Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Measurement and Information Systems

Advanced Techniques and Tools for Secure Collaborative Modeling

Ph.D. Thesis Booklet

Csaba Debreceni

Thesis supervisor:
Prof. Daniel Varro, D.Sc., Ph.D.

Co-supervisors:
Gabor Bergmann, Ph.D.
Istvan Rath, Ph.D.

Budapest
2019

1 Motivations

Modeling is considered as one of the most elementary discipline in engineering. Its purpose is to overcome complexity, increase understandability or ease design. When engineering complex and critical cyber-physical systems (like cars or aircrafts), the *model-based systems engineering (MBSE)* follows this concept by envisioning a process starting form the design of detailed system models through several well-defined abstraction and refinement steps. MBSE enables to detect design flaws early and automatically derive source code, documentation or configuration artifacts from high-quality system models. The adoption of MBSE by system integrators (like airframers or car manufacturers) has been steadily increasing in the recent years [WHR14].

Large-scale systems engineering projects are often carried out collaboratively to meet the aggressive delivery schedules while still maintaining a high standard of system correctness and safety. *Collaborative modeling* involves multiple engineers working together to develop system models concurrently. Modeling artifacts are traditionally developed either in an offline or online manner. In *offline collaboration*, engineers check out an artifact from a repository into a local copy and commit local changes to the repository in asynchronous (long) transactions. In *online collaboration*, engineers may simultaneously edit a model in short synchronous transactions which are immediately propagated to all other users. This strategy is similar to online collaborative office tools like Google Docs[GDocs].

As a common industrial practice, system integrators frequently outsource the development of various components to subcontractors in an architecture-driven supply chain where the collaboration between cross-organizational teams is facilitated by sharing models stored in model repositories [Roc+15]. However, effective collaboration is hindered by numerous factors.

Cross-organizational collaboration introduces significant challenges to protect the respective Intellectual Property (IP) of different parties. For instance, the detailed internal design of a component needs to be revealed to certification authorities, but it needs to be hidden from competitors who might supply a different component in the system. Furthermore, certain critical aspects of the system model may only be modified by domain experts with appropriate qualifications.

Access control is a process that grants/denies permission for resources when users attempts access them based on *access control policies*. *Access control management* is responsible to manage policies which are enforced during access control. Due to the lack of model-level access control management support for cross-company collaboration in existing modeling repositories, very strict infrastructure-level security policies are in place at companies, which prevent effective collaboration.

Effective collaborative development requires to prevent interference between teams potentially making updates to the same portion of the system model; ensure that local changes do not cause global inconsistencies; provide views of the system that are relevant to teams.

For code artifacts, these questions have traditionally been addressed by partitioning the code and assigning portions to different teams using file-level locking and textual merging techniques followed by testing/verification procedures such as integration testing or model checking.

Unfortunately, such traditional approaches for managing concurrent code development do not naturally extend to concurrent model-driven development. Partitioning into fixed model fragments is difficult due to the interconnected, graph-like nature of models. Fixed fragments are inflexible when faced with varying modeling tasks. Conflict avoidance techniques such as locking – that allows modifications only by the owner of the lock – lead to over-locking due to the high degree of interdependence between parts of a model. This significantly limits the degree of concurrent development and does not scale with the increasing number of collaborating teams. Model merging and conflict detection can be complex tasks, relying on comparing graphs instead of strings, and the interdependence within a model makes conflicts easy to introduce and hard to resolve. Finally, some model verification and validation techniques are too complex to be executed frequently, making quality control an expensive afterthought.

While traditional VCS frameworks provide efficient support for handling text-based design artifacts, their model-level counterparts require sophisticated techniques. Furthermore, the seamless integration of a collaboration layer with existing toolchains is a key industrial need.

2 General Collaboration Scenario

Figure 1 depicts a general scenario to develop complex system designs collaboratively which introduces the key concepts related to collaboration used throughout the thesis. Similarly to traditional software development, models are stored in a version control system (VCS) on the server side from and a local copy of the model is edited by collaborators on client-side.

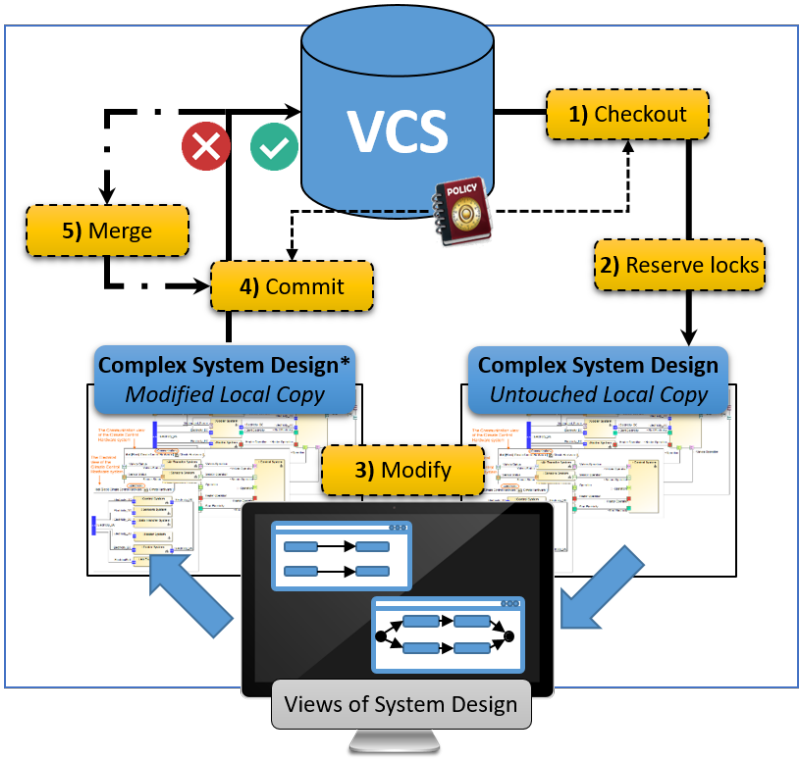

Figure 1. Overview of a general collaboration scenario

1. First, a collaborator with specific rights initiates a checkout action to download a local copy of the model. This step includes access control to ensure that collaborators access to those artifacts they are allowed to.
2. Before introducing modifications into the local copy of the model, collaborators can place locks to prevent contradicting modifications on model introduced concurrently by other collaborators.
3. System engineers usually introduce changes via views of the models representing only relevant aspect of the complex system design.
4. Changes are uploaded to the server by initiating a commit action. This step includes access control to ensure that collaborators modify only those artifacts they are allowed to. If the changes satisfies the access control policies, they can be accepted or rejected by the server. A commit is rejected if locks are violated (e.g. another collaborator's lock prevents the acceptance) or conflicts occurred (e.g. another collaborator concurrently modified the same parameter and it was not protected by locks).
5. If conflicts occurred during the commit, the collaborator who initiated the commit has to

execute a merge process on client side to eliminate conflicts between their local copy and latest version of the model in the VCS.

