MŰEGYETEM 1782

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

Incremental Model Queries in Model-driven Design

PhD Thesis Booklet

Gábor Bergmann
MSc in Technical Informatics

Supervisor:
Dr. Dániel Varró, DSc
associate professor

Budapest, July 2013

1 Model-driven engineering

1.1 The paradigm of model-driven engineering

The discipline of *model-driven engineering* (MDE) is gaining more and more prominence in certain areas of software and system engineering, primarily where faults can lead to human injury or significant damage in property, as it delivers higher-quality products in a shorter development lifecycle (see e.g. [HWR13]). According to MDE, the focus of the engineering process is on creating and analyzing models at different levels of abstraction, and deriving them from other models. These models conform to various *modeling languages*.

Modeling may start in an early phase of the engineering process, when *requirements* for the system under design are elicited. In light of the requirements, *system design* commences with creating high-level abstract models, then producing lower-level models enriched with design decisions and realization considerations after a series of refining steps. The models can be continuously verified in order to identify design faults as soon as possible.

Model Driven Architecture (MDA [OMG01]) by the Object Management Group is one of the MDE-based design approaches with the following characteristics. As the system under design is often required to be realized upon various target platforms corresponding to different technologies, such design processes may involve a *platform-independent model* (PIM), which encompasses significant application-specific behavioral principles and realization parameters, but technological aspects are not detailed yet. Afterwards, depending on the available technological context, the PIM may be mapped to various *platform-specific models* (PSM), from which program modules realizing the designed software components can finally be produced (partially automatically).

The concept of models in MDE is vague in the sense that it may even involve differential equations or spatial configurations in certain domains of system engineering. However, models involved in software engineering are essentially labeled graphs, and are typically sparse (i.e. the number of edges is roughly linearly proportional to the number of vertices). The labels applicable for vertices and edges of a modeling language (including types and attributes), along with their rules of interconnection, are defined by the *metamodel* of the language. Note that only the abstract, formal structure of the model (the so-called *abstract syntax*) is characterized here as graph-like; while the user-friendly visual depiction of the model (*concrete syntax*) can independently be of diagram, text, or any other form.

While there are some extensible formalisms intended as a general purpose way of representing models (such as UML [OMG11], SysML [OMG12b]), industrial practice seems to increasingly prefer *domain-specific modeling languages* (DSML) instead, which can be tailored to the needs of application domains and actual design processes. However, developing such a DSML (along with its associated tool support) is a cost-intensive task requiring special skills; therefore *domain-specific modeling* (DSM) technologies have emerged to provide aid. Built on the successful Eclipse platform [ECLb], the *Eclipse Modeling Framework* (EMF, [EMF]) is a leading DSM technology that is considered a de facto industrial standard. A DSML development process with EMF involves defining a metamodel (using the *Ecore* formalism), from which several components of the modeling tool can be automatically derived. Numerous generative and generic technologies assist the creation of tool support for EMF-based DSMLs; one can define a textual concrete syntax using a grammar or a visual concrete syntax using graphical elements, while code generators can be created by specifying textual templates for the modeling language.

1.2 Model transformation and model queries

Several steps of MDE can be partially or completely automated by *model transformation* (MT). First to gain wide-spread use was *code generation*, more precisely, model-to-text transformation (M2T). Code generators map models (such as PSM in case of MDA) to source code artifacts that will run on the target implementation platform. Deployment descriptors, test suites or documentation could be synthetized as well in addition to program code. Model-to-model transformation (M2M) is also gaining importance. A usage example from MDA would be the automated support of PIM-to-PSM mappings, which adds platform-specific knowledge to the PIM. Other kinds of model transformations may include the synchronization between different models representing the same system in different ways, for experts of different domains (e.g. security requirements and threat analysis). Finally, *model validation* or design rule checking can be thought of as a special case of model transformation, where the output is the detected violations of constraints.

A model transformation program can be implemented using any general-purpose programming language and toolkit. However, there are platforms specifically designed to support the creation of model transformations. There are basically three kinds of help one can expect from an MT system: (i) it can aid the transformation developer in processing the source model, such as in the form of queries, (ii) it can simplify the creation of elements of the target model, (iii) finally it can provide the control flow of the model transformation, managing state and traceability information.

From all these services, my work focuses primarily on investigating *declarative model queries*. Queries evaluated over model elements play an important role in code generation from PSM models, in M2M transformations for analysis or other purposes, in the *simulation* of behavioral models, in *state space exploration*, in *report generation*, etc. One of the important applications in DSM is automated validation of *well-formedness constraints* associated with the language. For instance, the AUTOSAR [AUT] standard defines hundreds of such constraints. DSM frameworks may provide an ability to express queries in a *query language* designed specifically for this task (see e.g. OCL [OMG12a], QVT [OMG08] queries), and evaluate them by a *query engine*.

The mathematical formalism of *graph rewriting* or *graph transformation* (GT [EEKR99, BV06]) provides a declarative, rule-based paradigm that can used, among many other purposes, to specify M2M transformations. The basic building block of GT is the *graph pattern*, which is essentially a declarative query: it identifies certain parts of the graph model based on structural and other criteria. Model manipulation is expressed in the form of *graph transformation rules* that consist of two graph patterns, and describe the transformation step where a subgraph matching the LHS pattern is substituted with a subgraph conforming to the RHS pattern. Therefore the GT formalism covers both the source model processing and target model manipulation aspects of MT, and in certain cases it does not require an externally defined control

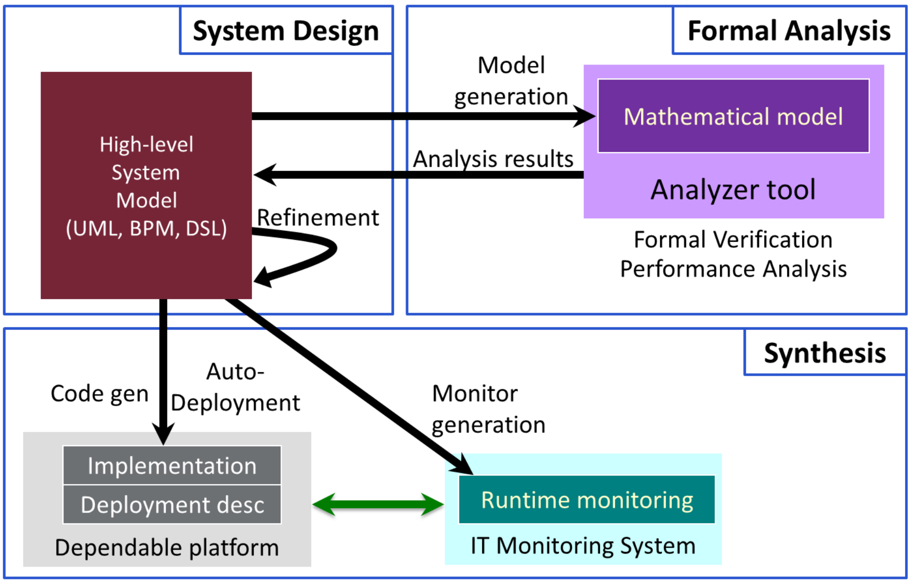

Figure 1: Roles of model transformations in MDE (inspired by [Pat06])

flow. Declarative model queries using graph patterns is a core topic of my thesis.

