MŰEGYETEM 1782
Budapest University of Technology and Economics
Department of Measurement and Information Systems

Metamodel-based model generation and validation techniques with applications

PhD thesis booklet

Zoltán Szatmári
MSc in technical informatics

Advisor:
István Majzik, PhD (BME)

Budapest, 2016

1 Objectives

Models always played an important role in system and software development, e.g., they were used to capture database-schemas, domain knowledge and requirements. Nowadays they become more important and act as a core concept in many areas of system engineering. Model-driven development is an emerging approach, especially in safety-critical system development, where models are used in several phases of the development process, e.g. during the requirement analysis, system design, implementation and also testing.

Starting with the Model Driven Architecture (MDA) [Obj01] issued by the Object Management Group (OMG), model-based technologies became essential technologies of software engineering. General purpose modeling languages like UML [OMG11] and SysML [Obj12] are used in different life cycle phases, although they are characterized by the lack of precise semantics. The domain-specific modeling languages are increasingly preferred, because these provide better focus on the target domain and also provide better tool-support and flexible integration opportunities. One of the main achievements of this approach is that it is more simple to capture the domain knowledge and the domain expert can be more easily involved in the system development.

1.1 Challenges in new application domains

This dissertation focuses on new and emerging application domains, which present several new challenges for the language engineering, as well as for the model manipulation, generation, validation and query activities.

The presented application domains belong to different industrial applications. Their challenges can be traced back to a common background, namely, domain specific modeling (DSM) that raises similar questions in these domains: (1) how can domain experts with limited IT and modeling skills capture the domain knowledge in order to support the solution of domain-related problems, and (2) how can domain-specific models be generated and validated in the development process.

1.1.1 Context-aware autonomous systems

Nowadays context-aware mobile and autonomous systems (AS) are wide-spread in different areas, like transportation systems (e.g., laser guided forklifts), military systems (e.g., unmanned aerial vehicles) or in the household (e.g., autonomous vacuum cleaners). This application domain was addressed in the R3-COP research project [R3C11] that aimed to support the development of robust and safe, autonomous and co-operative robotic systems.

A formal definition of autonomous systems (agents) [FG96] can be given in the following way: *An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.* So an autonomous system is a system that makes and executes a decision to achieve a goal without full, direct human control [Con+06]. The reference architecture is shown in Figure 1. This set up is very similar to the arrangement that authors use in [RN03], when defining the connection between an agent and its context.

The agent program utilizes an *internal representation of the context*, that stores the knowledge of the agent about its environment. This representation should describe all the context objects and events that are relevant to the behaviour (control algorithms) of the agent.

Testing is an essential step of all software development processes. Testing such autonomous agents is a challenging and complex task, because of the context-aware behaviour and the large number of potential environment configurations (context).

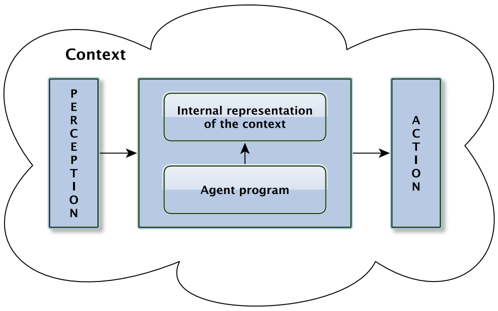

Figure 1: Architecture of an autonomous agent.

Writing test cases is labor-intensive and time consuming, thus facilitation or automation of the test data generation process is desired. The main challenge in test data generation is to avoid ad-hoc testing and to support systematic test case generation using measurable coverage metrics.

In order to support the testing of autonomous agents, contexts should be generated in which the mission of the agent can be executed and evaluated. These contexts can be represented as models and can be created using model generation techniques based on domain-specific information: namely, the metamodel that describes the elements of a context and different well-formedness and semantic constraints that add restrictions to the models based on the domain knowledge.

The test contexts include the static configuration of the environment and may also represent dynamic changes, cooperating partners and humans, as well as explicit messages and commands from them. In such generated test contexts, the context-aware behaviour of an AS can be analyzed against functional, safety and robustness requirements.

In this testing approach the black-box testing of autonomous agents is supported, where the environment (and the perception of the AS) can be simulated and the test verdict can be constructed based on the test traces captured in the simulation output.

The challenges in this approach are (1) how to support the domain expert in capturing the necessary domain knowledge in a formal model and (2) how to provide efficient context model generation methods to satisfy various test data generation strategies and coverage criteria.

1.1.2 Development processes of safety-critical software

Our everyday life depends on software to a considerable extent, this way the reduction of the risks of design and implementation failures is of utmost importance.

To reduce the risks of residual software design failures, the software development processes are more and more subject to regulations fixed in (domain-specific) standards that define criteria for the selection of proper techniques and measures. Accordingly, if software is deployed in a critical environment then an independent assessment is needed to certify that its development process is compliant to the criteria stated in the related standard.

Applying formal modeling techniques could improve the process assessment. This requires the representation of the domain-specific standard, where a formal model can be constructed by capturing the concepts, their relations and any additional requirements based on the (mainly textual) standard.

The challenge in this process certification lies in that the domain-specific standard-based requirements should be formalized using a domain-specific language and a *formal validation method* should be constructed based on the collected knowledge. This application domain was addressed in the CECRIS

EU FP7 project [CEC13] that aims to provide methodologies to improve the development, verification, validation and certification of critical systems.

1.2 Summarizing the new challenges

In summary, I identified the following *open research questions* that motivated my work and by identifying the underlying (more generic) challenges lead to the scientific results presented in the dissertation.

Challenge 1: *High-level definition and validation of domain-specific languages* How domain-specific languages can be defined in a high-level and user-friendly way, and how the language design can be validated for consistency? To address this challenge, the use of ontologies for capturing the domain knowledge and ontology related tools for the early validation of the language design is a promising approach.

