that model the same system. This wide variety of actors together with model transformations within a single development branch, can lead to inconsistencies both in and between models. The possibility to develop domain-specific modelling languages (DSMLs) with high level of adaption and extension capabilities could help in avoiding some sorts of inconsistencies. A solution would be a rapid prototyping in the DSMLs building in order to have continuous and incremental feedback. This evolution leads to the question of what happens to previous developed models after an evolution of the DSML and how the eventual inconsistencies between different versions may be solved. A need of built-in support for handling of these inconsistencies arises in both model-driven development environments, from models to generated code (model-code synchronization and round-trip engineering), and versioning tools.

3 The Community

In this section a list of the most relevant papers, research groups and conferences on the research area will be given.

3.1 Seminal Papers

MDE/MDA In 2002 S. Kent published in Lecture Notes in Computer Science (LNCS) a paper [1] on a framework for model driven engineering, which could be used as reference for activities in this area. Starting from the MDA definition by the Object Management Group (OMG), it goes through the main issues in a model driven engineering approach from how to locate models in the modelling space to how to achieve mappings between models. Importance of several tools is stated and the need of defining families of languages and transformations is arised. J. Bézivin published in 2005 a paper [4] which proposes a vision of MDE development based on the previous object technology approach. From the message ‘Everything is an object’ typical of the object technology, a new message driven by MDE is proposed: ‘Everything is a model’. The two core
object technology relations, inheritance and instantiation, are so replaced by representation and conformance. The discussed issues are very useful to understand the MDE area in general and the MDA approach in particular.

**Model Transformations** MDA aims at automatically generating platform-specific models from platform-independent models. Well-established standards for the modelling phases have been reached, but the specification of transformations between those models is not yet mature. In 2003, K. Czarnecki and S. Helsen proposed in [3] a taxonomy for the classification of both existing and proposed model transformation approaches, which are divided into a small set of major categories based on the authors’ analysis. T. Mens, K. Czarnecki and P. Van Gorp proposed in [5] a summary of the results of the discussion of the Dagsthul Seminar on Language Engineering for Model-Driven Software Development. In this paper essential characteristics of model transformations are discussed together with their supporting languages and tools. A comparison between different existing model transformation approaches is provided with a discussion on commonalities and variabilities among them. The paper aims at helping the developer in choosing the model transformation approach which best matches his needs.

### 3.2 Other Relevant Papers

C. Atkinson and T. Künhe published a paper in 2002 [6] in which they discuss the role of meta-modeling in Model-Driven Development and suggest a set of requirements that a supporting infrastructure should satisfy. Regarding the challenges in MDE, R. Van De Straeten, T. Mens and S. Van Baelen wrote a paper in 2009 [2] which, based on the progress in state-of-the-art and state-of-the-practice in MDE, aims at providing a set of MDE research and practice problems that are still unsolved. They summarize different challenges identified by major experts in the field.

### 3.3 Leading Research Groups

**Prof. Jean Bézivin, Faculty of Sciences, University of Nantes, France** The group led by Prof. Bézivin focus its research on several aspects of model-driven engineering and model transformation.

**Prof. Jean-Marc Jézéquel, IRISA, University of Rennes, France** Another French MDE research group, led by Prof. Jézéquel, focuses on the domain of component based reactive and large scale distributed systems with quality of service constraints. They developed a meta-modeling environment called Kernel Meta-modeling (KerMeta) [7].

**Prof. Krzysztof Czarnecki, Department of Electrical and Computer Engineering, University of Waterloo, Canada** Prof. Czarnecki and his research group are highly involved in the model transformation area, and have developed tools for feature modeling and round-trip support for frameworks-specific modeling languages.
Daniel Varró, Department of Measurement and Information Systems, University of Technology and Economics of Budapest, Hungary  The Hungarian research team from University of Budapest focuses on model transformations, particularly on graph transformations. They developed a framework called VIsual Automated mode TRAnsformations (VIATRA) [8] which provides a general-purpose support for the entire lifecycle of engineering model transformations from specification to maintenance.

Prof. Alfonso Pierantonio, Department of Computer Science, University of L’Aquila, Italy  The Italian team led by Prof. Pierantonio focuses on several specific aspects of MDE such as meta-model evolution, model co-evolution, model synchronization.

3.4 Conferences

– **ICSE (International Conference on Software Engineering):** since it can be considered the premier worldwide software engineering conference, ICSE hosts the most recent trends and innovation, including MDE topics [9].

– **ASE (International Conference on Automate Software Engineering):** is one of the world’s premier software engineering conferences and usually hosts MDE topics since considered an innovative trend [10].

– **MoDELS (International Conference on Model Driven Engineering Languages and Systems):** formerly called UML conference, it is a series of conferences entirely devoted to MDE topics, from languages to systems [11].

– **ECMDA (European Conference on Model-Driven Architecture):** European conference on the advancing of the MDA technology in industry. The main focus is on engaging key figures of research and industry to achieve more reliable software by the industry on the basis of research innovations [12].

– **ICMT (International Conference on Model Transformation):** international conference focusing on challenges for existing model transformation technology as well as future problems and possible solutions [13].

– **TOOLS (International Conference Objects, Models, Components, Patterns):** it focuses on the most modern approaches in software development with a special focus on Object-Oriented and components [14].

Workshops

– **TWOMDE (Workshop on Transformation and Weaving Ontologies in Model Driven Engineering):** it aims at providing discussions about the application of different aspects of transformation and weaving ontologies to enhance MDE/MDA [15].

– **MoDeVVA (Model Driven Engineering, Verification and Validation):** it focuses on model design, verification and validation and aims at sharing new V&V approaches in the MDE context [16].

Journals
– *Software and Systems Modeling (SoSyM)*: international journal on theoretical and practical issues about the development and application of software and system modeling languages and techniques. It emphasizes theoretical foundations of modeling languages and techniques and analyses real-world modeling experiences [17].

– *Transactions on Software Engineering and Methodology (TOSEM)*: it collects the best papers on the challenge of designing and building of large and complex software systems in every aspect, from specification to maintainance. The focus is not only limited to the methodology but also to tools, languages, algorithms and data structures [18].

– *Transactions on Software Engineering (TSE)*: international journal focusing on well-defined theoretical results and empirical studies in the areas of building, analysis and management of software [19].

4 Relevance for the PhD research

My research has MDE as core subject with concentration in model transformation and co-evolution. More specifically, the research will embrace model-based development of embedded systems in the telecom sector with a special focus on modeling system architecture and behavior in order to enable property prediction from models. Techniques that fuse model-based and component-based software-engineering methodologies will be also of special interest. All these aspects of MDE will be explored in the CHESS project. The research work carried out at Mälardalens Högskola will be mainly devoted to the Telecom domain in collaboration with Ericsson and ENEA, while other domains will be taken care by several other industrial and academic partners.

References