3 Overview of Challenges and Objectives

3.1 Challenges

Access Control in Collaborative Modeling An increased level of collaboration in a MBSE process introduces additional confidentiality challenges to sufficiently protect the intellectual property of the collaborating parties, which are either overlooked or significantly underestimated by existing initiatives (e.g. [CJC11; Tol16; Mar+14; EMFStore; Gen; Obeo], etc). Existing practices aim to restrict access to the files that store models that often result in inflexible fragmentation of models. In industrial practice, automotive models may be split into more than 1000 fragments, which poses a significant challenge for tool developer. This can be solved by fine-grained access control, where each model element and its features can have its own set of permissions. On the other hand, large industrial models can have millions of model elements, thus explicitly assigning permissions to each of them, as well as maintaining the permissions after changes to the model, would be labor-intensive and error-prone, and would make it difficult to understand the system of privileges.

> **Challenge C-I** How to specify and enforce high-level access control policies during collaborative modeling in a scalable way?

Conflict Prevention and Resolution Enabling a high degree of concurrent edits for collaborators is required to make the traditionally rigid development processes more agile. The increasing number of collaborators concurrently developing artifacts increases the probability of introducing conflicts. Conflicts occur when different collaborators modify the same part of the system model in a contradicting manner (e.g. a collaborator modifies a part that another one deletes). *Conflict avoidance* techniques such as locks [Kra+06; Alt+08; EMFStore; CDO; Tol16] try to prevent conflicts by letting the users request that certain engineering artifacts should be made unmodified by all other participants for a duration of time. But it usually leads to unnecessary preventions (locks) which significantly limits the degree of concurrent development and does not scale with the increasing number of collaborating teams. *Model merging* aims to resolve the conflicts, but, it can be complex tasks as the interdependence within a model makes conflicts easy to introduce and hard to resolve. Most merging approaches [EMF-Comp; EMF-Diff; Wes14; RC13; Bro+09] are semi-automated as they use a two-phase process: (i) first, they apply the non-conflicting operations and then (ii) let the user prioritize and select the operation to apply in case of two conflicting changes.

> **Challenge C-II** How to provide fine-grained conflict prevention and automatized conflict resolution strategies?

Bidirectional Synchronization of View Models Views are key concepts of domain-specific modeling in order to provide specific focus of the system to the engineers with various knowledge and expertise by abstracting the unnecessary details of the underlying model. Usually, these views are represented as models themselves (view models), computed from the source model. On one hand, the efficient forward propagation of changes from the source model to the views is challenging [Son+11; CJC11; Xio+07; Kol09], as recalculating the view from scratch has to be avoided to achieve scalability. On the other hand, the efficient backward propagation of complex changes from one or more abstract view models to the underlying source model resulting in valid

and well-formed models is also a challenging task [Sch94; Gho+15; Bru+15; QVT] which requires to limit the propagation to a well-defined part of the source model to achieve scalability.

> **Challenge C-III** How to derive and incrementally maintain view models and trace back complex changes to the underlying source models?

3.2 Objectives

In my thesis, I propose a secured collaboration framework to address the challenges. The complexity of challenges implies additional objectives to be addressed.

> **Objective O-I** Fine-grained access control management
>
> To address **C-I**, a generic modeling language will be used to capture fine-grained access control policies which needs to be evaluated efficiently in online and offline scenarios. Conflicting access control rules shall be handled where the results should provide a consistent modeling artifact.

> **Objective O-II** General secure collaboration scheme
>
> To address **C-I**, a provenly secure collaborative architecture shall include the enforcement of high-level fine-grained access control policies.

> **Objective O-III** Conflict resolution and handling
>
> To address **C-II**, a fine-grained property-based locking technique will be proposed to avoid conflicts during concurrent modification of models. An automated three-way model merge technique will be proposed to resolve conflicts when locking is unimpressive.

> **Objective O-IV** Synchronization of view models
>
> To address **C-III**, a novel bidirectional synchronization of view models will be proposed where the forward incremental synchronization is achieved by unidirectional derivation rules while the backward propagation of changes is generated using logic solvers.

4 Research Method

All the presented challenges are driven by industrial needs. In this thesis, I propose novel concepts and algorithms in the field of collaborative modeling to address these challenges. Algorithms and concepts are illustrated and evaluated on case studies carried out in multiple EU projects. I positioned my contributions wrt. state-of-the-art. Scalability evaluation has been carried out for all approaches. The thesis can be categorized as applied research in software engineering. The output of the research is software prototype implementation - which is a major output in mainstream software engineering research.

Offshore Wind-turbine Control Systems A case study of development offshore wind turbines extracted from MONDO EU FP7 project[Bag+14]. Offshore Wind Turbine Control Systems where of different artifacts and algorithms for controlling a wind turbine are specified and connected to sensors and actuators to actually operate the physical system.

Engineers develop the model in offline manner. Each user can download a model file containing those model elements that he is allowed to see. The user can then view, process, and modify his downloaded model file locally. A group of users may participate in online collaboration on the offshore platform to concurrently fine-tune the behavior of the system. Each user sees a live view of those parts of the model, that he is allowed to access. Changes need to be propagated

on-the-fly but they may be contradicting which could cause unexpected behavior. On one hand, it is necessary that the system model is consistent and well-formed all the time. On the other hand, the collaboration tool has to reject a modification immediately when it violates a security requirement.

TeleCare Systems TeleCare systems offer remote supervision for health care of elderly and physically less able people by collecting and process data from several sensors available in their home. A remote health care system is developed in the Concerto project [Conc] where *sensor* devices collect measurement results and report them to the GP's offices.

Employees (like *nurses* and other health care workers) are responsible for configuring TeleCare systems, but they may not qualified for setup such complex measurements. To ease the complexity of configuration, *dataflow* and *event ordering* views are used. A dataflow captures where to send certain type of *data*, while event ordering view defines the order of events including the trigger of measurements and reports.

5 Summary of the Research Results

The overview of the new scientific results are discussed in the following sections whereas Figure 2 depicts their dependencies with the corresponding challenges and objectives. At end of each section, I state my own contributions and emphasize their relations to the co-authors' contributions; I describe the uniqueness of the contributions related to the state-of-the-art; finally, I present the industrial application of the contributions.

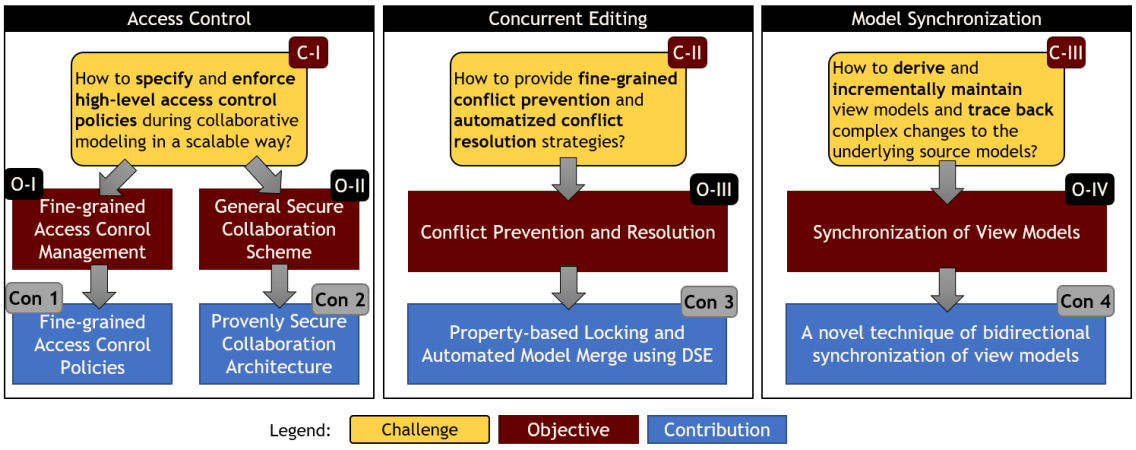

Figure 2. Overview of the contributions, objectives and challenges

5.1 Fine-grained Access Control Policies

To address **O-I**, a parameterizable fine-grained policy language is proposed to define concise access control rules and be able to fine-tune the resolution of conflicting rules. Rules can permit, obfuscate or deny permissions of read or write for model asset (object, link or slot) selected by a graph query. Default permissions are assigned to the rest of model assets.