Challenge 2: *Context model generation* for testing autonomous systems. How can the testing of autonomous agents, operating in a dynamic, frequently changing environment, be supported by the generation of test context models? The generation of models shall be based on the context (meta)model, domain constraints, and model coverage metrics representing the testing strategy.

Challenge 3: *Model validation* for standard-based assessment of development processes. How can the different requirements specified in various standards for safety-critical development processes be formalized, and how can these formalized requirements be checked to assess the standard compliance of development processes? When the development process is represented as a process model (that is commonly used in case of software development processes) then the formalized requirements can be checked by model validation in the form of model queries, making it possible to identify compliance problems and potential optimizations in the process model.

The generic challenges (i.e., that are independent of the mentioned applications) behind these research questions are related to domain-specific language engineering and domain-specific modeling. Namely, the following generic challenges can be identified:

1. Ontology based design and validation of a domain specific language and the related constraints,
2. Generation of instance models on the basis of metamodels, model metrics and constraints,
3. Validation of models on the basis of requirements represented as model queries.

The technologies proposed in the dissertation offer solutions to the specific research questions of these application domains as well as to similar questions in other domains.

The research questions emerged in the discussed application domains and the related generic challenges addressed in the dissertation are summarized in Table 1.

1.3 State of the art

The above presented research questions have a common background: model-based techniques are required to capture the domain-specific knowledge and – based on the constructed models – model validation and model generation techniques are needed.

There are several approaches in the model-driven development community to use ontologies to design domain-specific languages and to apply model-based testing, however, they need to be adapted or extended to suit the needs of these new application domains. In the following I summarize the state of the art in this area to provide the context of my work.

Application domain	Specific research question	Generic challenge
Capturing domain knowledge	Designing domain specific languages with early detection of flaws by validation of the language	Deriving domain specific languages from ontology specifications and application of ontology reasoners for validation
Testing context aware behavior of autonomous systems	Constructing test context models to check the safety and robustness of the behaviour of autonomous systems	Model generation on the basis of metamodel, domain constraints, and model metrics
Assessing the development processes of safety-critical software	Checking models of development processes on the basis of requirements derived from standards	Model validation based on model queries and concept matching

Table 1: Application domains, research questions and the related generic challenges

1.3.1 Domain-specific language engineering

DSLs were successfully applied in many areas, because of their flexibility and tool support. Domain-specific language engineering is the process, when a new DSL is constructed optionally reusing an existing one or starting from scratch.

The specification of complex DSLs necessitates a combination of different techniques. The abstract syntax of the DSL is usually captured by a metamodel, which can be augmented with well-formedness constraints, which capture additional restrictions that need to be respected by any well-formed instance model. Such constraints can be defined by model queries, graph patterns (special type of model queries) [Ber+11] or as invariants expressed in the Object Constraint Language (OCL). Domain specific requirements are usually captured using constraints and the requirement can be checked on an instance model by executing a model validation.

Capturing the requirements and domain concepts is one of the most important phase of model-driven development and DSL engineering. Based on the modeling language and requirement description, analysis, code generation and also model-based test data generation can be performed. Constructing a good modeling language is becoming less and less difficult, since domain-specific language engineering appears part of the industrial routine, but it still requires special IT technology and modeling skills. One of the addressed challenges of this dissertation is to support language engineering with a high-level approach, using ontologies and related tools for constructing and validating a domain-specific language.

On a lower, technology level, the Eclipse Modeling Framework (EMF) [EMF14] became a leading DSL technology and a de facto standard in the model-driven community. EMF provides a class diagram like structural description language for constructing a metamodel with sophisticated tool support for editing and model manipulation. Similarly, instance models can be created and manipulated with easy to use graphical editors (that are built on GMF, GEF or Graphiti) or textual editors (with Imp, Xtext, EMFText [EMF14]). Constraints can be captured using OCL or graph pattern language, which can be continuously evaluated during model manipulation, and errors can be displayed by marking inconsistent elements.

1.3.2 Ontology-based domain modeling

Ontologies using popular languages like OWL2 [OWL09] or SWRL [Gli+09] are able to capture domain knowledge in a very early phase of the design in a precise way. Ontologies provide a natural formalism for sketching the concepts of a domain or system and capturing additional constraints on a high level of abstraction. They can be written in several concrete textual syntaxes, including the machine processable RDF/XML (close to metamodel functional syntax) and the human-friendly Manchester syntax. Even controlled natural language representations are developed like the Attempto Controlled English, or the FluentEditor for OWL, which can help in language description or knowledge documentation. Ideas can be formalized incrementally as they come thanks to the open world assumption of ontologies. Distributed design is also supported, as unification and consistency check can be performed automatically with the help of reasoners.

I believe, that the ontology-based and metamodel-based domain-specific modeling approaches are both useful, but provide advantages in different phases of the language engineering. Ontologies provide high-level means for capturing the concepts and related constraints of systems and domains with precise semantics and automated meta-level reasoning techniques to identify specification flaws early in the design even if certain parts are underspecified. Metamodel based domain-specific modeling environments effectively support domain engineers in designing the system by providing efficient means for instance-level validation of well-formedness constraints.

Ontology reasoners are optimized for concept-level (i.e., meta-level) validation (like Pellet [Cla14] or RacerPro [Rac12]): they can detect inconsistent specifications and classifications in an early phase of design based upon the precise semantic foundations of ontologies. In contrary, when using metamodel based domain-specific techniques with OCL constraints, there is no support for detecting the inconsistencies in the defined metamodel and OCL statements and thus it is possible to create a language that is "unsatisfiable", i.e., no valid model can be constructed.

