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

Search-Based Techniques in Model-Driven Development

PhD thesis booklet

Ákos Horváth
MSc in technical informatics

Supervisor:
Dr. Dániel Varró, PhD
associate professor

Budapest, February 2013

1 Preliminaries

1.1 Model-driven engineering

Models are prime artifacts of engineering. In system development, they have played an important role as a way to capture real world notions as well as abstract constructs. In fact, system architects have been using models and modeling techniques long before model-driven development emerged as a trend, e.g. in the form of entity-relationship diagrams, graph-like data structures, abstract syntax trees etc. However, the term Model-Driven Engineering (MDE) [BCW12] implies that *models play a central role* that encompasses the entire system development lifecycle, starting from requirement analysis, system design, implementation, to verification and even maintenance.

Model-driven engineering aims to increase the efficiency and productivity of the software development process by introducing precise engineering practices based on formal modeling techniques. It is based on the paradigm that the developer should work *from the beginning of the development* with high abstraction level models and through well-defined steps the complete process should be automated to the highest possible degree using various MDE techniques.

By this approach, design intelligence is applied to capture all relevant information in the form of abstract models. These models can be used for (i) *documentation* purposes to store well-structured information about the system-under-design, (ii) *early validation*, where important properties of the systems (such as performance, robustness, security, complexity) can be evaluated before actual implementation begins and (iii) *generative development*, where target design artifacts (source code, configuration tables, test cases, textual documentation, etc.) is (semi-)automatically derived by tools. All of these techniques aim at reducing costs and improving modularity and quality.

1.2 Design space exploration

In early phases of designing complex systems, models are not sufficiently detailed to serve as an input for automated synthesis tools. Instead, a design space is constituted by multiple models representing different valid design candidates. Design space exploration (DSE) aims at *searching and constructing such candidates* defined in the design space that satisfy all design constraints.

DSE is a process to quickly obtain feasible, "good enough" solutions which meet all structural and numeric design constraints in order to identify the most suitable design chosen by system architects based on various quality metrics such as performance, cost, power, reliability, etc. Typically, the best solution is *flexible* in the sense that it provides a trade-off between the optimal solutions with respect to a single quality metrics. Design space exploration is thus a challenging problem in many application areas including the design of critical embedded systems or dynamic reconfiguration of complex IT infrastructures, where MDE techniques have already been quite popular. These problems in an MDE context are frequently addressed as a specific sort of constraint satisfaction problem [Nee01].

Traditionally, most of these constraints and quality attributes were numeric in nature for expressing time, throughput, budget, memory limits, etc. However, the birth of modular software architectures in critical systems (like AUTOSAR [AUT] in the automotive domain or Integrated Modular Avionics (IMA) [RTCa] in the aeronautical domain) introduced *complex structural constraints*, which express connectivity restrictions for the graph-based model of the system under design. In addition, in many practical scenarios (like IT systems management), design space exploration is further complicated by the *continuous evolution of the system*, which imposes further constraints and quality metrics.

1.3 Model Transformation

Model transformation is the backbone of model-driven engineering. It aims to carry out automated translation within and between modeling languages. Over the years a large variety of different tools emerged using different concepts and techniques. Among those, one of the most popular approach is the declarative rule-based graph transformation [Roz97] paradigm.

Graph transformations

Graph transformation (GT) provides a declarative language for defining the manipulation of graph models by means of GT rules. A GT rule consists of (i) a left-hand side (LHS) and (ii) a right-hand side (RHS) graph. Model manipulations are carried out by replacing a match of the LHS in the model by an image of the RHS. This is performed in two phases. In the performance critical pattern matching phase, matches of the LHS are searched in the underlying model. In the updating phase, the selected matching parts are modified based on the difference of LHS and RHS.

In the recent years, with the widespread of MDE approaches in several application domains the need for fast and effective execution of model transformation defined by graph transformation rules on models ranging in the hundred thousand and even millions of elements has become a major challenge.

1.4 A motivating application domain

As MDE is attracting increasing attention in the *aeronautical system development* [CRH], the original approach needs to be adapted to be in-line with the rigorous DO-178B [RTCb] certification requirements imposed by civil aviation authorities like FAA and EASA. These require (i) tightly integrated V&V activities into the development process, (ii) continuous verification activity from early specification through design to development and (iii) end-to-end traceability through the complete development process. Moreover, the upcoming DO-178C certification guidelines has a dedicated subgroup (SWG4) for *Model Based Design and Verification*, which aims to define the specific certification requirements for model-driven engineering approaches based on the already available experience and considerations from DO-178B.

2 Challenges

My research has been motivated by the practical challenge to adapt model-driven development to safety-critical civil avionics systems. I aimed to tackle the development of ARINC-653 configuration artifacts for integrated modular avionics (IMA) systems using a systematic model-driven development process.

Unfortunately, despite the significant investment of research and development into the application of model-driven techniques for the development of embedded software [KSLB03] there have been very few methods [KG08] directly aiming configuration development. Lack of techniques in this direction started my research in *model-driven design space exploration* as certain parts of the configuration development for civil avionics system required automated techniques that can solve complex structural constraints over the system design already captured by a set of models.