Deterministic application of the access control rules is defined to obtain the same effective permissions after every execution where all internal consistency rules are taken into account. During the process of conflict resolution, our approach maintains a set of permission set. The initial permission set is obtained from rules and defaults. Then direct and indirect consequences of the access control rules are propagated in permission set (e.g. in slot should be visible, its owner

object should be visible as well). Our approach results in a conflict-free effective permission set where exactly one read and one write permissions is assigned to each model asset without exception.

Contribution 1 I proposed a domain-customizable modeling language to capture fine-grained access control policies and I realized a framework to efficiently evaluate the policies in online and offline scenarios. [*j*1], [*j*2], [*c*7], [*c*8], [*c*12],

- **1.1 Access Control Language.** I proposed a rule-based access control language to describe high-level and fine-grained policies in both online and offline scenarios. Rules may allow, obfuscate or deny read and/or write permissions of model parts identified by graph patterns. [*j*1], [*c*7], [*c*8], [*c*12],
- **1.2 Read and Write Dependencies.** I analyzed read and write dependencies implied by high-level access control policies as read and write permissions of a model part may depend on other model parts implied by internal consistency rules. [*c*12]
- **1.3 Deriving Effective Permissions.** I implemented a prototype framework to derive a set of effective permissions from access control policies in the context of models providing batch and incremental evaluation to support offline and online collaboration, respectively. [*c*8], [*c*12]
- **1.4 Evaluation.** I evaluated the scalability of the proposed prototype framework on a case study of offshore wind turbine controllers. [*c*8], [*c*12]

Related Contributions. The algorithm and its formalization to derive effective permission based on the proposed language is the contribution of Gabor Bergmann whereas my contribution are the definition of policy language, dependency analysis, implementation of the algorithm and its evaluation. Istvan Rath introduced fine-grained access control as an extra protection layer for modeling tools used in collaborative modeling on top of traditional version control systems.

Uniqueness. The approach allows system engineers to capture high-level policy rules instead of explicitly assigning permissions for each of them where the result of effective permission provides consistent model. Effective permission set can be incrementally reevaluated if a modification occurred in the model.

Application. The proposed concepts are applied by IncQuery Server[Heg+18] product which can provide additional incremental query evaluation services including change impact analysis, validation, ad-hoc queries as well as *fine-grained access control management* to several model repositories (e.g. NoMagic's Teamwork Cloud, OpenMBEE MMS).

5.2 Provenly Secure Collaborative Architecture

To address **O-II**, bidirectional model transformations are defined to (i) derive filtered views (*front models*) for each collaborator from the original model (*gold model*) containing all the information and to (ii) propagate changes introduced into these views back to a server in both *online* and *offline* scenarios. Access control policies consist of rules that allow, obfuscate or deny read and/or write permissions of model parts identified by graph patterns.

A collaboration scheme between the clients of multiple collaborators and exactly one server is described to support fine-grained access control in offline scenario. The server stores the gold models and the clients can download their specific front models. Modifications, executed by a clients, are submitted to the server and they are accepted if write permissions are successfully checked. Right after the submission, the changes are propagated to the other front model while read permissions are enforced. Finally, clients can downloaded their updated front models.

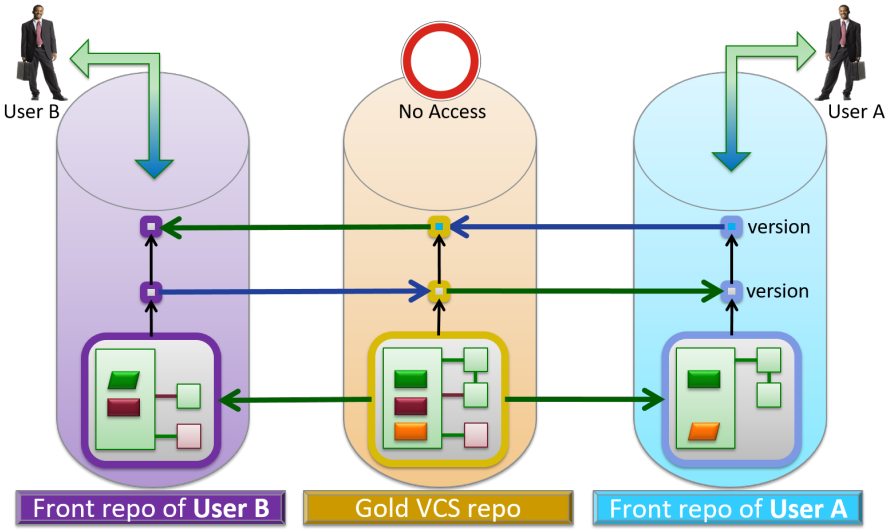

Figure 3. MONDO Offline Collaboration - Architecture

The scheme (depicted in Figure 3) is realized by extending *SVN*[SVN] using its hooks. The server and clients are realized as a *gold repository* and multiple *front repositories*, respecively. The *gold repository* contains *gold models*, but it is not accessible to collaborators. Each collaborator is assigned to a specific *front repository* containing a full version history of the front models. Change propagations are maintained between the repositories. As a result, each collaborator continues to work with a dedicated VCS as before, thus they are unaware that this front repository may contain filtered and obfuscated data.

Contribution 2 I formalized the enforcement of high-level fine-grained access control policies and realized provenly secure collaborative architecture the enforces such policies. [*j*1], [*j*2], [*c*7], [*c*6], [*c*9], [*c*10], [*c*14]

- **2.1 Formalization of Bidirectional Rules for Secure Views.** I formalized transformation rules to derive secure front models with respect to the read and write permissions. [*j*2], [*c*14]
- **2.2 Secure Collaboration Scheme.** I formalized a collaboration scheme as *communicating sequential processes* (CSP) to enforce high-level access control policies. I specified correctness criteria and proved the correctness of the scheme. [*j*2]
- **2.3 Realization of Secure Collaboration.** I realized the collaboration scheme in case of offline scenarios by extending an existing version control system to enforce fine-grained access control while collaborators can use off-the-shelf tools. [*j*1], [*c*6]
- **2.4 Evaluation.** I evaluated the scalability of the proposed architecture on a case study of offshore wind turbine controllers. [*j*2], [*c*9], [*c*14]

Related Contributions. The bidirectional transformation to enforce access control rules is the contribution of Gabor Bergmann whereas the concept of the common architecture to support both online and offline scenarios is the contribution of Istvan Rath.