Metamodel based domain-specific language engineering frameworks and tools are tailored to the needs of domain engineers, thus increasing their productivity by offering functionally rich programming environments. DSM tools can be efficiently used by domain engineers for the detailed design of the system and for model manipulation. Efficient instance-level validators (like Eclipse OCL or EMF-IncQuery [Ber+10]) can be applied that can quickly detect violations of design rules or constraints on the instance-level. These DSM tools also support the development of domain-specific applications, like model-based test data generators.

Discovering and exploiting the synergies between ontologies (of semantic web engineering) and metamodels of domain-specific language engineering has become a hot research topic in recent years [Hil11; WSS09; ROD10]. There are approaches [Wal+10; WSR10; WSS09] that deal with domain specification based on description logic, other approaches [SSW09; Tao+10] present instance model validation support and another ones propose a mapping between the OWL ontology and EMF-based representation, but only for the metamodel. However, there is no support (1) for mapping well-formedness rules, that are not expressible by metamodels, (2) for mapping and evaluating SWRL rules and (3) for systematic benchmarking that covers the instance model validation implementations.

2 Research method and new results

The starting point of my work was the question: how to exploit the advantages of ontologies in system and software development?

In order to provide support for language engineering, I analyzed the application conditions of ontology-based models and the classical domain-specific languages used in the model-based development processes. I identified the key advantages / disadvantages and proposed a mapping from the ontology-based languages to domain-specific languages.

Afterwards, I focused on the construction of metamodels and on model manipulation, generation, validation and query techniques.

Finally, I adapted the ontology- and model-based technologies to the presented new challenges. As an application of these modeling technologies a new context modeling and test data generation approach for autonomous agents and a new process assessment method were developed.

Figure 2 gives an overview on how my research results are related to each other. The first thesis about domain specific language engineering deals with the common theoretical background, and the related methods and techniques. The other two theses address the problems appeared in the two discussed application domains (process model assessment and test data generation) through the solution of the related generic challenges (namely, model validation and model generation).

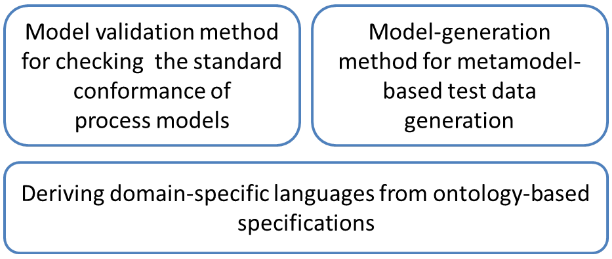
Model validation method for checking the standard conformance of process models
Model-generation method for metamodel-based test data generation
Deriving domain-specific languages from ontology-based specifications

Figure 2: Overview of the research results

The structure of each thesis is depicted in Figure 3. First I present the motivation, then I provide the methodology to address the identified challenges. Based on the provided methodology I propose specific technical solutions and finally I show the application of my results in a case study.

Motivation
Methodology
Technology
Application

Figure 3: Overview of the structure of each thesis

The rest of the section summarizes the new scientific results of the dissertation.

2.1 Deriving domain-specific languages from ontology-based specifications

I believe that the combined use of ontologies (where domain knowledge is captured in textual ontology languages like OWL2/SWRL) and DSM techniques will drive the development of DSM frameworks (i.e., domain specific-languages and the supporting tools for model manipulation, generation and validation). This thesis focuses on the domain specific language engineering and presents an approach how DSLs can be designed using ontologies.

Motivation The presented challenges of the application domains have a common background: domain specific languages are required to capture the domain specific knowledge, and these domain-specific languages and the constructed models need to be validated.

Accordingly, the overall goal of this research was to develop a domain-specific modeling framework, with the integration of OWL/SWRL and EMF based tools to combine the advantages of the two approaches. More specifically, to show that (1) domain specific language engineering can be efficiently supported by semantic web technologies, and (2) certain instance-level validation tasks can be executed on the domain-specific model by incremental model queries, especially when the underlying domain knowledge is changing or evolving.

Methodology The main contribution of this thesis is a method for deriving EMF based DSM languages from ontologies that describe the domain. First, I analyzed the commonalities and differences of the two modeling approaches and afterwards I presented the mapping of ontology based modeling structures to EMF-based domain-specific modeling artifacts by extending an existing ontology to eCore [EMF14] mapping [ROD10].

Formulating textual domain requirement specifications in ontology is described in related papers [DHK07] [DLS05]. This model capturing process requires fewer modeling and IT skills from the domain expert, since the domain knowledge (concepts and requirements) is captured using ontologies instead of strict metamodeling technologies. Based on the ontology-based representation the structures can be checked for consistency and then mapped to EMF-based domain-specific modeling artifacts. This allows the domain engineer to formulate and efficiently check consistency of requirements in an early design phase.

Technology On the technology level OWL2 and SWRL descriptions are used to capture high-level knowledge of a domain and Eclipse-based tools (e.g., EMF-IncQuery) are used for the DSL language development. The OWL2 and its extension, the SWRL language are used together to capture domain knowledge, while EMF-based technologies have better tool support for model manipulation. The proposed mapping is implemented as a transformation in the VIATRA2 framework [VB07].

Since the expressive power of these source and target languages of the transformation are not equal, the mapping consists of three steps. First, a part of the OWL2 ontology is mapped into metamodels in the industry-standard EMF platform, which provides the basic metamodel structure. Afterwards the remaining, complex OWL2 axioms are mapped into a textual graph pattern language (IQPL the IncQuery Pattern Language [Ber+11]) efficiently evaluated by the EMF-IncQuery incremental model query framework. Finally, the additional SWRL rules are also mapped into graph patterns.

Graph patterns as the target constraint description formalism was chosen (instead of OCL) since both SWRL and IQPL are graph pattern based languages, thus there was a smaller semantic gap between the two languages.