As a prerequisite for applying any advanced MDE based technique for the automated generation of configuration artifacts in the selected avionics domain is the ability to handle huge models in the range of millions of elements. However, at the start of my research in *graph pattern matching* (2005) model transformation tools were just beginning to scale up to problem sizes of a few ten thousand or hundred thousand model elements, while complex industrial problems were at least an order of magnitude larger. This hindered their use not only for direct, batch model-to-model and

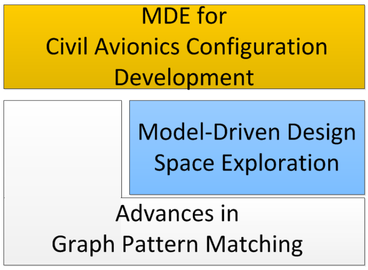

Figure 1: Overview on the Challenges and Contributions

model-to-code transformations but also their future application in different execution scenarios such as my model-driven design space exploration approach or model-based validation techniques like on-the-fly design contract evaluation. These applications do not only require fast model transformation executions but also immediate re-evaluation of model queries in case changes occur in their underlying model.

To sum up, Figure 1 gives an overview on the structure how my research directions are related to each other.

Challenge 1: How to speed up graph pattern matching to industrial size problems?

In order to provide acceptable performance in real-world application scenarios, graph transformation tools apply sophisticated pattern matching algorithms. These are mostly based on two conceptually different approaches: (i) local searches driven by search plans (like FUJABA [NNZ00] or GrGen [JBK10]) and incremental graph pattern matching [JT10, BOR+08] using caching mechanisms to store partial matches.

However, when I investigated existing tools for different implementations [GBG+06, Ren04, VVS06] of both local search and incremental pattern matching, I found that in general, (i) local search based approaches provide a good overall runtime performance on all different execution scenarios [27,21] using relatively low amount of memory compared to the underlying model, while (ii) incremental approaches has shown that in many application scenarios 17] – relevant to my other research directions – they lead to orders-of-magnitude increases in speed, for the price of using increased amount of memory especially when caching large fragments of the underlying model.

Unfortunately, several applications and industrial case studies revealed [16] that available memory can be insufficient (e.g., on restricted virtualized desktop environment) for caching match sets in case of the incremental approach. This problem is especially severe in design space exploration, where the traversed design space also needs to be stored in the memory.

Challenge 2: How to support structural constraint solving in evolutionary design space exploration problems?

Design space exploration problems in an MDE context are mainly tackled [WSNW07] as specific sort of constraint satisfaction problem (CSP). However, advanced constraint solvers typically apply certain restrictions for the CSP problem: (i) the domains of variables are required to be (a priori) defined, (ii) the number of variables are also a priori defined and finally, (iii) most approaches disallow the *dynamic* addition or retraction of constraints [MS00]. Furthermore, mapping graph models obtained in model-driven engineering to variables with finite domain can be a non-trivial task, especially when considering the evolution of models.

As a summary, existing constraint solvers fail to adequately handle flexible and dynamic structural constraints over graph-like models, which is necessitated for evolutionary design space exploration. Additionally, handling graph models directly in model-driven design space exploration problems necessitate both fast (i) graph pattern matching and (ii) manipulation, which emphasizes challenge 1.

Challenge 3: How to support the systematic configuration design of civil avionics systems?

The ARINC 653 standard [ARI] has taken a leading role within the aeronautical industry in the development of safety-critical systems based on the Integrated Modular Avionics (IMA) concept. One of the main promises of IMA is cost saving in reduced development, integration and verification and validation effort. In case of ARINC 653 compliant platforms many deployment and implementation details are defined in configuration tables. Typically, these configurations are hand defined by the system architect with limited tool support that only ease (i) the manipulation of its XML representation, (ii) their validation to the ARINC 653 schema definition and some consistency checks.

Despite the inherent complexity of avionics systems based on the ARINC 653 platform, current tools supporting configuration design offer very low-level support directly on the XML representation level. Existing tools lack support for (1) capturing the development process for configurations, (2) validating design constraints for configurations on-the-fly, (3) recording explicitly the critical design decisions made by the system architect, and (4) providing traceability between high-level requirements and the configuration tables, which require hand-crafted traceability lists. As a result, finding configuration design flaws as early as possible to reduce certification costs is a tedious design activity.

3 Research Method and new Results

3.1 Advances in Graph Pattern Matching

By reviewing how local search based graph pattern matching algorithms of existing model transformation optimize their search strategy [4], I found that research has been focusing so far on the performance optimal ordering of elementary graph pattern matching operations like the enumeration of objects and links of a certain type or the existence checks for links. However, the ordering of advanced pattern matching operations like attribute, injectivity and negative application condition constraint checking operations has been hard wired into the model transformation engines by using some simple heuristics, resulting in sub-optimal search strategies.