For the special task of model-to-model synchronization, *Triple Graph Grammars* (TGG, [Sch95, KW07]) and QVT [OMG08] provide an even higher level of abstraction in specifying transformations. TGG is based on GT, and TGG rules can be translated to GT rules. These languages allow very concise definition of common model mapping tasks; e.g. a single TGG rule can immediately be interpreted as a bidirectional, incremental (see Section 1.3) synchronization.

The above mentioned variety of roles and application areas of MT in MDE are illustrated on Figure 1.

1.3 Incremental, live and change-driven transformations

In a model-driven engineering process, models usually do not exist as static, immutable facts, but are rather undergoing constant evolution; implying that any previously conducted model analysis must be re-evaluated, and the effect of changes may propagate to other models as well. This evolution may happen due to requirement changes (potentially as late as years after the delivery of the system), or on a shorter time-scale the creation of ever newer model versions according to agile, iterative development methodologies, or simply the consequence of fixing problems detected by model validation. In fact, model editing actually consists of a sequence of small, atomic manipulation operations; this can also be regarded as the continuous evolution of a model, during which e.g. immediate feedback of model validation would be useful.

Repeatedly processing a (large scale) model after each small change can lead to significant performance issues. It can be more advantageous to apply *incremental* evaluation techniques [HLR06], taking into account the evolving nature of the model. In certain use cases (e.g. well-formedness checks) incremental queries have a great performance advantage **[11]**[ISR$^+$13].

Source incrementality is the property of a transformation that it only re-evaluates the modified parts of the source model. One of the central topics of my thesis is efficient evaluation of queries against evolving models through providing source incrementality.

Target incrementality, on the other hand, means that only the necessary parts of the target model are modified by the transformation, there is no need to recreate the new target model from scratch. The latter property, beyond direct gains in performance, has the benefit that connections, references between the target model and other external models are left intact and need not be recreated. Moreover, if the target model contains pieces of information (such as platform-specific design decisions in a PSM mapped from PIM) that do not stem from the source model, then the lack of target incrementality would lead to outright information loss.

After model evolution, the traditional MT approach restores the logical correspondence between source and target models by re-executing the transformation (which is efficient in case of source and target incrementality). A *live transformation* [HLR06], however, is continuously active, immediately reacting to events (changes of the source model) by keeping the target model synchronized. In this case source and target incrementality is highly beneficial.

Change-driven transformations [RVV09] are transformations that process changes of models – more precisely, even their specification is given in terms of consuming changes of the source model and producing changes of the target model. In this sense, source and target incrementality is a prerequisite for them. A transformation specified in a change-driven way can be executed as a live transformation, but this is far from being the only application scenario. It is possible to execute change-driven transformations even in cases where the source and target models of the M2M mapping are not actually available at the same computing resource, and can only communicate through the propagation of change information.

1.4 Example application domains

1.4.1 Modeling security requirements

Complex systems are typically designed to meet the needs of multiple *stakeholders*. *Requirement models* (such as the UML [OMG11] or SysML [OMG12b] standards, or KAOS [LL04]) help the designers obtain an overview of the needs and goals of various stakeholders, i.e. their *requirements*.

Security is a design concern aiming to avoid damages caused by adversarial persons, including damages to information assets (data security). It should not be confused with the related design concern of *safety*, which, regardless whether the damage is intentional, attempts to avoid primarily human injury, secondarily disproportionate physical damage to property. The thesis will address some concepts related to security.

System security is a broad area involving diverse design challenges. Software aspects include constructing secure cryptographic algorithms, communication protocols, and techniques for the prevention or detection of weaknesses and vulnerabilities in their implementation. Beyond software, technical aspects involve hardware solutions and also physical security. Finally, social aspects involve training humans involved with the system and establishing appropriate procedures for handling normal business and incidents.

However, none of these techniques can be applied unless we know what is there to protect and who should have access; therefore security design must be preceded by gathering security requirements.

Security requirement modeling [M+02, NNY10] is the process of creating and using requirement models that record the security needs and goals of stakeholders. For example, the modeling language Si* [MMZ07] can express trust between stakeholders and the delegation of responsibilities and permissions. The related field of *security risk modeling* [LSS+11] focuses on assessing the impact of potential threats to the system, the vulnerabilities against these threats, the characteristics of attackers that can exploit these vulnerabilities, and their associated risk of doing so.

With the application of model queries and transformations, security requirement models have the potential for conducting analysis that reveals inconsistent security needs, as well as for automatically providing solutions and guidelines for later phases of system design.

1.4.2 Embedded systems engineering in the automotive industry

As onboard electronic systems in automobiles are embedded systems with high safety requirements, the automotive industry benefits from rigorous methods of software and system engineering, calling for the use of model-driven techniques. In particular, proper analysis of models may lead to detecting design flaws early, which is a significant boost in efficiency. Due to the heterogeneity of software and hardware modules produced by various vendors, their integration is a challenge on the level of design models as well as on the level of implementation.

AUTOSAR (short for Automotive Open System Architecture, [AUT]) is an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers. The objectives of the AUTOSAR partnership include the implementation and standardization of basic system functions while providing a highly customizable platform which continues to encourage competition on innovative functions. The purpose of the common standard is to help the integration of functional modules from multiple suppliers and increase scalability to different vehicle and platform variants. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.

2 Challenges and contributions

2.1 Use cases of model queries

While declarative specification and execution of model transformations have now been widely studied and regarded as a significant field of research (see for instance the series International Conference on Model Transformations [ICM13]), the enabling support technology of declarative model queries deserves its own research focus. Use cases of model queries include the following:

Declarative model-to-model and model-to-text transformations. Model transformations provide automation for bridges between artifacts of an MDE workflow;

see Figure 1 for roles of MT. Transformations are commonly defined in rule-based declarative formalisms. Queries are used for specifying when and where one can apply the rules of the transformation specification; query evaluation then involves processing the source model to find the parts that will be transformed into the target model according to the rule.

Simulation of behavioral models with operational semantics defined using rules. Model simulation is the representation of system states and the application of state changes (transitions, evolution paths) to reach different system states, as described by the behavioral model. Model simulation is used for various dynamic analysis techniques such as model checking [JRG12], design space exploration [HV10], or stochastic simulation of trajectories to characterize typical behavior **[3]**. These analysis techniques may be used to verify a system under design, to assess its properties, and/or to support designing a safe and efficient system. Once again, the precondition of a transition rule is essentially a query that finds the applicable transitions in any given state of the model.

Analysis and reporting on models. Many kinds of static analysis can be formulated declaratively in a model query formalism, including gathering aggregated statistics, discovering correspondences of elements, or design rule checks by finding violations of well-formedness constraints and modeling conventions. In static analysis and reporting, queries can be used to provide a very direct and immediate feedback to the engineers.