The conceptual and technological overview of the entire transformation is provided in Figure 4.

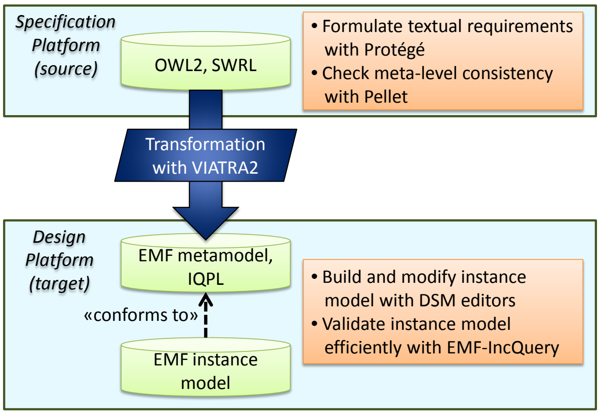
Specification Platform (source)
OWL2, SWRL
• Formulate textual requirements with Protégé
• Check meta-level consistency with Pellet
Transformation with VIATRA2
Design Platform (target)
EMF metamodel, IQPL
«conforms to»
EMF instance model
• Build and modify instance model with DSM editors
• Validate instance model efficiently with EMF-IncQuery

Figure 4: The ontology to domain-specific platform transformation process

The solution distinguishes the specification and the design platform.

- As the specification platform, it is required that domain engineers capture domain-specific knowledge and requirements using the SWRL extended OWL2 ontology language. In this phase ontology can be edited with Protégé [Bio14] and reasoners (like Pellet [Sir+07]) can be used for consistency checking.

- To form the design platform, the proposed model transformation automatically derives the EMF metamodel for the basic structures. From complex constraints which cannot be expressed directly in the EMF metamodel IQPL graph patterns [Ber+11] are generated.

Application I present an adaptation of a model validation benchmark to highlight and measure the main differences in model validation between the ontology-based tools and EMF metamodel based domain-specific modeling technologies. The benchmark helps the domain engineer to select an efficient representation and tooling for the validation of domain-specific models.

Benchmarks related to domain-specific modeling tools were already presented in [VSV05], where the performance of different model transformation tools were analyzed. Our research group focused on the incremental pattern matching approaches and a benchmark for different incremental and non incremental pattern matching algorithms were presented in [Ber+08]. Afterwards an incremental model query benchmark was proposed in [Ber+10] that implements the on-the-fly validation use-case for domain-specific models. Based on the typical usage scenarios of common DSLs, the benchmark addresses the on-the-fly model validation by checking well-formedness constraints during user-performed editing steps: at first, the entire model is validated, and after then each model manipulation step (e.g., deletion of a reference) is followed by an immediate re-validation.

Since I proposed a mathematically precise mapping between the ontology-based and EMF-based language definition formalisms, it was also possible to perform a comparison using benchmarks on the different model validation implementations. Based on the previous incremental model query benchmarking approach I extended the benchmark framework with the possibility to compare model validation technologies on different model representations. Additionally, I implemented the support for comparing the model-validation performance of several ontology-based (ABox reasoners) and EMF-based tools. Moreover, I introduced a new synthetic domain (called “railway domain”) with the related workload profile (queries and modifications), that supports the on-the-fly validation use-case.

Based on this extension of the domain-specific benchmarking framework the efficiency of available validation tools for various modeling formalisms can be examined using model queries with parametrized complexity.

Thesis 1 I defined the mapping from ontology based specification of domain concepts and constraints to domain specific metamodel and the corresponding graph-pattern based constraint description language. Using this mapping the design of domain-specific languages is effectively supported by the high-level ontology based representation of the domain knowledge.

1. *Mapping requirements formalized using OWL and SWRL to graph patterns*

   As an extension of an existing ontology to metamodel mapping, I defined a mapping from complex OWL-based requirements that cannot be mapped to the metamodel formalism (like class definitions given using complex statements) to graph patterns. Additionally, I defined a mapping from SWRL-based rules (that extend the ontology models) to graph patterns. The constructed graph patterns can be evaluated on instance models.

2. *Adaptation of the parametrized domain-specific benchmarking framework*

   I developed a new synthetic benchmark target (domain metamodel and domain constraints) and adapted the implementation of the domain-specific model validation benchmark to support this new benchmark target. I extended the benchmark framework by parameterized model queries which enable to compare the scalability of model query technologies with increasing query complexity. I implemented the queries for comparing ontology-based tools.

The work underlying Thesis 1 benefited from joint research with Benedek Izsó from our research group. My research results are (1) the definition of the mapping from OWL and SWRL description to the metamodel and GP formalism, (2) the definition of the new benchmark target and (3) the adaptation of the benchmark framework to the ontology-based tools.

Benedek Izsó's results are related to the proposed benchmark extension: he implemented the measurement framework, integrated the graph database implementations and he executed and analyzed the measurement results.

Gábor Szárnyas continued this research topic and he adapted the proposed benchmark framework to distributed environments, provided implementation for distributed graph databases and EMF-based tools (e.g. IncQuery-D [Szá+14]) and also provided model metrics similar to the proposed graph query metrics [Szá+16].

The results of Thesis 1 are presented in Chapter 2 of the dissertation. Related publications are the following: [1], [6], [9], [10].

2.2 Model-generation method for metamodel-based test data generation

In this thesis a model generation method is presented that is applied for the test data generation for autonomous agents (e.g., autonomous robots).

Motivation My motivation is to provide a test data generation method to support the testing of the safety and robustness aspects of the context-aware behaviour of an autonomous agent. The goal of testing is the checking of the behaviour in case of various configurations of the environment (context). These contexts can be built up based on the combination and extension of the simple context descriptions included in the functional and safety requirements of the agent, while the coverage of the different context object types can be taken into account. Test contexts can be represented using instance models that are generated based on the context metamodel and testing goals. Moreover, properly setting the parameters (e.g., attributes or number of objects or relations) of such contexts they can be used as extreme contexts for robustness testing. The systematic generation of such contexts requires context modeling and model instance generation.