Moreover, by surveying incremental graph pattern matching techniques [17], I found that different model transformation scenarios require different graph pattern matching approaches for optimal run-time performance and memory consumption. However, none of the available model transformation frameworks allows the fine grained integration of different pattern matching algorithms.

In order to overcome these weaknesses, I proposed a general framework [4,18] for uniformly representing a large variety of search plan operations by expressing them as cost-weighted predicates. As an appropriate ordering of these predicates defines an executable search plan, this approach allowed to uniformly guide the pattern matching process for advanced graph patterns regardless of how the actual costs to different search plan operations are assigned. As a result, the different phases of pattern matching (e.g. cost assignment, generation of search plans, execution of search plans etc.) are fully separated and independent, thus they can be adapted to very different graph transformation engines and strategies (metamodel-based vs. model-based search plans [VVF05]). Furthermore, new types of predicates can be introduced easily by assigning appropriate costs without altering the algorithms for search plan generation.

Based upon this general search operation representation, I proposed a hybrid pattern matching approach [16,2], which enables the transformation designer to combine local search based and incremental pattern matching to adapt to memory constraints. Incremental pattern matching was introduced as a separate search plan operation. As a result of this approach, I demonstrated that in certain application scenarios [2] the hybrid approach outperforms the other two approaches and provides a good balance between memory consumption and runtime performance.

However, I noticed that selecting the appropriate matching strategy for complex model transformation programs requires a deep understanding of both pattern matching algorithms. Therefore, I examined typical transformation scenarios from the literature [24, 17, 27]. As a result, we found a list of various factors (metrics) [2], which we experienced to have significant effect on run-time performance and memory consumption. Based on this analysis, we defined guidelines for transformation designers when a graph pattern should be matched using INC or LS algorithm.

Thesis 1 I elaborated a general framework for uniformly representing a large variety of search plan operations to guide the graph pattern matching process for advanced graph patterns. Moreover, I integrated local search based and incremental pattern matching techniques resulting in a hybrid graph pattern matching algorithm.

1. **Generalized search graphs concept.** I defined a general search graph representation based on hypergraphs [4, 18], which can guide graph pattern matching for advanced graph patterns like edge identities, type variables, negative application conditions, attribute conditions, and injectivity constraints. Based on this representation, all search plan operations are uniformly represented as special predicates with heuristically assigned costs.

2. **Hybrid graph pattern matching.** I elaborated a hybrid graph pattern matching algorithm [2, 16], which is able to combine local search based and incremental graph pattern matching algorithms to select a good graph pattern matching strategy for composite graph patterns.

3. **Identification and categorization of key-factors for matching strategy selection.** By analyzing the local search based and incremental graph pattern matching algorithms, I identified and categorized key factors of typical usage scenarios, which have significant impact on execution time and the selection of the matching strategy [2, 16, 17].

Additionally, we demonstrated the feasibility of the graph pattern matching approaches by experimental evaluation. The evaluations were carried out on several model transformation tool contests to demonstrate the effect of the various optimization techniques [2, 27, 21, 24].

The work on local pattern matching techniques presented in this thesis have been carried in cooperation with Gergely Varró. Varró laid down the basics of our search graph driven pattern matching approach in his thesis [Var08] and also is the founder of the recursive pattern matching algorithm detailed in [18]. My own contributions in our cooperation are the generalization of the search graph concepts using hypergraphs to be able to represent any search operation in a generic way and the adaptation of the compile and runtime algorithms and data-structures to this novel hypergraph representation. The hybrid approach is completely my own contribution.

The identification and categorization of the key-factors for selecting proper matching strategy has been carried out with help from Gábor Bergmann and István Ráth. We achieved further results for scaling model transformations to industrial size problems as presented in [8].

3.2 Constraint Satisfaction Problems over Models

By surveying the literature and available structural constraint satisfaction programming environments and design-space exploration frameworks, I concluded that there is no approach that effectively supports both (i) the definition of domain specific hints to guide the traversal of the state space and (ii) dynamic manipulation of the constraint set without recalculating the solution from scratch. Such a style of specification and execution is well-suited for a number of design space exploration problems involving complex structural constraints and a predefined set of manipulation operations.

I extended the definition of constraint satisfaction problems [1, 15] by using graph patterns to define structural (first-order logic) constraints, and graph transformation rules as labeling operations. Informally, all graph pattern constraints need to be satisfied by the underlying model when searching for a specific goal. However, instead of simple variable substitution, the labeling phase applies graph transformation rules to carry out model manipulations on the underlying graph domain. As an analogy, my approach allows to (i) dynamically add/remove constraints from the problem domain, (ii) modify the domain of the variables during search and (iii) define structural constraints in a more natural way.

Thesis 2 I developed an approach for defining and solving dynamic and flexible structural constraint satisfaction problems with domain specific manipulation operations to drive the solving process.

1. **Structural CSP problems.** I elaborated a novel way to define static [29, 15], dynamic and flexible [1] CSP problems with complex structural constraints over graph based models.