2.2 Challenges

As models in engineering practice are often subject to change, their evolving nature raises many challenges with respect to the engineering process and model transformation in particular. These problems may range from organizational issues of a change request approval process to propagating or migrating changes between models (see also Section 1.3). My thesis focuses on a single overall challenge: **model queries over evolving models**.

Solving this challenge may present various ways to improve the engineering process in the previously indicated use cases. Source incrementality may radically increase query performance in model-to-model transformation, simulation and static analysis. In some cases, this might bring a qualitative improvement in addition to the quantitative one. For instance, immediate feedback in static verification may elevate modeling to a highly productive interactive process. Finally, extending the query language into a change-driven formalism may make the specification of change-propagating transformations and evolutionary analysis easier.

This top-level challenge naturally involves the following aspects:

Language. The first challenge is finding a query language that is:

- expressive enough to capture complex relationships of model elements such as rich structural interconnections, attribute conditions, quantification, aggregation and transitive reachability;

- concise enough to formulate complex relationships in a straightforward way, without wasting effort;
- compositional to support top-down or bottom-up thinking and reuse;
- intuitive to understand in light of its direct correspondence to model structure;
- able to express conditions relating to the change between two versions of a model, in addition to the structure of a single static model, in order to support change-driven transformation specification;
- declarative in order to support various evaluation strategies.

Evaluation method. The second challenge is finding an evaluation strategy to the declarative query language that is

- source-incremental;
- efficient in terms of execution time, and also regarding memory footprint;
- capable of taking advantage of the parallel execution provided by modern symmetric multiprocessing hardware;
- supports the features of the query language.

Adaptation to technological platforms. The final challenge is realizing both the language concepts and the evaluation engine in context of technologies with industrial relevance, namely:

- primarily the Eclipse Modeling Framework (EMF);
- alternatively relational databases, especially in-memory implementations;
- while addressing compatibility with all scenarios of model evolution in case of change-driven execution.

2.3 Contributions of the thesis

My thesis will offer the following improvements over the state-of-the-art of model query technology:

- I propose an incremental evaluation strategy for model queries formulated as graph patterns, and demonstrate its efficiency. (Contribution 1)
- I integrate this strategy into the industrial Eclipse Modeling Framework by designing a run-time translation layer and adapting the query syntax, and evaluate the performance of the resulting solution. (Contribution 2)
- I extend the query formalism to support change-driven transformations by transparently capturing changes of the model regardless of the scenario, and design scenario-specific strategies for execution. (Contribution 3)
- I provide bidirectional model synchronization, change impact analysis and consistency checking in the domain of security requirement engineering, by applying the above techniques. (Contribution 4)

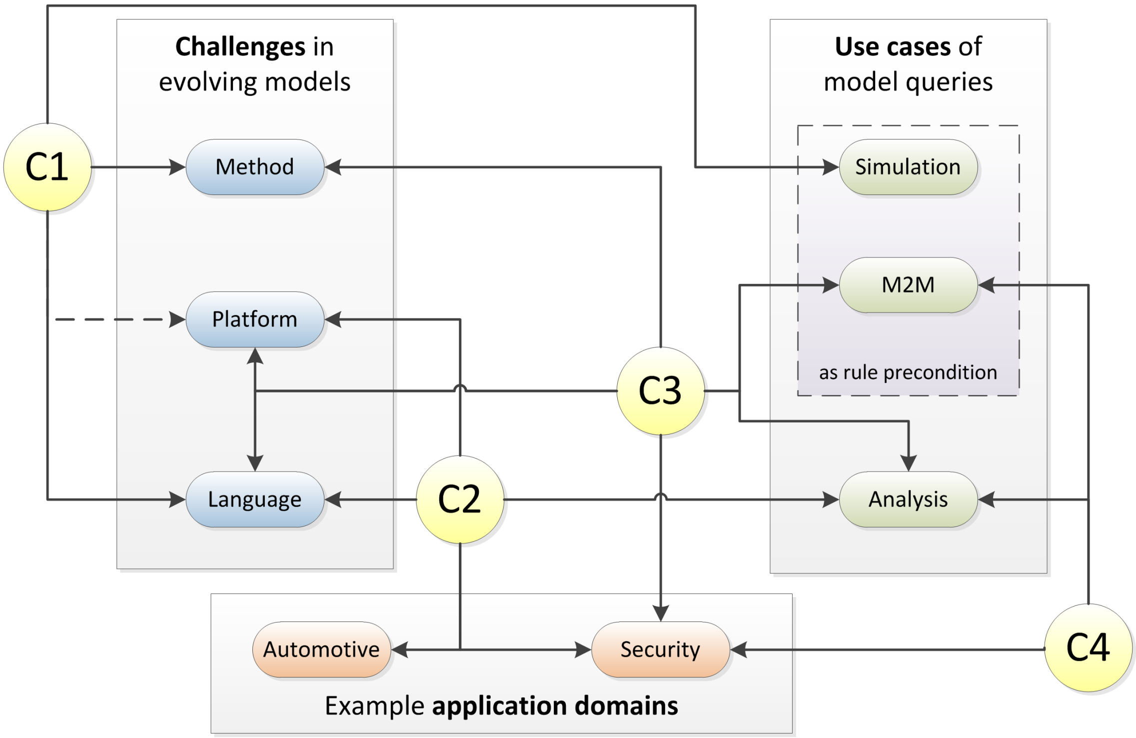

Figure 2: Contributions (C1-4), challenges, use cases and application domains

For each of these contributions, Figure 2 depicts the challenges that are addressed, as well as the application domains and use cases where the results are demonstrated.

3 New scientific results

I summarize the novel scientific contributions of my PhD thesis below. Each of the contributions stated here corresponds respectively to one of the goals that were set in Section 2.3.

3.1 Efficient, incremental pattern matching in a model-driven environment

In case of large-scale models, execution time may be a critical factor in the success of model transformation. A possible way to speed up transformations and queries is the application of incremental techniques. Source incrementality is not unavailable in most model transformation frameworks, and incremental graph pattern matching based approaches were in an early phase [BGT91, VVS06] of investigation at the beginning of my research.