Methodology I propose an ontology based construction of the context metamodel, where the hierarchy and relations of the elements (objects and changes) in the environment can be precisely formulated. This context metamodel is directly utilized when defining and computing context coverage as an important coverage metric during testing.

The context metamodel may include abstract elements (concepts and relations), which approach supports the equivalence class based testing method, i.e., abstract concepts and relations are used to represent (from testing point of view) an equivalence class of concrete concepts and attributes. Allowing such abstract relations, values and classes provides the opportunity to the test engineer to define the test requirements using these high level abstract concepts instead of concrete values. Namely, during the context modelling and test specification phase, equivalence classes can be defined based on different attribute values (e.g., instead of concrete distance values a *near* and *far* abstract relations can be introduced). Moreover, on the basis of the context metamodel, well-formedness and semantic constraints can be expressed that are included in the functional specification and generally characterize the domain (determining the valid context configurations, e.g., the arrangement of objects or timing of changes) in the form of model patterns.

Model based test data generation [8] can be executed based on the context metamodel and various testing strategies, where a generated test data represents a context where the AS is to be tested.

According to the different testing strategies, different constraints can be constructed forming this way a constraint set that should be satisfied by the generated test data. This constraint-based test data (i.e., context model) generation can be interpreted as a constraint satisfaction problem and accordingly I propose a constraint solver based solution for test data generation.

The test data generation process is split into two phases that was motivated by the selected test strategies and also by the problem complexity and solution efficiency. In the first phase, the process takes care of the structural requirements of the test data (i.e., selection of the types and relations of context objects), but does not resolve the abstract relations and does not deal with the concrete attribute values. The constructed test data with abstract elements is not directly applicable for testing in simulators or in real environments, where concrete values are needed to specify the context of the autonomous agent. In the second phase, the abstractions are resolved and the previously undefined attribute values are calculated based on the remaining (e.g., semantic) constraints and the testing strategy.

In summary, I propose a test data generation methodology that includes a mapping from the test data generation problem to a constraint satisfaction problem. Based on this constraint satisfaction problem, a constraint solver can be executed, its output can be interpreted in the specified context and test data can be constructed in the form of instance models.

Technology On the technology level I selected the Satisfiability Modulo Theories (SMT) as a language for constraint satisfaction problems and EMF-based technologies for the model manipulation and generation tasks. Based on the proposed mapping a transformation is developed that transforms the test data generation problem into an SMT problem. This approach is implemented in a test data generation framework based on the Eclipse platform.

Application As an application use case I present how the proposed test data generation methodology can be applied in case of testing laser guided vehicles (an use case for test data generation included in the R3-COP research project [R3C11]). I present experiments in generating test data for safety and robustness testing and provide details about the scalability and performance of the proposed test data generation framework.

Thesis 2 Following a systematic method for testing the safety and robustness of the context-aware behaviour of autonomous systems, I developed a modeling approach for describing the context, and proposed model-based test context generation method in case of different testing strategies.

1. *Constructing the context metamodel*

   I proposed a modeling approach for describing the context of context-aware autonomous agents which includes the ontology-based construction of a metamodel, and the definition of syntactic (well-formedness) and semantic constraints.

2. *Definition of a constraint solver based test data generation framework*

   I proposed a constraint solver based model generation method for test data generation and for this purpose I provided a two phased model generation process with feedback options based on a mapping from the context (meta)model and the test data generation strategy to the SMT problem. I implemented the method in a test data generation framework, validated it in case of testing autonomous vehicles, and analyzed its scalability and performance.

The work underlying Thesis 2 benefited from joint research with János Oláh and Zoltán Micskei from our research group. Based on the joint idea of applying model generation method for test data generation, I proposed the ontology-based context modeling approach and developed the two phased SMT-based test data generation method.

Oszkár Semeráth in his own research improved the test data generation framework by integrating different constraint solver implementations, (e.g. Alloy [Jac02]).

The results of Thesis 2 are presented in Chapter 3 of the dissertation. Related publications are the following: [2], [8], [7] and [14].

2.3 Model validation method for checking the standard compliance of process models

This thesis focuses on the assessment and certification of the development (including design, implementation, verification and validation) processes of critical systems.

Motivation The motivation of this research is to support the assessment of development processes and toolchains by elaborating a formal verification technique that allows the automated checking of the compliance to the domain-specific standards. This assessment and certification is performed nowadays manually, and thus an automated solution could improve the efficiency of this task.

Methodology The standards addressing the development processes for safety critical systems (e.g., IEC61508 and EN50128) require measures and techniques to be applied during the development process depending on the safety integrity level of the application. Namely, for each development step the mandatory (M), highly recommended (HR), recommended (R) and not recommended (NR) measures and techniques are given in a tabular form, where each technique can be performed by concrete tools or toolchains.

Based on the domain-specific language definition approach a formal representation of these requirements can be constructed by the domain expert. If a concrete development process is represented by a process model, then model-based techniques could be used for checking the satisfaction of these requirements, i.e., the standard compliance of the development process.

On the analogy of classical model checking (that is applied to examine whether a formal model of behaviour satisfies some temporal requirements regarding state reachability) here it is checked whether a process model satisfies the standard requirements regarding the selection of measures and techniques.

I proposed a methodology that provides an ontology-based validation of development processes. Based on the (domain specific) standard the ontology of the methods and the requirements is constructed. The development process is modelled on the basis of the SPEM metamodel and this process model is transformed to an ontology-based process description. Using these representations, the standard compliance of the development process can be checked using reasoner and query tools.