Uniqueness. The provenly correct collaboration scheme is able to enforce fine-grained access control policies of modeling artifacts over existing version control system in case of offline scenarios. The scheme and its realization is demonstrated in MONDO Collaboration Framework as an integration with SVN[SVN].

Application. A usability evaluation of the prototype implementation of the proposed collaboration schema has been carried out by industrial partners *IK4-Ikerlan* and *UNINOVA* within the evaluation phase of MONDO project in the context of (1) a wind turbine case study (with small models but many users) and (2) a building information model (with models over 100,000 elements but fewer users). The number of concurrent users working on different views of the same model and the time for propagating changes and notifications among the concurrent users were evaluated both in offline and online cases, and the engineers could successfully collaborate in both cases. An experience report on wind turbines control applications development is also presented in [Góm+17] by IK4-Ikerlan.

5.3 Property-based Locking and Automated Model Merge using DSE

To address **O-III**, property-based locking is introduced to provide fine-grained conflict reduction while DSE-Merge is proposed to handle conflicts implied by concurrent editing.

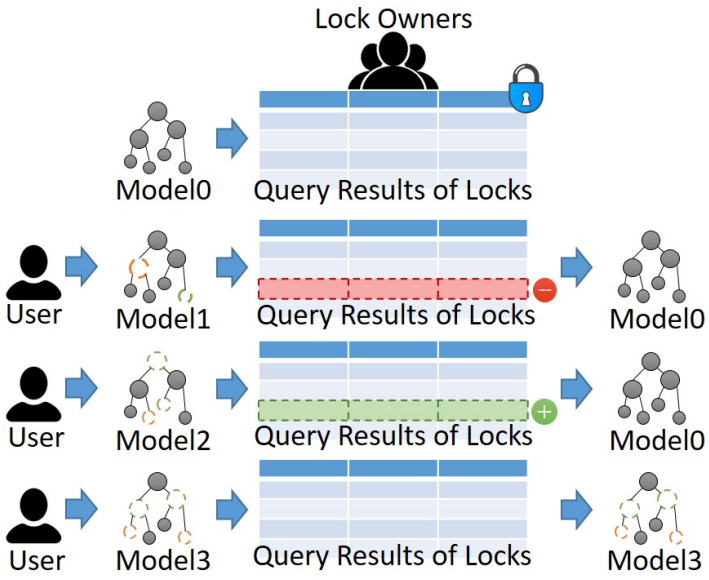

Figure 4. Behavior of Property-Based Locks

The concept of property-based locking (depicted in Figure 4) is described where collaborators request locks specified as a property of the model which need to be maintained as long as the lock is active. Hence, other collaborators are permitted to carry out any modifications that do not violate the defined property of the lock. The realization of property-based locking strategy is proposed as a common generalization of existing fragment-based and object-based locking approaches. Complex properties are described as graph patterns to express structural (and attribute) constraints for a model where the result set, i.e. the matches of graph pattern, can be calculated by pattern matchers or query engines. Only those modifications are allowed that do not change the result set of a list of queries.

DSE-Merge is proposed that exploits guided rule-based *design space exploration (DSE)* to automate the three-way model merge. Three-way model merge is applied to DSE problem (depicted in Figure 5) where the *initial model* consists of the original model O and two difference models (ΔL and ΔR); the *goal* is that there are no executable changes left in ΔL and ΔR; *operations* are defined by change driven transformation rules to process generic composite (domain-specific) operators; and *constraints* may identify inconsistencies and conflicts to eliminate certain trajectories. The output is a *set of solutions* consisting of (i) the well-formed merged model M; (ii) the

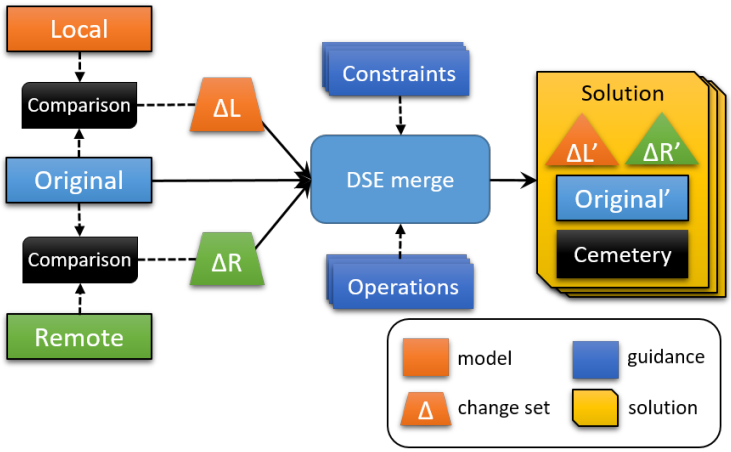

Figure 5. Architecture of DSE Merge

set of non-executed changes $\Delta L'$,$\Delta R'$; and (iii) the collection of the deleted objects stored in $Cemetery$.

Contribution 3 I proposed a fine-grained property-based locking technique to avoid conflicts and an automated three-way model merge technique to resolve conflicts. [c4], [j1], [c5], [c7], [c6], [c9], [c13], [c16]

- **3.1 Fine-grained Property-based Locking.** I proposed a property-based locking technique as generalization of traditional fragment-based and object-based locking techniques which captures fine-grained locks as graph patterns and exploits incremental query engines to maintain and evaluate locks. [j1], [c5], [c7], [c6], [c16]
- **3.2 Automated Model Merge using DSE.** I formalized an automated three way model merge technique by adapting rule-based design space exploration to derive consistent and semantically correct merged models. [c4], [c9], [c6], [c13]
- **3.3 Generic Scalability Benchmark.** I proposed a scalability benchmark for model merge by adapting an existing performance benchmark for model queries. [c13]
- **3.4 Evaluation.** I evaluated the scalability of the automated model merge and I compared the effectiveness of fine-grained property-based locking and traditional locking strategies for conflict prevention on a case study of offshore wind turbine controllers. [c5], [c13]

Related Contributions. The novel concept of property-based locking has been carried out in an international collaborative work [c16] with Marsha Chechik, Fabiano Dalpiaz, Jennifer Horkoff and Rick Salay where my contributions are the first adaption and implementation in a real practical setting and its evaluation in the context of MONDO EU FP7 research project. The concept of using DSE for merging purposes is the contribution of Istvan Rath, whereas VIATRA DSE implementation, on which DSE Merge relies, is the contribution of Akos Horvath [Hor13], Abel Hegedus [Heg14] and Andras Szabolcs Nagy[Abd+14]. Systematic evaluation of the efficiency of the DSE Merge technique from the user point of view is the contribution of Ankica Barisic[Bar+18] supported by MPM4CPS EU COST Action.