2. **Structural constraint language based on graph transformation.** I defined a structural constraint language for the proposed CSP problems [1, 15], where constraints are defined by graph patterns and domain specific manipulation operations are specified as graph transformation rules.

3. **Efficient solver for the structural constraint problems defined by graph transformation rules.** I developed efficient solution algorithms [15] based upon incremental graph pattern matching for static, dynamic and flexible constraint satisfaction problems over models.

4. **Heuristic based traversal optimization.** I elaborated a guided traversal algorithm [1] using efficient heuristics based upon Petri-net based abstraction to minimize the traversed state space.

Additionally, by experimental evaluation and comparison with other open-source and industrial structural constraint solvers, we proved the feasibility of the developed structural constraint solver. The implemented structural constraint evaluation engine (VIATRA2 DSE) [1] is based on the upon the incremental graph pattern matcher of the VIATRA2 framework, which is part of the PhD work of Gábor Bergmann.

Additionally, this work is continued in the PhD work of Ábel Hegedüs who further advanced the capabilities of the VIATRA2 DSE framework by using rule-dependency and occurrence vector based guiding strategies [12,3].

Finally, the Petri net based abstraction technique is a work of Szilvia Varró-Gyapay and Dániel Varró. My contribution lies in its adaptation as a guidance strategy for design space exploration.

3.3 Application of Graph Transformation based Techniques for the Development of Avionics Systems

By investigating existing ARINC 653 configuration design environments, I concluded that they give only low-level support, usually directly on the XML representation, for the design and validation of configuration artifacts under development. However, as the complexity of the avionics systems grow - modern ARINC 653 based avionics system can have more than 40000 configuration elements [Wil07] - there is no approach that (i) support development starting from high-level models, (ii) allows fine grain (step level) validation of model changes throughout the whole development process and (iii) provides traceability through the complete development process as required by DO-178B.

I designed and participated in the development of the DIANA mapping framework [14] for systematically designing standard ARINC 653 configuration tables in the context of the DIANA EU FP6 project [DIA]. The framework is based on a platform independent architectural modeling language (PIADL) [DECa] that allows integration of industry leading architectural language AADL and system simulation language Matlab Simulink. The precise low-level details of a specific configuration for the ARINC 653 platform are captured by a platform specific Integrated Architecture Model (IAM). Mapping the PIADL to the IAM is handled by a complex interactive model transformation process that needs to bridge a large abstraction gap where critical design decisions are made by the system architect; thus it cannot be fully automated. Therefore, the mapping process is subdivided into well-defined design steps and precisely defined the contracts, interactions and interfaces of each step. Individual design steps are then organized into complex workflow-driven transformation chains, which are closely aligned with the designated development process followed by the airframer or function provider. Finally, configuration tables for the standard ARINC 653 and VxWorks specific Module descriptions are generated based on the IAM models.

Additionally, to support certification, end-to-end traceability links from the PIADL to the generated configuration files is generated using both (i) inter-model traceability based on an integration model and (ii) model-to-configuration traceability with XMI files connecting generated configuration elements to their corresponding model elements.

My research in the context of DIANA was focused on three topics: (i) adaptation of MDE techniques for defining a specific development process for ARINC 653 configuration design that is in-line with DO-178B (ii) definition and reuse of model-based validation techniques for early error detection and localization and (iii) implementation of end-to-end traceability from high-level models to generated artifacts through the complete process that conforms to the certification requirements.

Thesis 3 I proposed novel techniques for systematic model-based development of configuration artifacts for integrated modular avionics systems.

1. **Design contracts defined by graph patterns for model-driven development steps in avionics.** Following the concept of strong separation of components through precise interface descriptions as defined by DO-297 [RTCa], I elaborated a contract language [14, 6, 26] for model-driven development of avionics system configuration artifacts.
2. **On-the-fly validation of design contracts in avionics systems defined by graph patterns.** Aligned with the recommendation as proposed in DO-178B for early error detection, I defined an on-the-fly design contract validation approach based on incremental pattern matching [14, 13] to support the model-driven development process for avionics system configuration artifacts.
3. **End-to-end traceability in model-based design process for avionics configuration design.** I proposed an approach to support end-to-end traceability [14, 23] from high-level architectural models to XML based artifacts following the certification requirements defined by DO-178B [26,38].

The DIANA framework is built upon the foundations of systematic model-based design for critical embedded systems laid down by wide international collaboration in the DECOS [DECb] project with the Fault Tolerant Systems Research Group as key contributor, and used in the PhD thesis of András Balogh [Bal]. However, the target domain of the DECOS project was mainly time-triggered architecture for the automotive domain (AUTOSAR) that compared to the ARINC 653 platform and its DO-178B certification guidelines rise different design and certification requirements.

The mapping framework was developed in collaboration with Dénes Monostori, who was an MSc student under my supervision. Finally, the on-the-fly contract validation module is based on the EMF-INCQUERY framework [13], which was developed as a cooperative work between the members of the Fault Tolerant Systems Research Group and OptXWare Ltd.