To implement this approach, first, I formalized the requirements in standards that concern the selection of methods and tools, afterwards I defined (or adapted) modeling techniques to describe the relation of methods (e.g., the hierarchy of the methods or required combination of them), the capabilities of tools (or toolchains), and the construction of (domain-specific) development processes. Finally, I elaborated techniques that check the compliance of concrete development processes (constructed by process designers) to the requirements.

The formalization of the requirements and the model-based description of tools and methods open a way to support also the synthesis of processes and toolchains that are compliant to the standard. The process designer can be assisted by

- identifying missing methods and tools,
- proposing alternative solutions,
- offering a library of toolchain patterns,

Technology On the technology level I defined and implemented the transformation between the SPEM-based process models and ontology-based models. I constructed the validation framework based on the Eclipse platform and ontology tools, which includes a reasoner (e.g,. Pellet) and model query tools (e.g. SPARQL query tools).

Application I present how the proposed methodology is applied in case of the EN50128 standard for software development in railway control systems. I present the constructed ontology of the methods defined in the standard and show a validation of the model of a development process.

Thesis 3 I analyzed the requirements described in the standards for development and V&V processes of safety critical systems (especially in IEC61508 and EN50128), and proposed an ontology-based approach to construct their formal representation. Based on this approach I prepared a framework for checking the standard compliance of development processes.

1. *Formulating the requirements included in the standard using ontologies*

   I developed an ontology-based approach to representing the requirements contained in the standards for the development and V&V processes. I proposed a method for verifying these requirements using ontology-based reasoners and query languages. The method offers constructive feedback to reach the compliance with the standard and to support the tool selection. As a concrete study I formalized requirements for development processes in the railway area as contained by the EN50128 standard.

2. *Validating SPEM-based process models using ontology-based representation*

   I proposed a process validation approach on process models created using the industrially relevant SPEM metamodel. I constructed a model validation framework for checking the standard compliance of these process models, which applies a mapping from SPEM-based process models to ontology based representation.

The results of Thesis 3 are presented in Chapter 4 of the dissertation. Related publications are the following: [3], [4], [5], [11], [12], [13] and 15.

3 Applications of the new scientific results

This section summarizes the practical applications of the results of the dissertation.

3.1 Systematically analysis of MDE Scalability

The benchmark framework (*Thesis 1*) was adapted, extended and used in the Mondo (Scalable Modeling and Model Management on the Cloud) research project [MON15].

In collaboration with industrial and academic partners, the members of the research group defined model and model-query metrics and analyzed the scalability aspects of the model-driven technologies and tools using the proposed benchmark framework (the metamodel, queries and scenarios).

3.2 Test generation for autonomous agents

The context modeling approach (*Thesis 2*) was developed and applied in the R3-COP (Resilient Reasoning Robotic Co-operating Systems) research project [R3C11].

In collaboration with the industrial partners ontology-based context descriptions were constructed and these were mapped to domain-specific languages using the method presented in *Thesis 1*. Based on the context metamodel, abstract test data (test context models) were generated. The abstract test data were mapped to concrete test data using the method presented in *Thesis 2*.

3.3 Assessment of process models

The assessment of process models (*Thesis 3*) was first applied in the MOGENTES (Model-based Generation of Tests for Dependable Embedded Systems) EU FP7 research project [MOG08]. In this project, the standard compliance of test generation and testing steps (and the supporting toolchains) was checked.

Furthermore, in the CECRIS (CErtification of CRItical Systems) research project [CEC13] a complex process verification framework was developed in which the standard compliance of complete development processes of critical systems can be checked.

4 Publication list

Number of peer-reviewed publications:	19
Number of independent citations:	35

4.1 Publications related to the theses

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Thesis 1:	●					●			●	●					
Thesis 2:		●					●	●						●	
Thesis 3:			●	●	●						●	●	●		●

Journal paper

[1] Z. Ujhelyi, G. Bergmann, Á. Hegedüs, Á. Horváth, B. Izsó, I. Ráth, Z. Szatmári, and D. Varró. "EMF-IncQuery: An Integrated Development Environment for Live Model Queries". In: *Science of Computer Programming* 98 (Feb. 2015). DOI: 10.1016/j.scico.2014.01.004

[2] O. Semeráth, Á. Barta, Z. Szatmári, Á. Horváth, and D. Varró. "Formal Validation of Domain-Specific Languages with Derived Features and Well-Formedness Constraints". In: *International Journal on Software and Systems Modeling* (In press 2015)

International conference

[3] B. Polgár, I. Ráth, Z. Szatmári, and I. Majzik. "Model-based Integration, Execution and Certification of Development Tool-chains". In: *2nd ECMDA Workshop on Model-Driven Tool and Process Integration.* 2009, p. 35

[4] Z. Szatmári. "Standards-based Assessment of Development Toolchains in Safety-Critical Systems". English. In: *Proceedings of the 12th European Workshop on Dependable Computing, EWDC 2009.* Ed. by H. WAESELYNCK. Toulouse, France, 2009, 4 pages. URL: http://hal.archives-ouvertes.fr/hal-00381960

[5] Z. Szatmári, B. Izsó, B. Polgár, and I. Majzik. "Ontology-based assessment of software models and development processes for safety-critical systems". In: *Monographs of System Dependability Vol. 2.* Wroclaw, June 2010, pp. 1–12. URL: http://mycite.omikk.bme.hu/doc/88962.doc

[6] A. Pataricza, L. Gönczy, A. Kövi, and Z. Szatmári. "A methodology for standards-driven metamodel fusion". In: *Proceedings of the First international conference on Model and data engineering.* MEDI'11. Obidos, Portugal: Springer-Verlag, 2011, pp. 270–277. URL: `http://dl.acm.org/citation.cfm?id=2050199.2050234`