Uniqueness. Our property-based approach is general and can be used for both implicit locking of subtrees and set of elements or explicit locking of a certain element and its incoming and outgoing references. In addition it extends these lock types with the definition of properties to provide less restrictive locking for the collaborators.

The closest to our merge approach are [DRE14] and [Man+15], but we rely on state-based comparison, apply a guided local-search strategy (vs. [Man+15]), detect conflicts at runtime and allow complex generic merge operations (vs. [DRE14]). Internally, we uniquely use incremental and change-driven transformations to derive the merged models. Finally, we reported scalability of merge process for models which are at least one order of magnitude larger compared to [DRE14] and [Man+15].

Application. The efficiency of the DSE Merge technique has been systematically evaluated from the user point of view using a experimental software engineering approach. The empirical tests included the involvement of the intended end users (i.e. engineers), namely undergraduate students, which were expected to confirm the impact of design decisions. In particular, we asked users to merge the different versions of the same model using DSE Merge when compared to using Diff Merge. The experiment showed that to use DSE Merge participant required lower cognitive effort, and expressed their preference and satisfaction with it.

5.4 A Novel Technique of Bidirectional Synchronization of View Models

To address **O-IV**, an approach is introduced where view models are conceptually equivalent to regular models and they are defined using a fully declarative, rule based formalism. *Preconditions* of rules are defined by graph patterns, which identify parts of interest in the source model. *Derivation rules* then use the match set of a graph pattern to define elements of the view model.

Informally, when a new match of a query appears then the corresponding derivation rule is fired to create elements of the view model. When an existing match of a query disappears, the inverse of the derivation rule is fired to delete the corresponding view model elements.

View models derived by a unidirectional transformation are read-only representations, and they cannot be changed directly. To tackle this problem, we propose an approach (depicted in Figure 6) to automatically calculate possible source model candidates for a set of changes in different view models. First, the possibly impacted partition of the source model is need to be identified by observing traceability links to restrict the impact of a view modification. Then the modified view models and the query-based view specification are transformed into logic formulae. Finally, multiple valid resolutions of the source model are generated using logic solvers corresponding to the changes of view models and the constraints of the source model from the users can manually select a proper solution.

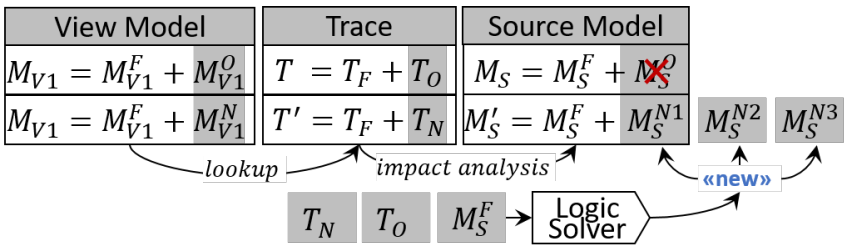

Figure 6. Overview of backward change propagation

Contribution 4 I proposed a novel technique of bidirectional synchronization of view models where the forward incremental synchronization is achieved by unidirectional derivation rules while the backward propagation of changes is generated using logic solvers. [c15], [c11], [c17], [c18], [e19]

4.1 Incremental Forward Synchronization. I formalized a fully forward incremental,

unidirectional synchronization technique of view models allowing chaining of views where the object of view model depend on the match set of the precondition of derivation rules. [*c*15], [*c*18], [*e*19]

4.2 Change Impact Analysis. I analyzed the impact of changes in underlying source models in case of backward propagation. The impacted part is added to the logic solver as additional constraints to calculate minimally modified source model candidates. [*c*15], [*c*11]

4.3 Realization of Forward Synchronization. I realized the incremental and forward view synchronization technique where elementary derivation rules are captured by graph patterns and the reactive synchronization process uses the Viatra EVM. [*c*15], [*c*18]

4.4 Evaluation. I evaluated the scalability of the proposed approaches on case studies from the avionics and the health-care domain. [*c*15], [*c*18]

Related Contributions. The query-based forward synchronization from an arbitrary model to its notional model for visual presentation is the contribution of Zoltan Ujhelyi[Ujh16], whereas my contribution generalizes his approach to view models. Introducing backward change propagation are shared contributions with Oszkár Semeráth[Sem19]. The transformation of the preconditions described by graph patterns and the impacted parts to first order logic is the contribution of Oszkár Semeráth whereas my contribution is the impact analysis for selecting changing parts.

Uniqueness. Definition of a view model is *unidirectional*, while the forward propagation of the *operation-based* changes are *live*, *incremental* and executed *automatically* that also maintains *explicit traces*. At backward propagation, using partitioning as an additional input of the logic solver improves scalability issues and limits the impact of changes to a well-defined part of the source model.

Application. The proposed forward incremental view synchronization and chaining of such view models are applied by Embraer[*c*18] to provide *functional architecture model* and its graphical representation as view models derived from low-level Simulink models. Backward change propagation is used by the CONCERTO project for the synchronization of certain views and the underlying systems in the TeleCare domain.

6 Tooling

The prototype implementation of the proposed concepts are developed within the MONDO Collaboration Framework whereas the view model synchronization is the fundamental approach of the Viatra Viewers component.

6.1 MONDO Collaboration Framework

The MONDO Collaboration Framework[1] aims to extend traditional version control systems with advanced secure collaborative modeling features such as fine-grained model-level access control, property-based locking, automated model merge in both offline and online collaboration scenarios.[2]

[1] Source code of the framework can be found at the following link: https://github.com/FTSRG/mondo-collab-framework

[2] Screencast demonstration of the framework can be found at the following link: https://youtu.be/Ix3CgmsYIU0

Compared to traditional file-based collaborative version control systems (e.g. SVN, Git), we provide an extra layer of fine-grained server-side access control and locking that flexibly decouple model hierarchy from permissions. Compared to model repositories (e.g. CDO[CDO], EMFStore[EMFStore], MetaEdit+[Tol07]), our collaboration framework is transparent, i.e. it does not require any modifications to existing front-end (single-user) modeling tools.

6.2 VIATRA Viewers

The goal of the VIATRA Viewers component is to help developing model-driven user interfaces by filling and updating model viewer results with the results of model queries. The VIATRA Viewers component can bind the results of queries to various viewer components such as JFace Viewers or GraphViewers.[VVi]

The implementation of VIATRA Viewers uses the proposed forward incremental synchronization approach. In this case, the view model is defined as a notion model consisting of *items*, *edges*, *containment edges* and *formatting settings*, while the source model can be any be EMF model. Then the notational view model can be used as the input of one of the available renderers that displays the model using e.g. JFace[JFace] viewers, Zest[Zest] or yFiles[YFiles] for Java graph visualization engines.

7 Future work

For each contribution, we identify several research topics for further investigation as follows:

Fine-grained Access Control Management We plan to (i) extend the security language with preselected sets of policy options (such as the resolution strategies of XACML) and accompanying "design patterns" on how policies should be constructed (ii) investigate incrementality of conflict resolution algorithms with respect to policy changes.

General Secure Collaboration Scheme We would like to (i) address the limitations of the current state providing distributed VCS, handling ordered lists during the transformation and providing more detailed feedbacks to the users, (ii) investigate the possibilities of building correspondence relations between the original model and filtered copy of it dedicated to a certain collaborator, and (iii) realize our collaboration scheme with other frameworks (e.g. Git, GenMyModel) and with support for continuous integration and review / change request management systems.