My *primary contribution* lies here in the *adaptation of general modeling and model transformation techniques* in the context of avionics systems for systematic configuration design for ARINC 653 architectures with support for the automated generation of certification artifact as required by DO-178B.

4 Application of Results

In order to demonstrate the practical relevance of the approaches and methods outlined in the current thesis the current section highlights the applications of the result of my thesis.

4.1 Pattern Matching Algorithms in VIATRA2

The results of Thesis 1 provided the theoretical basis of implementing the local-search based graph pattern matching engine of the VIATRA2 framework. Additionally, it also serves as the backbone of the graph transformation module, which utilizes the search graph to generate the appropriate manipulation operations. Moreover, the hybrid pattern matching strategy is also integrated into the VIATRA2 framework and provides a fine-grained optimization capability between memory consumption and runtime performance for transformation designers. All modules are part of the current official release of the open source VIATRA2 framework hosted by the Eclipse Foundation[Ecl].

As a key part of VIATRA2 it has been applied in many research projects from various tool integration tasks (DECOS FP6, DIANA FP6, MOGENTES FP7 EU projects) to early-model based verification and validation (HIDENETS FP6 EU project) and source code and configuration generation (SENSORIA FP6, E-Freight FP7 EU projects).

4.2 VIATRA2 Design Space Exploration framework

The results of Thesis 2 have been implemented in the VIATRA2 Design Space Exploration (VIATRA2 DSE) framework, an add-on to the VIATRA2 DSE release. Its implementation has been evaluated and compared with several state-of-the-art constraint solvers like KORAT, GROOVE and the industry leading SICStus Prolog CLP(FD) library. As a result, the VIATRA2 DSE framework has provided comparable results and especially is case of dynamic problems outperformed all other approaches.

Moreover, it has been effectively used in the DIANA project for the allocation of safety-critical software components over airborne ready ARINC 653 compatible real-time operating system.

The framework is available from the department's VIATRA2 site at http://viatra.inf.mit.bme.hu.

4.3 Model-Driven Development of Integrated Modular Avionics Systems

In the context of the DIANA EU FP6 project, I participated in the development of a complete model-driven mapping framework from high-level platform independent models to configuration artifacts for the underlying ARINC 653 RTOS. I was responsible for the design and development of the complete mapping process carried out by the framework consisting more than 25 separate design steps. I also developed the necessary integration module to the VIATRA2 DSE solver to support the mapping of avionics software payload to the underlying implementation platform consisting of partitions and modules.

The developed framework was evaluated by leading industry partners like Embraer the 3rd largest airframer, the Dutch National Aerospace Laboratory and GMV Aeronautics the largest avionics company in Portugal. Based on their feedback we fine tuned the implementation and introduced the results (i) at the 2008 Farnborough Air Show as part of the DIANA tutorial on future 3rd generation IMA platform [35] and (ii) a joint publications with GMV at a premier industrial avionics conference: the 29th IEEE/AIAA Digital Avionics Systems Conference [14].

An other major follow-up of our approach is that Embraer initiated a cooperative research project with our group on a related topic.

4.4 EMF-INCQUERY

In order to apply our technology to a broader industrial domain, incremental pattern matching technology has been adapted to EMF, one of the most widely used modeling environment as of today. EMF-INCQUERY [13,20,9,7] provides an effective query API for EMF models with additional support for automated validation and change analysis. Apart from its application for the validation of design constraints. A collaborative work with Ábel Hegedüs and Tamás Szabó (an MSc student partly under my supervision) has started to adapt the VIATRA2 DSE framework to EMF using EMF-INCQUERY. The EMF-INCQUERY framework is a major research contribution of Gábor Bergmann's PhD thesis and lead by István Ráth a colleague of mine.

5 Publication list of Ákos Horváth

Number of publications: 41
Number of peer-reviewed publications: 27
Number of independent citations: 51

International, peer-reviewed journal papers (4)

[1] Ákos Horváth and Dániel Varró. Dynamic constraint satisfaction problems over models. *International Journal on Software and Systems Modeling*, 11(3):385–408, July 2012. DOI: 10.1007/s10270-010-0185-5, IF = 1,23.

[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.

[3] Ábel Hegedüs, Ákos Horváth, and Dániel Varró. Towards guided trajectory exploration of graph transformation systems. *Electronic Communications of the EASST, Petri Nets and Graph Transformations 2010*, 40, August 2011.

[4] Ákos Horváth, Gergely Varró, and Dániel Varró. Generic Search Plans for Matching Advanced Graph Patterns. *Graph Transformation and Visual Modeling Techniques in ECEASST*, 6:57–68, 2007.

National journal papers (1)

[5] Gergely Varró, Ákos Horváth and Dániel Varró. Automatic generation of transformer plugins by meta-transformations. *Journal of the Scientific Association for Infocommunications Hungary (Híradástechnika)*, (7):40–45, 2006.