[7] Z. Szatmári, J. Oláh, and I. Majzik. "Ontology-based Test Data Generation using Metaheuristics." In: *ICINCO (2).* Ed. by J.-L. Ferrier, A. Bernard, O. Y. Gusikhin, and K. Madani. SciTePress, 2011, pp. 217–222. URL: `http://dblp.uni-trier.de/db/conf/icinco/icinco2011-2.html#SzatmariOM11`

[8] Z. Micskei, Z. Szatmári, J. Oláh, and I. Majzik. "A Concept for Testing Robustness and Safety of the Context-Aware Behaviour of Autonomous Systems". In: *Agent and Multi-Agent Systems. Technologies and Applications.* Ed. by G. Jezic, M. Kusek, N.-T. Nguyen, R. Howlett, and L. Jain. Vol. 7327. LNCS. Springer, June 2012, pp. 504–513. DOI: `10.1007/978-3-642-30947-2_55`

[9] B. Izsó, Z. Szatmári, G. Bergmann, Á. Horváth, I. Ráth, and D. Varro. "Ontology driven design of EMF metamodels and well-formedness constraints". In: *Proceedings of the 12th Workshop on OCL and Textual Modelling.* OCL '12. ACM. New York, NY, USA: ACM, 2012, pp. 37–42. DOI: `10.1145/2428516.2428523`. URL: `http://doi.acm.org/10.1145/2428516.2428523`

[10] B. Izsó, Z. Szatmári, G. Bergmann, Á. Horváth, and I. Ráth. "Towards Precise Metrics for Predicting Graph Query Performance". In: *28th IEEE/ACM International Conference on Automated Software Engineering (ASE 2013).* 2013, pp. 421–431

[11] B. Gallina and Z. Szatmári. "Ontology-based Identification of Commonalities and Variabilities among Safety Processes". In: *16 th International Conference on Product-Focused Software Process Improvement.* Dec. 2015. URL: `http://www.es.mdh.se/publications/4041-`

Local conference

[12] Z. Szatmári. "Standards-based Assessment of Development Toolchains". In: *Proceedings of the 16th PhD Minisymposium.* Budapest University of Technology, Economics, Department of Measurement, and Information Systems. 2009, pp. 20–21

[13] Z. Szatmári. "Ontology Based Assessment of Development Processes". In: *Proceedings of the 17th PhD Minisymposium.* Budapest University of Technology, Economics, Department of Measurement, and Information Systems. 2010, pp. 38–41

[14] Z. Szatmári. "Supporting the Testing of Autonomous Systems Using Context Ontologies". In: *Proceedings of the 18th PhD Minisymposium.* Budapest University of Technology, Economics, Department of Measurement, and Information Systems. 2011, pp. 34–37

Local event

[15] Z. Szatmári. "Fejlesztési folyamatok ontológia alapú ellenőrzése". In: *Műszaki Tudományos Füzetek - XV. FMTÜ Nemzetközi Tudományos Konferencia.* Erdélyi Múzeum-Egyesület, 2010

4.2 Additional publications

Journal paper

[16] I. Kocsis, Á. Tóth, Z. Szatmári, T. Dabóczi, A. Pataricza, and G. Guta. "Towards cyber-physical system technologies over Apache VCL". in: *International Journal of Cloud Computing* 5.1-2 (2016), pp. 91–111

International conference

[17] Z. Szatmári, A. Kövi, and M. Reitenspiess. “Applying MDA approach for the SA Forum platform”. In: *Proceedings of the 2nd workshop on Middleware-application interaction: affiliated with the DisCoTec federated conferences 2008.* ACM. 2008, pp. 19–24

Local event

[18] Z. Szatmári, Z. Micskei, and I. Majzik. “Monitor komponensek okostelefon platformra”. In: *Műszaki Tudományos Füzetek - XIX. FMTÜ Nemzetközi Tudományos Konferencia.* Erdélyi Múzeum-Egyesület, 2014

[19] I. Kocsis, A. Pataricza, G. Huszerl, B. Izsó, Z. Szatmári, Á. Tóth, D. Varró, and A. Voros. “Informatikaoktatás felhőben: egy új oktatási modell bevezetése”. In: *BeleSTEM - Felsőoktatási jó gyakorlatok a tudomány, a technológia, a műszaki tudományok és a matematika szolgálatában.* Tempus Közalapítvány, 2014, pp. 36–41

Master’s thesis

[20] Z. Szatmári. “Applying MDA approach for the SA Forum platform”. Master’s thesis. Budapest University of Technology and Economics, 2008

References

[Ber+08] G. Bergmann, Á. Horváth, I. Ráth, and D. Varró. “A Benchmark Evaluation of Incremental Pattern Matching in Graph Transformation”. In: *Proc. 4th International Conference on Graph Transformations, ICGT 2008.* Vol. 5214. Lecture Notes in Computer Science. Acceptance rate: 40%. Springer. 2008, pp. 396–410.

[Ber+10] G. Bergmann, Á. Horváth, I. Ráth, D. Varró, A. Balogh, Z. Balogh, and A. Ökrös. “Incremental Evaluation of Model Queries over EMF Models”. In: *Model Driven Engineering Languages and Systems.* Vol. 6394. Lecture Notes in Computer Science. 2010, pp. 76–90. DOI: `10.1007/978-3-642-16145-2_6`. URL: `http://dx.doi.org/10.1007/978-3-642-16145-2_6`.

[Ber+11] G. Bergmann, Z. Ujhelyi, I. Ráth, and D. Varró. “A Graph Query Language for EMF models”. In: *Theory and Practice of Model Transformations, ICMT 2011.* Vol. 6707. LNCS. Springer. 2011, pp. 167–182. DOI: `10.1007/978-3-642-21732-6_12`.