Conflict Reduction and Handling We plan to improve our model merge technique by further search strategies to better exploit the dependencies between rules and constraints and compare it with other search-based merge techniques [Man+15]. In case of property-based locking, we plan to extend our evaluation with respect to *under-locking* and investigate the use of incremental pattern matchers to support on-line collaboration where the collaborators work with short transactions of modifications and the response time needs to be immediate.

Synchronization of View Models We would like to (i) prioritize the synthesized solutions, (ii) improve the calculation of the source model by calling multiple solvers to minimize the size of the solution or (iii) to use SAT/SMT solvers as replacement of the Alloy Analyzer used currently in our approach.

8 Conclusion and Practical Benefits

Current dissertation is focused on developing (i) a modeling language to capture high-level access control policies (ii) a general secure collaboration scheme that guarantees that high-level access control policies are respected during collaboration and it can be integrated into existing

version control systems (e.g. SVN) to support offline scenario; (iii) automated merging and fine-grained locking to enhance the efficiency of conflict resolution and prevention upon concurrent modification of the models; (iv) derivation and incremental maintenance of view models to provide specific focus of the designers by abstracting from unnecessary details of the underlying system model.

Practical Benefits Key benefits of our secure collaborative modeling framework for MBSE include the following:

Collaboration of heterogeneous stakeholders Our framework supports collaborative modeling between engineering teams of different companies (e.g. an integrator and its subcontractors) while protecting their intellectual property with the secure storage of the gold model.

Extra layer of access control Model-level fine-grained access control using secure views injects an extra layer of protection on top of existing protection offered by the underlying repository.

Validation of access control policies Access control rules support the consistent assignment and maintenance of permissions for large models and enable the systematic validation of access control policies (e.g. to ensure export control regulations or investigate a security breach).

Compliance with SCM practices Access control policies can be defined for modern SCM practices to collaborate along multiple branches, formal change request, etc.

Smooth integration with existing tools Our framework extends existing server-side repositories while keeping client-side modeling tools intact, thus engineers may continue using existing collaborative tools.

As such, powerful existing collaboration practices used in software engineering can be complemented when collaborating over models.

Acknowledgement

I would like to thank my advisor, Daniel Varro for his guidance during my research. I would also like to express my gratitude to Istvan Rath, Gabor Bergmann, Oszkar Semerath and Akos Horvath as well as Marsha Chechik, Fabiano Dalpiaz, Jennifer Horkoff and Rick Salay along with numerous colleagues and co-authors for sharing their ideas.

I would like to express my gratitude for the support of the MTA-BME Lendulet Cyber-Physical Systems Reasearch Group project. This research was partially supported by the EU projects MONDO (ICT-611125) and CONCERTO (ART-2012-333053), the Hungarian CERTIMOT (ERC_HU-09-1-2010-0003) project and a collaborative project with Embraer called TRANS-IMA.

Publication List

Number of publications:	20
Number of peer-reviewed journal papers (written in English):	2
Number of articles in journals indexed by WoS or Scopus:	2
Number of publications (in English) with at least 50% contribution of the author:	3
Number of peer-reviewed publications:	20
Number of independent citations:	68

Publications Linked to the Theses

	Journal papers	International conference and workshop papers
Thesis 1	[*j*1],[*j*2]	[*c*6],[*c*7],[*c*3]
Thesis 2	[*j*1],[*j*2]	[*c*14],[*c*12],[*c*6],[*c*9],[*c*7],[*c*10]
Thesis 3	[*j*1]	[*c*5],[*c*13],[*c*16],[*c*17],[*c*6],[*c*9],[*c*7],[*c*4]
Thesis 4	—	[*c*15],[*c*11],[*e*19],[*c*18],[*c*7]

Journal Papers

[*j*1] Csaba Debreceni, Gábor Bergmann, István Ráth, and Dániel Varró. Secure views for collaborative modeling. *IEEE Software*, 2018. DOI: 10.1109/MS.2018.290101728.

[*j*2] Csaba Debreceni, Gábor Bergmann, István Ráth, and Dániel Varró. Enforcing fine-grained access control for secure collaborative modeling using bidirectional transformations. *Software and System Modeling, MODELS 2016 Special Section*, 2017. DOI: 10.1007/s10270-017-0631-8.

International Conference and Workshop Papers

[*c*3] Ábel Hegedüs, Gábor Bergmann, Csaba Debreceni, Ákos Horváth, Péter Lunk, Ákos Menyhért, István Papp, Dániel Varró, Tomas Vileiniskis, and István Ráth. IncQuery Server for Teamwork Cloud: scalable query evaluation over collaborative model repositories. In: *Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS 2018, Copenhagen, Denmark, October 14-19, 2018*, pp. 27–31. 2018. DOI: 10.1145/3270112.3270125.

[*c*4] Ankica Barisic, Csaba Debreceni, Dániel Varró, Vasco Amaral, and Miguel Goulão. Evaluating the efficiency of using a search-based automated model merge technique. In: *2018 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC 2018, Lisbon, Portugal, October 1-4, 2018*, pp. 193–197. 2018. DOI: 10.1109/VLHCC.2018.8506512.

[*c*5] Csaba Debreceni, Gábor Bergmann, István Ráth, and Dániel Varró. Property-based locking in collaborative modeling. In: *20th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2017, Austin, TX, USA, September 17-22, 2017*, pp. 199–209. 2017. URL: https://doi.org/10.1109/MODELS.2017.33.

[*c*6] Csaba Debreceni, Gábor Bergmann, Márton Búr, István Ráth, and Dániel Varró. The MONDO collaboration framework: secure collaborative modeling over existing version control systems. In: *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2017, Paderborn, Germany, September 4-8, 2017*, pp. 984–988. 2017. DOI: 10.1145/3106237.3122829.

[*c*7] Csaba Debreceni. Advanced techniques and tools for secure collaborative modeling. In: *Proceedings of MODELS 2017 Satellite Event: Workshops (ModComp, ME, EXE, COMMitMDE, MRT, MULTI, GEMOC, MoDeVVa, MDETools, FlexMDE, MDEbug), Posters, Doctoral Symposium, Educator Symposium, ACM Student Research Competition, and Tools and Demonstrations co-located with ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS 2017), Austin, TX, USA, September, 17, 2017.* Pp. 549–554. 2017. URL: http://ceur-ws.org/Vol-2019/acm_src_1.pdf.

[*c*8] Gábor Bergmann, Csaba Debreceni, István Ráth, and Dániel Varró. Towards efficient evaluation of rule-based permissions for fine-grained access control in collaborative modeling. In: *Proceedings of MODELS 2017 Satellite Event: Workshops (ModComp, ME, EXE, COMMitMDE, MRT, MULTI, GEMOC, MoDeVVa, MDETools, FlexMDE, MDEbug), Posters, Doc-*

toral Symposium, Educator Symposium, ACM Student Research Competition, and Tools and Demonstrations co-located with ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS 2017), Austin, TX, USA, September, 17, 2017. Pp. 135–144. 2017. URL: http://ceur-ws.org/Vol-2019/commitmde_2.pdf.