Contribution 1 *I have adapted incremental algorithms of expert systems to realize graph pattern matching over large evolving models. I have demonstrated the efficiency of the approach in different application scenarios with performance measurements.*

1. ***Adapting Rete for incremental graph pattern matching.*** *I introduced a general theoretical framework for the semantics and algorithmic complexity of incremental pattern matching. In this context, I formalized the Rete [For82] algorithm from the field of rule-based expert systems. I have implemented the algorithm to operate on a rich graph pattern language as used in a model-driven context.* ***[23]***

2. ***Parallel incremental pattern matching.*** *I proposed parallel execution methods for incremental model query evaluation. I have identified three ways of parallelization: (a) concurrent execution of model manipulation and pattern matching (the maintenance phase in particular), and applying a multi-threaded strategy separately and independently (b) in model manipulation and (c) in matcher maintenance.* ***[26,4]***

3. ***Extending incremental pattern matching by transitive closure.*** *I proposed an efficient, incremental query evaluation method for handling generic transitive closure of graph edges and binary graph patterns.* ***[8]***

4. ***Adapting incremental graph pattern matching to relational databases.*** *I proposed an incremental method of query evaluation over models persisted in relational databases, which integrates with existing data manipulation software. I proposed a mapping from graph patterns to an event-driven SQL program that implements the TREAT [ML91] incremental algorithm.* ***[7]***

5. ***Quantitative performance analysis of incremental graph pattern matching.*** *I demonstrated the efficiency of the proposed incremental pattern matching strategy on model transformation benchmarks. I have identified scenarios (such as behavioral model simulation, M2M live synchronization) when its application is beneficial compared to traditional search-based approaches* ***[16,4,14,2]***

Several research results of my colleagues István Ráth and Ákos Horváth rely on the incremental pattern matcher I have implemented in the Viatra2 framework, including simulation-based analysis of DSM languages [RVV08], model-based design constraint satisfaction by design space exploration [HV09, HV10], or stochastic graph transformation **[3]**[THR10, KHTR10]. I applied the incremental query technique for simulation-based calibration of sensor networks **[13]**.

The core idea of incremental pattern matching was presented in the DSc thesis of my PhD supervisor Dániel Varró. I have developed the prototype implementation used for my investigations as a pattern matcher plug-in module for the Viatra2 [VIA] model transformation framework. Gergely Varró offered extensive help in starting my research as my master thesis supervisor. Elaborating measurements was joint work with István

Ráth and Ákos Horváth, and the strategies for combining different pattern matching approaches is now part of the PhD thesis of Ákos Horváth. Under my supervision, Tamás Szabó contributed the prototype implementation and conducted the measurements of incremental pattern matching extended by generic transitive closure. Under my supervision, Dóra Horváth contributed a prototype implementation and performed the measurements for incremental pattern matching over relational databases.

3.2 Incremental model queries over industrial EMF models

Model queries have various use cases in MDE. My aim is to allow declarative specification of these queries in a high-level language, and enable efficient evaluation.

Industrially accepted technologies offer various languages for specifying model queries (such as OCL [OMG12a]), but these formalism cannot always easily express the connection structure of several objects, which is important for use cases such as complex well-formedness constraints. Furthermore, most query evaluators for these languages are not incremental, and the exceptions are mostly academic tools. Therefore industrial platforms such as EMF could benefit greatly from a graph pattern based language that would be able to express complex queries that would be efficiently evaluated by an incremental matcher.

Contribution 2 *I proposed a declarative and expressive query language for specifying queries over the industrial Eclipse Modeling Framework. I have designed an incremental pattern matcher for the efficient evaluation of these queries. I have demonstrated the efficiency of the approach by performance measurements.*

1. ***Graph pattern based query language for EMF models.*** *I proposed a graph pattern based model query language for EMF models. The syntax is based on the pattern language [BV06] of Viatra2, extending it by new features including path expressions, and adjusting it to the characteristics of the EMF model representation.* ***[10]***

2. ***Incremental evaluation for EMF model queries.*** *I integrated the Rete-based incremental pattern matcher algorithm into the context of EMF models, and provided a translation that performs Rete maintenance according to the EMF notification scheme* ***[11,9,28,29,24]****.*

3. ***Performance analysis of model query frameworks over EMF.*** *I have demonstrated the expressiveness of the proposed language and the efficiency of incremental evaluation based on static model validation problems from the automotive domain.* ***[11]***

The EMF-IncQuery model query technology has been presented at multiple public tutorial sessions (including **[28,29]**), and gained significant attention in both academic and industrial audiences. These presentations have been a joint work with István Ráth, Ákos Horváth, Ábel Hegedüs, and others.

The defined query language contains ideas of István Ráth, Zoltán Ujhelyi and the authors of the original VTCL pattern syntax [BV06] on which it is based, and has since been used in further research (e.g. **[17]**). I built the prototype implementation into the EMF-IncQuery **[11]** tool in cooperation with Ákos Horváth. The above tutorials were joint work of this team. The case study and experiments from the domain of automotive industry is joint work with our industrial co-authors (András Balogh, Zoltán Balogh, András Ökrös). As a result of our joint work, EMF-IncQuery is now an official part of the Modeling project of the Eclipse Foundation [ECLb].

3.3 Supporting change-driven transformation specification by queries

The specification of transformations which process evolving models can greatly benefit from change-driven reactions. While some modeling platforms (VPM, EMF) provide notifications of elementary model deltas, their granularity is too low. In order to support the detection of complex changes, all preceding changes and their context must be taken into account. My results provide these capabilities for the concept of change-driven transformations that was proposed in [RVV09].

A further challenge is posed by the wide range of application scenarios where change detection may be necessary. One of these cases is live transformation (see **[15]**), where a continuously active transformation reacts immediately to model changes. However, if live transformation is not applicable, processing changes requires a different strategy.

Contribution 3 *By extending the formalism of graph patterns, I designed a new change pattern language for high-level, context-aware detection of structural changes of models. For each application scenario characterized by the information available on the model and its changes, I have proposed a dedicated strategy to efficiently evaluate a change pattern according to its formal semantics.*

1. ***Categorizing change scenarios.*** *I proposed a taxonomy of application scenarios for change-driven transformations, based on the available information describing the model and its change (such as model difference, change notification, archive version).* ***[1]***

2. ***Language for defining change patterns.*** *I designed a change-driven transformation language based on graph patterns. The language can express queries against the changes of a model, independently of the application scenario.* ***[25,27,1]***

3. ***Formal semantics of change patterns.*** *I formally defined the match set of a change pattern, in context of the change of a model.* ***[1]***

4. ***Pattern matcher strategies for change-driven transformations*** *I have designed strategies for evaluating change patterns in accordance with their formal semantics. There is a separate implementation strategy for each of the identified*

change scenarios, which efficiently computes the match set of change patterns based on the information available in the specific change scenario. ***[15,1]***

The proposed results were used by Ábel Hegedüs for back-propagating simulation results **[12]**. My work extends the concepts of change-driven transformations [RVV09], used in the PhD thesis of István Ráth.

3.4 Queries and transformation in modeling security requirements

In requirement modeling, requirements may have to be represented in multiple formalisms. Moreover, in security engineering processes, the requirement model is often interrelated with other models. Security experts can investigate security issues in requirements in a long and costly process, and their system of arguments can only be recorded in structured informal models. Automatic detection of simple security problems is not supported, and if the requirements evolve, all arguments need to go through the costly process of re-evaluation.