Book chapter (1)

[6] András Balogh, Gábor Bergmann, György Csertán, László Gönczy, Ákos Horváth, István Majzik, András Patarciza, Balázs Polgár, István Ráth, Dániel Varró, and Gergely Varró. Workflow-Driven Tool Integration Using Model Transformations. In Gregor Engels, Claus Lewerentz, Wilhelm Schaefer, Andy Schuerr, and Bernhard Westfechtel, editors, *Graph Transformations and Model-Driven Engineering*, volume 5765 of *Lecture Notes in Computer Science*, pages 224–248. Springer, 2010. 10.1007/978-3-642-17322-6_11.

International conferences (12)

[7] Ábel Hegedüs, Ákos Horváth, István Ráth, and Dániel Varró. Query-driven soft interconnection of EMF models. In *ACM/IEEE 15th International Conference on Model Driven Engineering Languages & Systems*, Innsbruck, Austria, pages 134–150. 2012. Springer. Acceptance rate: 23%.

[8] Gábor Bergmann, Dóra Horváth, and Ákos Horváth. Applying incremental graph transformation to existing models in relational databases. In *Sixth International Conference on Graph Transformation*, Bremen, Germany, pages 371–385. September 2012. Springer.

[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, Czech Republic, pages 1–8. May 2012. Springer. Acceptance rate: 31%

[10] Zoltán Ujhelyi, Ákos Horváth, and Dániel Varró. Dynamic backward slicing of model transformations. In *International Conference on Software Testing and Validation, 2012*, Montreal, Canada, pages 1–10. April 2012. IEEE. Acceptance rate: 27%.

[11] Ábel Hegedüs, Ákos Horváth, István Ráth, Moisés Castelo Branco, and Dániel Varró. Quick fix generation for DSMLs. In *IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2011*, Pittsbourgh, PA, USA, pages 17–24. September 2011. IEEE Computer Society. Acceptance rate: 33%.

[12] Ábel Hegedüs, Ákos Horváth, István Ráth, and Dániel Varró. A model-driven framework for guided design space exploration. In *26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, Lawrence, Kansas, USA, pages 173–189. November 2011. IEEE Computer Society. ACM Distinguished Paper Award, Acceptance rate: 15%.

[13] Gábor Bergmann, Ákos Horváth, István Ráth, Dániel Varró, András Balogh, Zoltán Balogh, and András Ökrös. Incremental Evaluation of Model Queries over EMF Models. In *Proc. of MODELS'10, ACM/IEEE 13th International Conference On Model Driven Engineering Languages And Systems*, volume 6395 of *Lecture Notes in Computer Science*. pages 76–90. Springer, 2010. Acceptance rate: 21%.

[14] Ákos Horváth, Dániel Varró, and Tobias Schoofs. Model-Driven Development of ARINC 653 Configuration Tables. In *29th IEEE & AIAA Digital Avionics System Conference (DASC)*, pages 6.E.3–1 – 6.E.3–15, Salt Lake City, US, October 2010. IEEE.

[15] 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%.

[16] Gábor Bergmann, Ákos Horváth, István Ráth, and Dańiel Varró. Efficient Model Transformations by Combining Pattern Matching Strategies. In *Proceeding of the 2nd International Conference on Theory and Practice of Model Transformations*, volume 5563 of *Lecture Notes in Computer Science*, pages 20–34. Springer, 2009. Acceptance rate: 23%.

[17] Gábor Bergmann, Ákos Horváth, István Ráth, and Dańiel Varró. A Benchmark Evaluation of Incremental Pattern Matching in Graph Transformation. In Reiko Heckel and Gabriele Taentzer, editors, *Proc. of ICGT '08, 4th International Conference on Graph Transformation*, volume 5214 of *Lecture Notes in Computer Science*. Springer, 2008. Acceptance rate: 40%.

[18] Gergely Varró, Ákos Horváth, and Dániel Varró. Recursive Graph Pattern Matching With Magic Sets and Global Search Plans. In A. Schürr, M. Nagl, and A. Zündorf, editors, *Proc. 3rd Intl. Workshop on Applications of Graph Transformation with Industrial Relevance (AGTIVE '07)*, volume 5088 of *Lecture Notes in Computer Science*. Springer, 2008. Acceptance rate: 27%.

International workshops and tutorials (10)

[19] Gábor Bergmann, Ákos Horváth, István Ráth, and Dániel Varró. *Incremental Evaluation of Model Queries over EMF Models: A Tutorial on EMF-IncQuery*, volume 6698 of *Lecture Notes in Computer Science*, pages 389–390. 2011.

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

[21] 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, 2010.

[22] 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), `http://models2010.ifi.uio.no/tutorials.shtml/Bergmann`.

[23] Ábel Hegedüs, Zoltán Ujhelyi, István Ráth, and Ákos Horváth. Visualization of Traceability Models with Domain-specific Layouting. In *Proceedings of the Fourth International Workshop on Graph-Based Tools*, 2010.

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

[25] Ákos Horváth. Towards a Two Layered Verification Approach for Compiled Graph Transformation. In Reiko Heckel and Gabriele Taentzer, editors, *Proc. of ICGT '08, 4th International Conference on Graph Transformation*, volume 5214 of *Lecture Notes in Computer Science*. Springer, 2008.