[*c*9] Abel Gómez, Xabier Mendialdua, Gábor Bergmann, Jordi Cabot, Csaba Debreceni, Antonio Garmendia, Dimitrios S. Kolovos, Juan de Lara, and Salvador Trujillo. On the opportunities of scalable modeling technologies: an experience report on wind turbines control applications development. In: *Modelling Foundations and Applications - 13th European Conference, ECMFA 2017, Held as Part of STAF 2017, Marburg, Germany, July 19-20, 2017, Proceedings*, pp. 300–315. 2017. DOI: 10.1007/978-3-319-61482-3_18.

[*c*10] Csaba Debreceni. Approaches to identify object correspondences between source models and their view models. In: *Proceedings of the 24th PHD Mini-Symposium (MINISY@DMIS), Budapest, Hungary, January 30-31, 2017*, pp. 14–17. 2017.

[*c*11] Oszkár Semeráth, Csaba Debreceni, Ákos Horváth, and Dániel Varró. Change propagation of view models by logic synthesis using SAT solvers. In: *Proceedings of the 5th International Workshop on Bidirectional Transformations, Bx 2016, co-located with The European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 8, 2016.* Pp. 40–44. 2016. URL: http://ceur-ws.org/Vol-1571/paper_6.pdf.

[*c*12] Csaba Debreceni, Gábor Bergmann, István Ráth, and Dániel Varró. Deriving effective permissions for modeling artifacts from fine-grained access control rules. In: *Proceedings of the 1st International Workshop on Collaborative Modelling in MDE (COMMitMDE 2016) co-located with ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2016), St. Malo, France, October 4, 2016.* Pp. 17–26. 2016. URL: http://ceur-ws.org/Vol-1717/paper6.pdf.

[*c*13] Csaba Debreceni, István Ráth, Dániel Varró, Xabier De Carlos, Xabier Mendialdua, and Salvador Trujillo. Automated model merge by design space exploration. In: *Fundamental Approaches to Software Engineering - 19th International Conference, FASE 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, pp. 104–121. 2016. DOI: 10.1007/978-3-662-49665-7_7.

[*c*14] Gábor Bergmann, Csaba Debreceni, István Ráth, and Dániel Varró. Query-based access control for secure collaborative modeling using bidirectional transformations. In: *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, Saint-Malo, France, October 2-7, 2016*, pp. 351–361. 2016. ACM Distinguished Paper Award. URL: http://dl.acm.org/citation.cfm?id=2976793.

[*c*15] Oszkár Semeráth, Csaba Debreceni, Ákos Horváth, and Dániel Varró. Incremental backward change propagation of view models by logic solvers. In: *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, Saint-Malo, France, October 2-7, 2016*, pp. 306–316. 2016. URL: http://dl.acm.org/citation.cfm?id=2976788.

[*c*16] Marsha Chechik, Fabiano Dalpiaz, Csaba Debreceni, Jennifer Horkoff, István Ráth, Rick Salay, and Dániel Varró. Property-based methods for collaborative model development. In: *Joint Proceedings of the 3rd International Workshop on the Globalization Of Modeling Languages and the 9th International Workshop on Multi-Paradigm Modeling co-located with ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, GEMOC+MPM@MoDELS 2015, Ottawa, Canada, September 28, 2015.* Pp. 1–7. 2015. URL: http://ceur-ws.org/Vol-1511/paper-01.pdf.

[c17] Hani Abdeen, Dániel Varró, Houari A. Sahraoui, András Szabolcs Nagy, Csaba Debreceni, Ábel Hegedüs, and Ákos Horváth. Multi-objective optimization in rule-based design space exploration. In: *ACM/IEEE International Conference on Automated Software Engineering, ASE '14, Vasteras, Sweden - September 15 - 19, 2014*, pp. 289–300. 2014. DOI: 10.1145/2642937.2643005.

[c18] Csaba Debreceni, Ákos Horváth, Ábel Hegedüs, Zoltán Ujhelyi, István Ráth, and Dániel Varró. Query-driven incremental synchronization of view models. In: *Proceedings of the 2nd Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling, VAO@STAF 2014, York, United Kingdom, July 22, 2014*, pp. 31–38. 2014. DOI: 10.1145/2631675.2631677.

Local Conference and Workshop Papers

[e19] Csaba Debreceni. Automated abstraction in model-driven engineering. In: *Mesterpróba 2014 Tudományos konferencia végzös MSc és elsöéves PhD hallgatóknak Távközlés és infokommunikáció témakörében, Budapest, Hungary, May 29, 2014*, pp. 67–70. 2014.

Additional Publications (Not Linked to Theses)

International Conference and Workshop Papers

[c20] Gábor Szárnyas, Oszkár Semeráth, Benedek Izsó, Csaba Debreceni, Ábel Hegedüs, Zoltán Ujhelyi, and Gábor Bergmann. Movie Database Case: An EMF-IncQuery Solution. In: *Proceedings of the 7th Transformation Tool Contest part of the Software Technologies: Applications and Foundations (STAF 2014) federation of conferences, York, United Kingdom, July 25, 2014.* Pp. 103–115. 2014. URL: http://ceur-ws.org/Vol-1305/paper14.pdf.

References

[Alt+08] Kerstin Altmanninger, Gerti Kappel, Angelika Kusel, Werner Retschitzegger, Martina Seidl, Wieland Schwinger, and Manuel Wimmer. Amor–towards adaptable model versioning. In: *1st International Workshop on Model Co-Evolution and Consistency Management, in conjunction with MODELS*, vol. 8, pp. 4–50. 2008.

[Bag+14] Alessandra Bagnato, Etienne Brosse, Andrey Sadovykh, Pedro Maló, Salvador Trujillo, Xabier Mendialdua, and Xabier De Carlos. Flexible and scalable modelling in the mondo project: industrial case studies. In: *XM@ MoDELS*, pp. 42–51. 2014.

[Bro+09] Petra Brosch, Martina Seidl, Konrad Wieland, and Manuel Wimmer. We can work it out: collaborative conflict resolution in model versioning. In: *Proceedings of the Eleventh European Conference on Computer Supported Cooperative Work, ECSCW 2009, 7-11 September 2009, Vienna, Austria*, pp. 207–214. 2009. URL: http://www.ecscw.org/2009/15-BroschEtAl.pdf.

[Bru+15] Hugo Brunelière, Jokin Garcıa Perez, Manuel Wimmer, and Jordi Cabot. EMF views: A view mechanism for integrating heterogeneous models. In: *Conceptual Modeling - 34th International Conference, ER 2015, Stockholm, Sweden, October 19-22, 2015, Proceedings*, pp. 317–325. 2015. DOI: 10.1007/978-3-319-25264-3_23.

[CDO] The Eclipse Foundation. CDO. http://eclipse.org/cdo.