Contribution 4 *I designed an integrated environment for security requirement analysis, by using model queries and change-driven transformations,*

1. ***Bidirectional change-driven synchronization between security requirement models*** *I proposed change-driven, live transformations to support security requirement elicitation and analysis. I designed an environment architecture with a central abstract model (conforming to the SeCMER [MMP+11] formalism) in a bidirectional synchronization relationship with a different requirement model syntax (Si* [MMZ07]).* ***[19]***

2. ***Continuous validation of security criteria over evolving requirement models.*** *I proposed automated analysis of security requirements to check simple security criteria and identify violations. I applied graph pattern based queries to formalize the security constraints. I designed an implementation architecture where the requirements engineer is continuously informed by problem markers maintained according to incrementally evaluated queries.* ***[19]***

3. ***Analyzing the impact of requirement changes on informal argument models.*** *I proposed a traceability relationship between ground facts used in informal argument models and requirement model elements that serve as evidence. I designed a method based on change-driven techniques that identifies invalidated elements of the argument model for further consideration of argumentation experts, based on the changes of the requirement model.* ***[19]***

The work was performed in the security requirements workgroup of the SecureChange [EU 12] EU research project, in close collaboration with Fabio Massacci, Federica Paci, Thein Tun and Yijun Yu, as well as my PhD supervisor. The results presented above are my contributions. The proposed techniques were applied in the SeCMER prototype security requirements engineering tool of the SecureChange project.

4 Applications of new scientific results

Finally, I showcase some practical applications of my new conceptual results.

4.1 Incremental pattern matcher module of the Viatra2 model transformation framework

Viatra2 is a general-purpose graph transformation-based model transformation framework, which is part of the Generative Modeling Technologies project [ECLa] of the Eclipse Foundation [ECLb]. It has been developed for almost 10 years at the Department of Measurement and Information Systems, Budapest University of Technology and Economics. The incremental graph pattern matching module of the current Viatra2 version is built on conceptual results of Contribution 1.

Viatra2 itself has been applied in numerous international research projects, for tool integration (DECOS FP6, DIANA FP6 **[6]**, MOGENTES FP7, SecureChange FP7 [EU 12] EU projects), model validation (HIDENETS FP6 EU project), source code synthesis (SENSORIA FP6 **[5]**, E-Freight FP7 EU projects), and even behavior model simulation (sensor network analysis **[13]** in French-Hungarian intergovermental project). Viatra2 has regularly appeared in tool contests for transformation frameworks **[34,35]**, where the incremental pattern matcher module was used.

4.2 EMF-IncQuery

A recent project of the developer group behind Viatra2 is the EMF-IncQuery **[11]** framework, which enables wide-spread immediate application of results in Viatra2 and Contribution 2 on the EMF platform. The main run-time component of EMF-IncQuery implements the EMF-based query language and incremental pattern matching method of Contribution 2.

Through EMF-IncQuery, many of the results of the thesis can now be integrated with numerous open and proprietary products. Our research group, partners and early external adopters have already applied the tool in several projects. The tool was used in multiple national research grants (Jedlik, CertiMoT, TÁMOP) and for the realization of the SeCMER tool prototype (see Section 4.3) in the EU FP7 project SecureChange.

EMF-IncQuery already has a number of foreign uses. At least the following organizations have introduced EMF-IncQuery to their development practice, or conducted pilot investigations:

- Thales Group
- Itemis AG

- Obeo
- ThyssenKrupp Presta Hungary Ltd
- Montages
- evopro Informatikai és Automatizálási Kft.
- CERN
- CEA
- INRIA
- TU München
- KU Leuven
- University of York
- University of Nantes
- Austria Institute of Technology
- TU Eindhoven
- Universität Innsbruck

The following example applications of EMF-IncQuery have all been carried out independently of me:

- Incremental dependency analysis in large source code models at CERN
- Detection of change patterns in security architecture modeling at KU Leuven
- Declarative definition and incremental maintenance of derived features at Itemis
- Driving test oracles in MT testing at University of Nantes
- Providing query-driven soft interconnection of EMF models at BME

4.3 SeCMER tool prototype

The EU FP7 project SecureChange [EU 12] is concerned with the evolution of security critical systems; the demonstrator tool **[19]** of the project for security requirements engineering relies on several of my results. The tool provides the query and transformation based support proposed in Contribution 4, and EMF-IncQuery (see Section 4.2) played a big role in its implementation.

The validation of this demonstrator tool was performed in September 2011, according to the rules of the SecureChange project. At the validation event, the tool was presented through an air traffic management case study to participating flight security and air traffic control experts, who provided feedback that was incorporated in the tool.

The SeCMER prototype was highly acknowledged by project reviews.

5 List of publications

Number of publications: 36
Number of peer-reviewed publications: 26
Number of known independent citations: over 120

Journal papers (4)

[1] Gábor Bergmann, István Ráth, Gergely Varró, and Dániel Varró. Change-driven model transformations. Change (in) the rule to rule the change. *Software and Systems Modeling*, 11:431–461, 2012. Known independent citations: 4. Impact factor: 1.061.

[2] Ákos Horváth, Gábor Bergmann, István Ráth, and Dániel Varró. Experimental assessment of combining pattern matching strategies with VIATRA2. *International Journal on Software Tools for Technology Transfer*, 12:211–230, 2010. Known independent citations: 6.

[3] Paolo Torrini, Reiko Heckel, István Ráth, and Gábor Bergmann. Stochastic graph transformation with regions. *Electronic Communications of the EASST, Proceedings of the Ninth International Workshop on Graph Transformation and Visual Modeling Techniques*, 29, 2010. Known independent citations: 5.

[4] Gábor Bergmann, István Ráth, and Dániel Varró. Parallelization of graph transformation based on incremental pattern matching. *Electronic Communications of the EASST, Proceedings of the Eighth International Workshop on Graph Transformation and Visual Modeling Techniques*, 18, 2009. Known independent citations: 8.

Book chapters (2)

[5] Gábor Bergmann, Artur Boronat, Reiko Heckel, Paolo Torrini, István Ráth, and Dániel Varró. *Rigorous Software Engineering for Service-Oriented Systems - Results of the SENSORIA project on Software Engineering for Service-Oriented Computing*, chapter Advances in model transformation by graph transformations: Specification, Analysis and Execution. Springer, 2010. Known independent citations: 2.

[6] András Balogh, Gábor Bergmann, György Csertán, László Gönczy, Ákos Horváth, István Majzik, András Pataricza, Balázs Polgár, István Ráth, Dániel Varró, and Gergely Varró. Workflow-driven tool integration using model transformations. In Gregor Engels, Claus Lewerentz, Wilhelm Schäfer, Andy Schürr, and Bernhard Westfechtel, editors, *Graph Transformations and Model-Driven Engineering*, volume 5765 of *Lecture Notes in Computer Science*, pages 224–248. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-17322-6_11. Known independent citations: 9.

International conferences (10)

[7] Gábor Bergmann, Dóra Horváth, and Ákos Horváth. Applying incremental graph transformation to existing models in relational databases. In *Sixth International Conference on Graph Transformation (ICGT 2012)*, Bremen, Germany, 09 2012.