[26] László Gönczy, István Majzik, Ákos Horváth, Dániel Varró, András Balogh, Zoltán Micskei, and András Pataricza. Tool Support for Engineering Certifiable Software. In *Proc. SafeCert 2008, Int. Workshop on the Certification of Safety-Critical Software Controlled Systems*, Elsevier, pages 68–73, 2008.

[27] Rubino Geiss, Gabriele Taentzer, Enrico Biermann, Dénes Bisztray, Bernd Bohnet, Iovka Boneva, Artur Boronat, Leif Geiger, Ákos Horváth, Ole Kniemeyer, Tom Mens, and Benjamin Ness. Generation of Sierpinski Triangles: A Case Study for Graph Transformation Tools. In A. Schürr, M. Nagl, and A. Zündorf, editors, *Proc. 3rd Intl. Workshop on Applications of Graph Transformation with Industrial Relevance (AGTIVE '07)*, volume 5088 of *Lecture Notes in Computer Science*. Springer, 2008.

[28] Ákos Horváth. Automatic Generation of Compiled Model Transformations. In *Proc. of Fifth Conference of PhD Students in Computer Science (CSCS)*, pages 54–55, Szeged, Hungary, Jun. 27-30 2006. Institute of Informatics of the University of Szeged.

Hungarian conferences (4)

[29] Ákos Horváth. Graph Transformation Based Constraint Solving. In *Proceedings of the 16th PhD Minisymposium*, page 12–15. Department of Measurement and Information Systems, BUTE, 2009.

[30] Ákos Horváth. Verification of Model Transformation. In *Proceedings of the 15th PhD Minisymposium*, page 54–58. Department of Measurement and Information Systems, BUTE, 2008.

[31] Ákos Horváth. Model Transformation Plugins on EMF Models. In *Proceedings of 14th PHD Mini-Symposium*, page 110–111. Department of Measurement and Information Systems, BUTE, 2007.

[32] Ákos Horváth. Modelltranszformációk implementációjainak automatikus, transzformáció alapú generálása. In Bitay E., editor, *Proc. of XI. Fiatal Mûszakiak Tudományos Ülésszaka (FMTU)*, pages 149–153, Kolozsvar, Romania, March 18–19 2006. Erdélyi Múzeum Egyesület.

Technical reports and online content (9)

[33] Zoltán Ujhelyi, Ákos Horváth, and Dániel Varró. Generic static analysis of transformation programs. Technical Report TUB-TR-09-EE19, Budapest University of Technology and Economics, June 2009.

[34] Zoltán Ujhelyi, Ákos Horváth, and Dániel Varró. Static type checking of model transformations by constraint satisfaction. Technical Report TUB-TR-09-EE20, Budapest University of Technology and Economics, June 2009.

[35] Ákos Horváth, Dániel Varró, and Tobias Schoofs. Model Driven System Development in the Avionics Domain, July 2008. Tutorial track on the 3rd generation IMA Platform at the 46th Farnborough Air Show.

[36] DIANA Consortium. DC 2.1: Specification of the Devlopment Means for AIDA. Technical report, The DIANA Project, EU FP6, 2008. `http://diana.skysoft.pt`.

[37] DIANA Consortium. DC 2.5: AIDA System Specification. Technical report, The DIANA Project, EU FP6, 2008. `http://diana.skysoft.pt`.

[38] DIANA Consortium. DC 3.1: Report on the Definition of the AIDA Development Means. Technical report, The DIANA Project, EU FP6, 2010. `http://diana.skysoft.pt`.

[39] DIANA Consortium. DC 4.2: Report on the Integration of the AIDA Simulations. Technical report, The DIANA Project, EU FP6, 2010. `http://diana.skysoft.pt`.

[40] DIANA Consortium. DC 4.4: Report on the Synthesizes of the Project and Lessons Learnt. Technical report, The DIANA Project, EU FP6, 2010. `http://diana.skysoft.pt`.

[41] DIANA Consortium. AIDA Platform ICD. Technical report, The DIANA Project, EU FP6, 2009. `http://diana.skysoft.pt`.

5.1 Acknowledgement

The current work was partially supported by the Hungarian CERTIMOT (ERC_HU-09-01-2010-0003) project, the European Union DIANA (AERO1-030985), SENSORIA (IST-3-016004) and SecureChange (ICT-FET-231101) research projects and a collaborative project with Embraer. Additionally, this work is connected to the scientific program of the "Development of quality-oriented and cooperative R+D+I strategy and functional model at BME" project. Finally, the theis is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

References

[ARI] ARINC - Aeronautical Radio, Incorporated. A653 - Avionics Application Software Standard Interface.

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

[Bal] András Balogh. Model Transformation-based Design of Dependable Systems. PhD Thesis, Budapest, Hungary.

[BCW12] Marco Brambilla, Jordi Cabot, and Manuel Wimmer. *Model-Driven Software Engineering in Practice*. Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, 2012.