[Bio14] S. C. for Biomedical Informatics Research. *Protégé ontology editor.* `http://protege.stanford.edu/`. 2014.

[CEC13] CECRIS. *CErtification of CRItical Systems.* EU FP7 research project. 2013. URL: `http://www.cecris-project.eu/`.

[Cla14] Clarkparsia. *Pellet: OWL 2 Reasoner for Java.* `http://clarkparsia.com/pellet/`. 2014.

[Con+06] J. Connelly, W. Hong, R. Mahoney Jr, and D. Sparrow. “Current challenges in autonomous vehicle development”. In: *Defense and Security Symposium.* International Society for Optics and Photonics. 2006, pp. 62300D–62300D.

[DHK07] G. Dobson, S. Hall, and G. Kotonya. “A Domain-Independent Ontology for Non-Functional Requirements”. In: 2007. DOI: `10.1109/ICEBE.2007.76`.

[DLS05] G. Dobson, R. Lock, and I. Sommerville. "Quality of service requirements specification using an ontology". In: 2005.

[EMF14] EMF - The Eclipse Project. *Eclipse Modeling Framework*. `http://www.eclipse.org/emf/`. 2014.

[FG96] S. Franklin and A. Graesser. "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents". In: *Proc. of the Third International Workshop on Agent Theories, Architectures, and Languages*. 1996.

[Gli+09] B. Glimm, M. Horridge, B. Parsia, and P. F. Patel-Schneider. *A Syntax for Rules in OWL 2*. 2009.

[Hil11] G. Hillairet. *EMFTriple: a tool that brings semantic web languages to the EMF*. `http://code.google.com/a/eclipselabs.org/p/emftriple/`. 2011.

[Jac02] D. Jackson. "Alloy: a lightweight object modelling notation". In: *ACM Trans. Softw. Eng. Methodol.* 11.2 (2002), pp. 256–290. DOI: `http://doi.acm.org/10.1145/505145.505149`.

[MOG08] MOGENTES. *Model-based Generation of Tests for Dependable Embedded Systems*. EU FP7 research project ID: ICT-216679. 2008. URL: `http://mogentes.eu/`.

[MON15] MONDO. *Scalable Modeling and Model Management on the Cloud*. EU FP7 research project. 2015. URL: `http://www.mondo-project.org/`.

[Obj01] Object Management Group. *Model Driven Architecture — A Technical Perspective*. `http://www.omg.org`. 2001.

[Obj12] Object Management Group. *SysML v1.3 Specification*. `http://www.omg.org`. 2012.

[OMG11] Object Management Group. *Unified Modeling Language (UML) 2.4.1 Superstructure Specification*. formal/2011-08-06. 2011.

[OWL09] OWL Working Group. *OWL 2 Web Ontology Language*. `http://www.w3.org/2007/OWL/`. 2009.

[R3C11] R3-COP. *Resilient Reasoning Robotic Co-operating Systems*. ARTEMIS research project nr. 100233. 2011. URL: `http://www.r3-cop.eu/`.

[Rac12] Racer Systems GmbH. *RacerPro*. `http://www.racer-systems.com/products/racerpro/`. 2012.

[RN03] S. Russell and P. Norvig. *Artifical Intelligence. A Modern Approach*. Second. Pearson Education Inc., 2003.

[ROD10] T. Rahmani, D. Oberle, and M. Dahms. "An Adjustable Transformation from OWL to Ecore". In: 2010. URL: `http://dx.doi.org/10.1007/978-3-642-16129-2%5C_18`.

[Sir+07] E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz. "Pellet: A Practical OWL-DL Reasoner". In: *Web Semantics: science, services and agents on the World Wide Web* 5.2 (2007), pp. 51–53.

[SSW09] E. Sirin, M. Smith, and E. Wallace. "Opening, Closing Worlds - On Integrity Constraints." In: *OWLED*. Vol. 432. CEUR Workshop Proceedings. 15, 2009. URL: `http://dblp.uni-trier.de/db/conf/owled/owled2008.html#SirinSW08`.

[Szá+14] G. Szárnyas, B. Izsó, I. Ráth, D. Harmath, G. Bergmann, and D. Varró. "IncQuery-D: A Distributed Incremental Model Query Framework in the Cloud". In: *ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems, MODELS 2014*. Acceptance rate: 26%. Springer. 2014.

[Szá+16] G. Szárnyas, Z. Kővári, Á. Salánki, and D. Varró. "Towards the Characterization of Realistic Models: Evaluation of Multidisciplinary Graph Metrics". In: *ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, MODELS 2016.* 2016.

[Tao+10] J. Tao, E. Sirin, J. Bao, and D. L. McGuinness. "Integrity Constraints in OWL". In: *AAAI.* 2010.

[VB07] D. Varró and A. Balogh. "The model transformation language of the VIATRA2 framework". In: *Science of Computer Programming* 68.3 (2007), pp. 214–234.

[VSV05] G. Varro, A. Schurr, and D. Varro. "Benchmarking for graph transformation". In: *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium on.* IEEE. 2005, pp. 79–88.

[Wal+10] T. Walter, F. Silva Parreiras, S. Staab, and J. Ebert. "Joint Language and Domain Engineering". In: *Proc. of 6th European Conference on Modelling Foundations and Applications , ECMFA 2010, Paris.* Vol. 6138. LNCS. 2010.

[WSR10] T. Walter, H. Schwarz, and Y. Ren. "Establishing a Bridge from Graph-based Modeling Languages to Ontology Languages". In: *Proceedings of the of the Third Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE) at TOOLS.* 2010.

[WSS09] T. Walter, F. Silva Parreiras, and S. Staab. "OntoDSL: An Ontology-Based Framework for Domain-Specific Languages". In: *ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems, 12th International Conference, MODELS 2009.* Vol. 5795. LNCS. 2009, pp. 408–422.