[8] Gábor Bergmann, István Ráth, Tamás Szabó, Paolo Torrini, and Dániel Varró. Incremental pattern matching for the efficient computation of transitive closure. In *Sixth International Conference on Graph Transformation (ICGT 2012)*, Bremen, Germany, 09 2012.

[9] Gábor Bergmann, Ábel Hegedüs, Ákos Horváth, Zoltán Ujhelyi, István Ráth, and Dániel Varró. Integrating efficient model queries in state-of-the-art EMF tools. In *TOOLS Europe 2012*, Prague, 2012. Springer. Acceptance rate: 31%. Known independent citations: 1.

[10] Gábor Bergmann, Zoltán Ujhelyi, István Ráth, and Dániel Varró. A graph query language for EMF models. In Jordi Cabot and Eelco Visser, editors, *Theory and Practice of Model Transformations, Fourth International Conference, ICMT 2011, Zurich, Switzerland, June 27-28, 2011. Proceedings*, volume 6707 of *Lecture Notes in Computer Science*, pages 167–182. Springer, 2011. Acceptance rate: 27%. Known independent citations: 1.

[11] Gábor Bergmann, Ákos Horváth, István Ráth, Dániel Varró, András Balogh, Zoltán Balogh, and András Ökrös. Incremental evaluation of model queries over EMF models. In *Model Driven Engineering Languages and Systems, 13th International Conference, MODELS'10*. Springer, Springer, 10/2010 2010. Acceptance rate: 21%. Known independent citations: 9.

[12] Ábel Hegedüs, Gábor Bergmann, István Ráth, and Dániel Varró. Back-annotation of simulation traces with change-driven model transformations. In *Proceedings of the Eighth International Conference on Software Engineering and Formal Methods*, pages 145–155, Pisa, 09/2010 2010. IEEE Computer Society, IEEE Computer Society. Acceptance rate: 22%. Known independent citations: 7.

[13] Gábor Bergmann, Miklós Molnár, László Gönczy, and Bernard Cousin. Optimal period length for the CGS sensor network scheduling algorithm. In *International Conference on Networking and Services*, pages 192–199. IEEE Computer Society, IEEE Computer Society, 2010. Known independent citations: 3.

[14] Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. Efficient model transformations by combining pattern matching strategies. In Richard F. Paige, editor, *Theory and Practice of Model Transformations, Second International Conference, ICMT 2009, Zurich, Switzerland, June 29-30, 2009. Proceedings*, volume 5563 of *Lecture Notes in Computer Science*, pages 20–34. Springer, Springer, 2009. Acceptance rate: 22%. Known independent citations: 3.

[15] István Ráth, Gábor Bergmann, András Ökrös, and Dániel Varró. Live model transformations driven by incremental pattern matching. In Antonio Vallecillo, Jeff

Gray, and Alfonso Pierantonio, editors, *Proc. First International Conference on the Theory and Practice of Model Transformations (ICMT 2008)*, volume 5063/2008 of *Lecture Notes in Computer Science*, page 107–121. Springer Berlin / Heidelberg, Springer Berlin / Heidelberg, 2008. Acceptance rate: 31%. Known independent citations: over 25.

[16] Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. A benchmark evaluation of incremental pattern matching in graph transformation. In Hartmut Ehrig, Reiko Heckel, Grzegorz Rozenberg, and Gabriele Taentzer, editors, *Proc. 4th International Conference on Graph Transformations, ICGT 2008*, volume 5214 of *Lecture Notes in Computer Science*, pages 396–410. Springer, Springer, 2008. Acceptance rate: ~40%. Known independent citations: over 10.

International workshops (7)

[17] Benedek Izsó, Zoltán Szatmári, Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. Ontology driven design of EMF metamodels and well-formedness constraints. In Mira Balaban, Jordi Cabot, Martin Gogolla, Claas Wilke, editors, *Proc. 12th Workshop on OCL and Textual Modeling*, pages 37–42. ACM New York, 2012. 10.1145/2428516.2428523.

[18] Gábor Bergmann, Fabio Massacci, Federica Paci, Thein Than Tun, Dániel Varró, and Yijun Yu. A tool for managing evolving security requirements. In Selmin Nurcan, Wil Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, and Clemens Szyperski, editors, *IS Olympics: Information Systems in a Diverse World*, volume 107 of *Lecture Notes in Business Information Processing*, pages 110–125. Springer Berlin Heidelberg, 2012. 10.1007/978-3-642-29749-6_8.

[19] Gábor Bergmann, Fabio Massacci, Federica Paci, Thein Tun, Dániel Varró, and Yijun Yu. A tool for managing evolving security requirements. In Selmin Nurcan, editor, *CAiSE'11 Forum at the 23rd International Conference on Advanced Information Systems Engineering*, volume 734, pages 49–56, London, UK, 06/2011 2011. CEUR-WS, CEUR-WS. urn:nbn:de:0074-734-7.

[20] Gábor Bergmann, Fabio Massacci, Federica Paci, Thein Tun, Dániel Varró, and Yijun Yu. SeCMER: A tool to gain control of security requirements evolution. In Witold Abramowicz, Ignacio Llorente, Mike Surridge, Andrea Zisman, and Julien Vayssière, editors, *Towards a Service-Based Internet*, volume 6994 of *Lecture Notes in Computer Science*, pages 321–322. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-24755-2_35.

[21] Ábel Hegedüs, Zoltán Ujhelyi, and Gábor Bergmann. Saying Hello World with VIATRA2 - A Solution to the TTC 2011 Instructive Case. In Pieter Van Gorp, Steffen Mazanek, and Louis Rose, editors, *TTC 2011: Fifth Transformation Tool Contest, Zürich, Switzerland, June 29-30 2011, Post-Proceedings*, volume 74, pages 302–324, Zürich, Switzerland, 11/2011 2011. Open Publishing Association, Open Publishing Association.

[22] Ábel Hegedüs, Zoltán Ujhelyi, and Gábor Bergmann. Solving the TTC 2011 reengineering case with VIATRA2. In Pieter Van Gorp, Steffen Mazanek, and Louis Rose, editors, *TTC 2011: Fifth Transformation Tool Contest, Zürich, Switzerland, June 29-30 2011, Post-Proceedings*, volume 74, pages 136–148, Zürich, Switzerland, 11/2011 2011. Open Publishing Association, Open Publishing Association.

[23] Gábor Bergmann, András Ökrös, István Ráth, Dániel Varró, and Gergely Varró. Incremental pattern matching in the VIATRA model transformation system. In G. Karsai and Gabriele Taentzer, editors, *Proc. Graph and Model Transformations (GRAMOT 2008)*. ACM, 2008. Known independent citations: over 25.

Domestic conference proceedings (3)

[24] Gábor Bergmann. Graph triggers and incrementally evaluated queries over EMF models. In *Proceedings of the 18th PhD Minisymposium*, pages 38–41, Budapest, 02/2011 2011. Budapest University of Technology and Economics, Department of Measurement and Information Systems, Budapest University of Technology and Economics, Department of Measurement and Information Systems.