[BKM02] Chandrasekhar Boyapati, Sarfraz Khurshid, and Darko Marinov. Korat: Automated testing based on java predicates. In *International Symposium on Software Testing and Analysis (ISSTA*, pages 123–133. ACM Press, 2002.

[BOR+08] Gábor Bergmann, András Ökrös, István Ráth, Dániel Varró, and Gergely Varró. Incremental pattern matching in the VIATRA transformation system. In *GRaMoT'08, 3rd International Workshop on Graph and Model Transformation*. 30th International Conference on Software Engineering, 2008.

[CRH] Lauerence Clarck, Terry Ruthruff, and Bary Hogan. Development of Lockheed Martin's, F16 Modular Mission Computer Application Software using MDA . Technical report, Lockheed Martin.

[DECa] DECOS - Dependabe Embedded Components and Systems consortium . The DECOS Platform Independent Metamodel, public deliverable. Technical report.

[DECb] DECOS project. (Dependable Embedded Components and Systems - EU FP6 Research Project. `http://decos.at`.

[DIA] DIANA project. Distributed, equipment Independent environment for Advanced avioNic Application - EU FP6 Research Project. `http://dianaproject.com`.

[Ecl] The Eclipse project. `www.eclipse.org`.

[GBG+06] Rubino Geiß, Gernot Veit Batz, Daniel Grund, Sebastian Hack, and Adam Szalkowski. Grgen:A fast SPO-based graph rewriting tool. In *Proc. of the 3rd International Conference on Graph Transformations (ICGT)*, pages 383–397, 2006.

[Int] Intelligent Systems Laboratory, Swedish Institute of Computer Science. Sicstus User's manual (2009). `http://www.sics.se/sicstus/docs/latest4/pdf/sicstus.pdf`.

[JBK10] Edgar Jakumeit, Sebastian Buchwald, and Moritz Kroll. Grgen.net. *International Journal on Software Tools for Technology Transfer (STTT)*, 12(3):263–271, July 2010.

[JT10] Frédéric Jouault and Massimo Tisi. Towards incremental execution of ATL transformations. In *Proceedings of the Third international conference on Theory and practice of model transformations*, ICMT'10, pages 123–137, Berlin, Heidelberg, 2010. Springer-Verlag.

[KG08] Amogh Kavimandan and Aniruddha Gokhale. Automated middleware qos configuration techniques for distributed real-time and embedded systems. In *Proceedings of the 2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS '08, pages 93–102, Washington, DC, USA, 2008. IEEE Computer Society.

[KSLB03] Gabor Karsai, Janos Sztipanovits, Akos Ledeczi, and Ted Bapty. Model-integrated development of embedded software. In *Proceedings of the IEEE*, pages 145–164, January 2003.

[MS00] Ian Miguel and Qiang Shen. Dynamic flexible constraint satisfaction. *Applied Intelligence, 13(3)*, pages 231–245, 2000.

[Nee01] Sandeep Neema. Analysis of matlab simulink and stateflow data model. March 2001.

[NNZ00] U. Nickel, J. Niere, and A. Zündorf. Tool demonstration: The FUJABA environment. In *The 22nd International Conference on Software Engineering (ICSE)*, Limerick, Ireland, 2000. ACM Press.

[Ren04] Arend Rensink. The GROOVE simulator: A tool for state space generation. In *Applications of Graph Transformations with Industrial Relevance (AGTIVE)*, pages 479–485, 2004.

[Roz97] Grzegorz Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation, volume 1: Foundations*. World Scientific, 1997.

[RTCa] RTCA - Radio Technical Commission for Aeronautic. Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations (DO-297), August 2005.

[RTCb] RTCA - Radio Technical Commission for Aeronautic. Software Considerations in Airborne Systems and Equipment Certification (DO-178B), 1992.

[Var08] Gergely Varró Varró. *Advanced Techniques for the Implementation of Model Transformation Systems*. PhD thesis, Budapest University of Technology and Economics, Department of Measurement and Information Systems, April 2008.

[VVF05] Gergely Varró, Dániel Varró, and Katalin Friedl. Adaptive Graph Pattern Matching for Model Transformations using Model-sensitive Search Plans. In Gabor Karsai and Gabriele Taentzer, editors, *Proc. of Int. Workshop on Graph and Model Transformation (GraMoT'05)*, volume 152 of *ENTCS*, pages 191–205, Tallinn, Estonia, September 2005. Elsevier.

[VVS06] Gergely Varró, Dániel Varró, and Andy Schürr. Incremental Graph Pattern Matching: Data Structures and Initial Experiments. In *Proceedings of the Second International Workshop on Graph and Model Transformation*, Electronic Communications of EASST. European Association of Software Science and Techn, September 2006.

[Wil07] Alex Wilson. The evolving ARINC 653 standrad and it's application to IMA, November 13th 2007. ARTIST2 meeting on Integrated Modular Avionics, Rome, Italy.

[WSNW07] Jules White, Dougles Schmidt, Andrey Nechypurenko, and Egon Wuchner. Introduction to the generic eclipse modelling system. *Eclipse Magazine*, (6):11–18, 2007.