[CJC11] Cauê Clasen, Frédéric Jouault, and Jordi Cabot. Virtualemf: A model virtualization tool. In: *Advances in Conceptual Modeling. Recent Developments and New Directions - ER 2011 Workshops FP-UML, MoRE-BI, Onto-CoM, SeCoGIS, Variability@ER, WISM, Brussels, Belgium, October 31 - November 3, 2011. Proceedings*, pp. 332–335. 2011. DOI: 10.1007/978-3-642-24574-9_43.

[Conc] CONCERTO ARTEMIS project. http://concerto-project.org/.

[DRE14] Hoa Khanh Dam, Alexander Reder, and Alexander Egyed. Inconsistency resolution in merging versions of architectural models. In: *2014 IEEE/IFIP Conference on Software Architecture, WICSA 2014, Sydney, Australia, April 7-11, 2014*, pp. 153–162. 2014. DOI: 10.1109/WICSA.2014.31.

[EMF-Comp] The Eclipse Foundation. EMF Compare. http://eclipse.org/emf/compare/.

[EMF-Diff] The Eclipse Foundation. EMF Diff/Merge. http://eclipse.org/diffmerge/.

[EMFStore] The Eclipse Foundation. EMFStore. http://eclipse.org/emfstore.

[GDocs] Google. Google Sheets. docs.google.com.

[Gen] Axellience. GenMyModel. http://genmymodel.com.

[Gho+15] Hamid Gholizadeh, Zinovy Diskin, Sahar Kokaly, and Tom Maibaum. Analysis of source-to-target model transformations in quest. In: *Proceedings of the 4th Workshop on the Analysis of Model Transformations co-located with (MODELS 2015, Ottawa, Canada*, pp. 46–55. 2015. URL: http://ceur-ws.org/Vol-1500/paper6.pdf.

[Heg14] Ábel Hegedüs. Back-annotation of Execution Sequences by Advanced Search and Traceability Techniques. PhD thesis. Budapest University of Technology and Economics, 2014.

[Hor13] Ákos Horváth. Search-Based Techniques in Model-Driven Engineering. PhD thesis. Budapest University of Technology and Economics, 2013.

[JFace] The Eclipse Foundation. JFace. https://wiki.eclipse.org/JFace.

[Kol09] Dimitrios S. Kolovos. Establishing correspondences between models with the epsilon comparison language. In: *Model Driven Architecture - Foundations and Applications, 5th European Conference, ECMDA-FA 2009, Enschede, The Netherlands, June 23-26, 2009. Proceedings*, pp. 146–157. 2009. DOI: 10.1007/978-3-642-02674-4_11.

[Kra+06] Gerhard Kramler, Gerti Kappel, Thomas Reiter, Elisabeth Kapsammer, Werner Retschitzegger, and Wieland Schwinger. Towards a semantic infrastructure supporting model-based tool integration. In: *Proceedings of the 2006 international workshop on Global integrated model management*, pp. 43–46. 2006.

[Man+15] Usman Mansoor, Marouane Kessentini, Philip Langer, Manuel Wimmer, Slim Bechikh, and Kalyanmoy Deb. MOMM: multi-objective model merging. *Journal of Systems and Software* 103, 2015, pp. 423–439. DOI: 10.1016/j.jss.2014.11.043.

[Mar+14] Miklós Maróti, Tamás Kecskés, Róbert Kereskényi, Brian Broll, Péter Völgyesi, László Jurácz, Tihamer Levendovszky, and Ákos Lédeczi. Next generation (meta)modeling: web- and cloud-based collaborative tool infrastructure. In: *Proceedings of the 8th Workshop on Multi-Paradigm Modeling co-located with the 17th International Conference on Model Driven Engineering Languages and Systems, MPM@MODELS 2014, Valencia, Spain, September 30, 2014.* Pp. 41–60. 2014. URL: http://ceur-ws.org/Vol-1237/paper5.pdf.

[Obeo] Obeo. Obeo Designer. https://www.obeodesigner.com.

[QVT] OMG. MOF 2.0 Query/View/Transformation specification (QVT), version 1.1. http://www.omg.org/spec/QVT/1.2/.

[RC13] Julia Rubin and Marsha Chechik. N-way model merging. In: *Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE'13, Saint Petersburg, Russian Federation, August 18-26, 2013*, pp. 301–311. 2013. DOI: 10.1145/2491411.2491446.

[Roc+15] J. Di Rocco, D. Di Ruscio, L. Iovino, and A. Pierantonio. Collaborative repositories in model-driven engineering [software technology]. *IEEE Software* 32(3), 2015, pp. 28–34. DOI: 10.1109/MS.2015.61.

[Sch94] Andy Schürr. Specification of graph translators with triple graph grammars. In: *Graph-Theoretic Concepts in Computer Science, 20th International Workshop, WG '94, Herrsching, Germany, June 16-18, 1994, Proceedings*, pp. 151–163. 1994. DOI: 10.1007/3-540-59071-4_45.

[Sem19] Oszkár Semeráth. Formal Validation and Model Generation for Domain-Specific Languages by Logic Solvers. PhD thesis. Budapest University of Technology and Economics, 2019.

[Son+11] Hui Song, Gang Huang, Franck Chauvel, Wei Zhang, Yanchun Sun, Weizhong Shao, and Hong Mei. Instant and incremental QVT transformation for runtime models. In: *Model Driven Engineering Languages and Systems, 14th International Conference, MODELS 2011, Wellington, New Zealand, October 16-21, 2011. Proceedings*, pp. 273–288. 2011. DOI: 10.1007/978-3-642-24485-8_20.

[SVN] Apache. Subversion. https://subversion.apache.org/.

[Tol07] Juha-Pekka Tolvanen. MetaEdit+: domain-specific modeling and product generation environment. In: *Software Product Lines, 11th Int. Conf. SPLC 2007, Kyoto, Japan*, pp. 145–146. 2007.

[Tol16] Juha-Pekka Tolvanen. Metaedit+ for collaborative language engineering and language use (tool demo). In: *Proceedings of the 2016 ACM SIGPLAN International Conference on Software Language Engineering, Amsterdam, The Netherlands, October 31 - November 1, 2016*, pp. 41–45. 2016. URL: http://dl.acm.org/citation.cfm?id=2997379.

[Ujh16] Zoltán Ujhelyi. Program Analysis Techniques for Model Queries and Transformations. PhD thesis. Budapest University of Technology and Economics, 2016.

[VVi] VIATRA. VIATRA Viewers Documentation. https://www.eclipse.org/viatra/documentation/addons.html.

[Wes14] Bernhard Westfechtel. Merging of EMF models - formal foundations. *Software and System Modeling* 13(2), 2014, pp. 757–788.

[WHR14] Jon Whittle, John Edward Hutchinson, and Mark Rouncefield. The state of practice in model-driven engineering. *IEEE Software* 31(3), 2014, pp. 79–85. DOI: 10.1109/MS.2013.65.

[Xio+07] Yingfei Xiong, Dongxi Liu, Zhenjiang Hu, Haiyan Zhao, Masato Takeichi, and Hong Mei. Towards automatic model synchronization from model transformations. In: *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering*, pp. 164–173. 2007.

[YFiles] yWorks. yFiles. https://www.yworks.com/products/yfiles.

[Zest] The Eclipse Foundation. Zest. https://www.eclipse.org/gef/zest/.