[25] Gábor Bergmann. Contextual graph triggers. In *Proceedings of the 17th PhD Minisymposium*, page 22–25. Budapest University of Technology and Economics, Department of Measurement and Information Systems, Budapest University of Technology and Economics, Department of Measurement and Information Systems, 2010.

[26] Gábor Bergmann. Parallelization of incremental pattern matching in graph transformation. In *Proceedings of the 16th PhD Minisymposium*, page 10–11. Budapest University of Technology and Economics, Department of Measurement and Information Systems, Budapest University of Technology and Economics, Department of Measurement and Information Systems, 2009.

Hungarian-language conference article (1)

[27] Bergmann Gábor. Új formalizmus eseményvezérelt gráftranszformációhoz. In *XV. Fiatal Műszakiak Tudományos Ülésszaka*. Erdélyi Múzeum-Egyesület, Erdélyi Múzeum-Egyesület, 2010.

Tutorials (3)

[28] Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. Incremental Pattern Matching over EMF: a Tutorial on EMF-INCQuery, October 2010. Tutorials track of the ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems (MODELS).

[29] Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. Incremental evaluation of model queries over EMF models: a tutorial on EMF-INCQuery, June 2011. Tutorials track of the Seventh European Conference on Modelling Foundations and Applications (ECMFA 2011).

[30] Gábor Bergmann, Ábel Hegedüs, Ákos Horváth, István Ráth, Zoltán Ujhelyi, and Dániel Varró. Implementing efficient model validation in EMF tools. In *26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, Lawrence, Kansas, USA, 11/2011 2011. IEEE Computer Society, IEEE Computer Society. Known independent citations: 2.

EU research project deliverables (3)

[31] Gábor Bergmann, Elisa Chiarani, Edith Felix, Stefanie Francois, Benjamin Fontan, Charles Haley, Fabio Massacci, Zoltán Micskei, John Mylopolous, Bashar Nuseibeh, Federica Paci, Thein Tun, Yijun Yu, and Dániel Varró. *SecureChange Deliverable D3.2 Methodology for Evolutionary Requirements*, 2011. `http://www.securechange.eu/sites/default/files/deliverables/D3.2-%20Methodology%20for%20Evolutionary%20Requirements_v3.pdf`.

[32] Michela Angeli, Gábor Bergmann, Fabio Massacci, Bashar Nuseibeh, Federica Paci, Bjornar Solhaug, Thein Tun, Yijun Yu, and Dániel Varró. *SecureChange Deliverable D3.3 Algorithms for Incremental Requirements Models Evaluation and Transformation*, 2012. `http://www.securechange.eu/sites/default/files/deliverables/D3.3b%20Algorithms%20for%20Incremental%20Requirements%20Models%20Evaluation%20and%20Transformation.pdf`.

[33] Michela Angeli, Karmel Bekoutou, Gábor Bergmann, Elisa Chiarani, Olivier Delande, Edith Felix, Fabio Massacci, Bashar Nuseibeh, Federica Paci, Thein Tun, Dániel Varró, Koen Yskout and Yijun Yu. *SecureChange Deliverable D3.4 Proof of Concept Case Tool*, 2012. `http://www.securechange.eu/sites/default/files/deliverables/D3.4%20Proof-of-Concept%20CASE%20Tool%20%28Y3%29.pdf`.

Reports (2)

[34] Ábel Hegedüs, Zoltán Ujhelyi, Gábor Bergmann, and Ákos Horváth. Ecore to Genmodel case study solution using the VIATRA2 framework. In Pieter Van Gorp, Steffen Mazanek, and Arend Rensink, editors, *Transformation Tool Contest (TTC '10)*, Malaga, Spain, 07/2010 2010.

[35] Gábor Bergmann and Ákos Horváth. BPMN to BPEL case study solution in VIATRA2. In *5th International Workshop on Graph-Based Tools,*, Zürich, Switzerland, 07/2009 2009. Springer, Springer.

Master's thesis (1)

[36] Gábor Bergmann. Incremental graph pattern matching and applications. Master's thesis, Budapest University of Technology and Economics, May 2008.

Acknowledgements

First and foremost, I would like to thank Dr. Dániel Varró. As my advisor, he has continuously provided me with guidance, valuable feedback, and interesting challenges to solve.

I would have never gotten this far without the team of colleagues here at the Fault Tolerant Systems Research Group, Budapest University of Technology and Economics. I am grateful to Prof. Dr. András Pataricza and Dr. István Majzik, for providing financial and other kinds of support for my research during their leadership of the group. Dr. István Ráth, Ákos Horváth, Ábel Hegedüs, Zoltán Ujhelyi, Benedek Izsó, along with past collaborators András Ökrös, Dr. Gergely Varró, and others, deserve my thanks for being excellent team players, co-authors, co-developers, professional role models, and often a sounding board for my ideas. I had the great opportunity to build upon their work, and I find it a very satisfying feeling whenever I see that they are also benefit from my results.

As pursuing the PhD required a lot of work and effort, with many harsh deadlines along the way, I have amassed a great debt towards my family and my girlfriend by not always being able to spend enough time with them. I have to thank them for showing great tolerance, as well as invaluable support.

This research was realized in the frames of TÁMOP 4.2.4. A/1-11-1-2012-0001 „National Excellence Program – Elaborating and operating an inland student and researcher personal support system". The project was subsidized by the European Union and co-financed by the European Social Fund. The research was also partially supported by the Hungarian grant project TÁMOP 4.2.1. B-09/1/KMR-2010-0002, the Hungarian CERTIMOT (ERC-HU-09) project, the Hungarian–French Intergovernmental S&T Cooperation Program under the "Methods and algorithms to enhance the dependability of services in sensor networks" (F-Egide-PHC-19476SH, FR 6/2008), as well as EU projects SENSORIA (IST-3-016004) FP6 and SecureChange (ICT-FET-231101) FP7. My publishing was partially supported by the Schnell László Foundation.

Besides the financial support, some of these projects have also provided me with the opportunity for fruitful cooperation with excellent international colleagues who are too numerous to list, many of whom are now also my co-authors. I am extending my thanks to them as well.

References

[AUT] AUTOSAR Consortium. *The AUTOSAR Standard.* http://www.autosar.org/.

[BGT91] Horst Bunke, Thomas Glauser, and T.-H. Tran. An efficient implementation of graph grammars based on the RETE matching algorithm. In Hartmut Ehrig, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *Graph-Grammars and Their Application to Computer Science*, volume 532 of *Lecture Notes in Computer Science*, pages 174–189. Springer, 1991.

[BV06] András Balogh and Dániel Varró. Advanced Model Transformation Language Constructs in the VIATRA2 Framework. In *ACM Symposium on Applied Computing — Model Transformation Track (SAC 2006)*, 2006. In press.

[ECLa] Generative Modeling Technologies. http://www.eclipse.org/gmt/.

[ECLb] The Eclipse Foundation. *The Eclipse Project.* http://www.eclipse.org.

[EEKR99] Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors. *Handbook on Graph Grammars and Computing by Graph Transformation*, volume 2: Applications, Languages and Tools. World Scientific, 1999.

[EMF] The Eclipse Foundation. *Eclipse Modeling Framework.* http://www.eclipse.org/emf/.

[EU 12] EU FP7 research project ICT-FET-231101. *SecureChange: Security Engineering for Lifelong Evolvable Systems*, 2009-2012. http://www.securechange.eu.

[For82] Charles L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1):17–37, September 1982.

[HLR06] David Hearnden, Michael Lawley, and Kerry Raymond. Incremental model transformation for the evolution of model-driven systems. In Oscar Nierstrasz, Jon Whittle, David Harel, and Gianna Reggio, editors, *Proc. of the 9th International Conference on Model Driven Engineering Languages and Systems*, volume 4199 of *Lecture Notes in Computer Science*, pages 321–335, Genova, Italy, October 2006. Springer.

[HV09] Ákos Horváth and Dániel Varró. CSP(M): Constraint Satisfaction Problem over Models. In Andy Schürr and Bran Selic, editors, *Model Driven Engineering Languages and Systems, 12th International Conference, MODELS 2009, Denver, CO, USA, October 4-9, 2009. Proceedings*, volume 5795 of *Lecture Notes in Computer Science*, pages 107–121. Springer, 2009. Acceptance rate: 18%.

[HV10] Ákos Horváth and Dániel Varró. Dynamic constraint satisfaction problems over models. *International Journal on Software and Systems Modeling*, December 2010. Officially appeared online, DOI: 10.1007/s10270-010-0185-5, IF = 1,533.

[HWR13] John Hutchinson, Jon Whittle, and Mark Rouncefield. Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure. *Science of Computer Programming*, (0):–, 2013.

[ICM13] *International Conference on Model Transformation (ICMT)*. `http://www.model-transformation.org/`, 2008-2013.

[ISR+13] Benedek Izsó, Zoltán Szatmári, István Ráth, Ákos Horváth, Gábor Bergmann, Balázs Polgár, Gergely Varró, and Dániel Varró. *Efficient Instance-level Ontology Validation by Incremental Model Query Techniques*, 2013. `http://incquery.net/publications/trainbenchmark`.

[JRG12] Arash Jalali, Arend Rensink, and Amir Hossein Ghamarian. Incremental pattern matching for regular expressions. *ECEASST*, 47, 2012.

[KHTR10] Ajab Khan, Reiko Heckel, Paolo Torrini, and István Ráth. Model-based stochastic simulation of P2P VoIP using graph transformation. In *Proceedings of the 17th International Conference on Analytical and Stochastic Modeling Techniques and Applications*, 2010.

[KW07] Ekkart Kindler and Robert Wagner. Triple Graph Grammars: Concepts, Extensions, Implementations, and Application Scenarios . Technical Report TR-RI-07-428, 2007.

[LL04] Axel Lamsweerde and Emmanuel Letier. From object orientation to goal orientation: A paradigm shift for requirements engineering. In Martin Wirsing, Alexander Knapp, and Simonetta Balsamo, editors, *Radical Innovations of Software and Systems Engineering in the Future*, volume 2941 of *Lecture Notes in Computer Science*, pages 325–340. Springer Berlin Heidelberg, 2004.

[LSS+11] Mass Soldal Lund, Bjørnar Solhaug, Ketil Stølen, Mass Soldal Lund, Bjørnar Solhaug, and Ketil Stølen. The CORAS risk modelling language. In *Model-Driven Risk Analysis*, pages 47–72. Springer Berlin Heidelberg, 2011. 10.1007/978-3-642-12323-8_4.

[M+02] Haralambos Mouratidis et al. A natural extension of Tropos methodology for modelling security. In *Agent Oriented Methodologies Workshop. Object Oriented Programming, Systems, Languages (OOPSLA), Seattle-USA*. ACM, 2002.

[ML91] D. P. Miranker and B. J. Lofaso. The organization and performance of a TREAT-based production system compiler. *IEEE Transactions on Knowledge and Data Engineering*, 3(1):3–10, 1991.

[MMP+11] Fabio Massacci, John Mylopoulos, Federica Paci, Thein Than Tun, and Yijun Yu. An extended ontology for security requirements. In *CAiSE Workshops*, pages 622–636, 2011.

[MMZ07] Fabio Massacci, John Mylopoulos, and Nicola Zannone. Computer-aided support for secure tropos. *Automated Software Engg.*, 14:341–364, September 2007.

[NNY10] Armstrong Nhlabatsi, Bashar Nuseibeh, and Yijun Yu. Security requirements engineering for evolving software systems: a survey. *International Journal of Secure Software Engineering (IJSSE)*, 1(1):54–73, 2010.

[OMG01] Object Management Group. *Model Driven Architecture — A Technical Perspective*, September 2001. `http://www.omg.org/mda/`.

[OMG08] OMG. *MOF Query View Transformation Specification.* Object Management Group, April 2008.

[OMG11] Object Management Group. *UML Version 2.4.1*, August 2011. `http://www.omg.org/spec/UML/2.4.1/`.

[OMG12a] Object Management Group. *Object Constraint Language Specification (Version 2.3.1)*, 2012. `http://www.omg.org/spec/OCL/2.3.1/`.

[OMG12b] Object Management Group. *OMG SysML, v1.3*, June 2012. `http://www.omg.org/spec/SysML/1.3/`.

[Pat06] András Pataricza. *Model-based Dependability Analysis.* Hungarian Academy of Sciences, Budapest, 2006. DSc thesis.

[RVV08] István Ráth, Dávid Vágó, and Dániel Varró. Design-time Simulation of Domain-specific Models By Incremental Pattern Matching. In *2008 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2008.

[RVV09] István Ráth, Gergely Varró, and Dániel Varró. Change-driven model transformations. In *Proc. of MODELS'09, ACM/IEEE 12th International Conference On Model Driven Engineering Languages And Systems*, volume 5795/2009 of *Lecture Notes in Computer Science*, pages 342–356, 2009.

[Sch95] A. Schürr. Specification of graph translators with triple graph grammars. In *Proc. of the 20th Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG '94)*. Springer, 1995.

[THR10] Paolo Torrini, Reiko Heckel, and István Ráth. Stochastic simulation of graph transformation systems. In *Proc. of FASE2010, Fundamental Approaches to Software Engineering*, 2010.

[VIA] *VIATRA2 Framework.* An Eclipse GMT Subproject (`http://www.eclipse.org/gmt/`).

[VVS06] Gergely Varró, Dániel Varró, and Andy Schürr. Incremental Graph Pattern Matching: Data Structures and Initial Experiments. In Gabor Karsai and Gabi Taentzer, editors, *Graph and Model Transformation (GraMoT 2006)*, volume 4 of *Electronic Communications of the EASST*. EASST, 2006.