Semi-automatic Matching of Heterogeneous Model-based Specifications

Konrad Voigt
SAP Research CEC Dresden
Chemnitzer Str. 48
01187 Dresden, Germany
konrad.voigt@sap.com *

Abstract:
IT-systems are often described by a variety of specifications such as UML, Java, BPMN, WSDL, etc. These heterogeneous specifications constitute different views on the same system, resulting in the challenge of matching. That is in context of Model Driven Engineering the discovery of semantic correspondences between model elements, which can be used for tasks such as transformation or trace link generation. Although support by semi-automatic matching has been proposed, current approaches show deficits. They leave room for improvement in matching quality, do not appropriately address scalability, and miss a thorough evaluation. They were proposed to specifically target differencing and versioning in contrast to matching.

We tackle these issues by proposing a configurable combination of matchers that considers both: meta-models and models. We propose to adopt established schema matching techniques and to utilize information gained from meta-model matching for the task of model matching. Additionally, the graph qualities planarity and reducibility are used to apply graph isomorphism and clustering algorithms for enhanced structural matching.

Finally, we build upon a generic model-based infrastructure (EMF) allowing for an easy integration of heterogeneous specifications and the realization of scenarios from the area of service engineering for our proposed evaluation.

1 Introduction

IT-systems have to deal with a diversity of specifications, which are often heterogeneous, thus naturally leading to redundancy and interdependencies. Maintaining and defining relations and dependencies between specifications is a nontrivial task and has been a known challenge since the early 1970s [ARNRSG06]. The inception of model-driven engineering (MDE) entails the use of models as specifications for different domains, such as the modelling of business processes, system requirements, architecture, and tests. Although

*This work is done in context of a doctoral work at the chair of Software Technology of Prof. Uwe Aßmann at the Technical University of Dresden. The work was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ07012". The author takes responsibility for the contents.

MDE advocates a support of linking and round trip engineering by model transformations, reality still fails to deliver on this promises [Sel03].

This is tackled by semi-automatic calculation of links based on similarity as in model differencing and model matching. Semi-automatic model differencing aims on calculating the difference between models, e. g. which element has been removed. In contrast, meta-model matching calculates similarities between elements, i. e. it targets link creation. Matching and differencing can be applied on meta-models as well as models. Recently, several approaches arose to support model differencing with focus on model versioning, e. .g targeting the meta-model [FHLN08] level. Thereby, EMF Compare [BP08], SiDiff [SG08] and DSM-Diff [LGJ07] are the ones supporting differencing at meta-model and model level. However, due to their aim of differencing rather than matching, none of the approaches tackles the *similarity* calculation for trace link generation, which has been identified as a major challenge in model traceability [ARNRSG06]. Automatic trace link generation is supported by most model transformation engines, however nowadays many transformations are implemented manually (e. g. in Java), thus lacking an automatic trace link generation.

We propose to tackle these problems of trace link generation by presenting MatchBox; a configurable matching system capable of meta-model matching along with an infrastructure to manage and create mappings. Thereby, we adapt and extend schema matching techniques [VIR10] for model matching. To present these contributions, we first introduce the challenges of model matching by means of trace link creation in Section 2. Afterwards, we describe our contributions to address these challenges in detail by explaining MatchBox, its components, process, and current status. Subsequently, we present our evaluation and give an overview of related work, to finally conclude in Section 5.

2 Challenges of Matching for Trace Link Creation

The illustrative scenario for trace link generation in a MDE process is depicted in Figure 1 on the left. Thereby, a model is transformed into another, preferably this is done using a transformation engine. If so, an approach as in [GV09] will create trace links connecting the original and resulting model. Then, the resulting model is adjusted manually, e. g. to serve specific user needs, which potentially leads to unlinked elements. In the worst-case scenario a transformation engine creating the links is missing. The arising challenge is to create the missing trace links. The right side depicts our approach on applying our matching system which creates trace links based on the calculated similarities of elements on meta-model and model level. The projection of the meta-model elements on model level is illustrated, denoting the reuse of meta-model mappings for search space reduction on model level.

In the context of trace link creation, the participating models (esp. on instance level) easily constitute 5000 or more elements[1], which need to be evaluated for trace links. Considering a matching approach, 5000 x 5000 elements have to be compared resulting in 25 million

[1] For instance 400 Line Java Class represented in JaMopp (http://jamopp.inf.tu-dresden.de/)

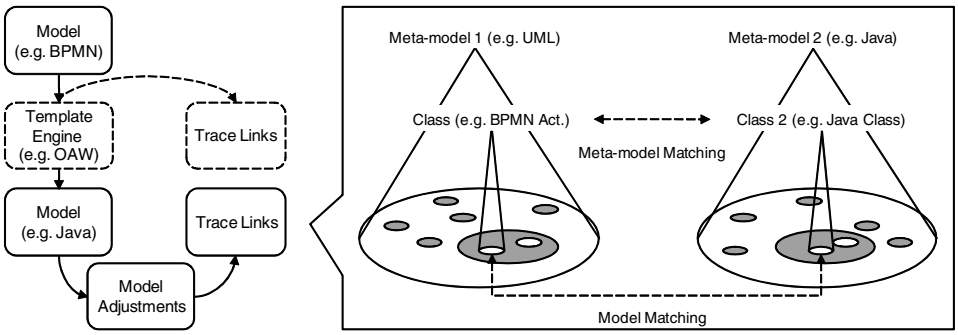

Figure 1: Example trace link scenario and relation to model and meta-model matching

calculations. Therefore, applying matching techniques for trace link creation requires a high *scalability*. In this context scalability means that the system is capable of handling instances of growing dimensions and that they are processed in a reasonable time.

Another challenge covers the *quality* of trace links obtained, i. e. the amount of correctly identified trace links and the ratio of all identified links to the correct set of links. To support common traceability scenarios as impact, orphan or system analysis [GV09], a certain quality has to be ensured in order to provide reliable results.

Finally, a significant *evaluation* of the concepts is a major challenge. In order to demonstrate the capabilities and limitations and evaluate the framework an evaluation based on a broad range of examples is needed providing a variety in structure and complexity as well as heterogeneity in specifications (models).

3 The MatchBox Approach

We propose MatchBox a system that provides an infrastructure for meta model and model matching which addresses the challenges of trace link generation. Thereby, we make the following contributions (1) A generic architecture for a model matching infrastructure for trace link generation, (2) a graph-based clustering approach to cope with the dimensions of model matching, (3) planar graph-based matching techniques for improved matching results, (4) a utilization of meta-model matching results for model matching, and (5) a large-scale evaluation concept for model matching approaches.

Figure 2 depicts an overview of MatchBox along with its four processing steps, each supported by a component. The process starts with the adaptation component (1) which is responsible for an import of meta-models and models into our internal graph-based representation. The clustering component (2) separates the input graph into different parts, thus reducing the dimension of the matching problem. The matching core (3) is an exchangeable component. It follows an interface that requests a similarity value between two model elements which is the task of matching. The matching techniques applied range from name-bases similarity, to data types, parent-child relations, and graph-based

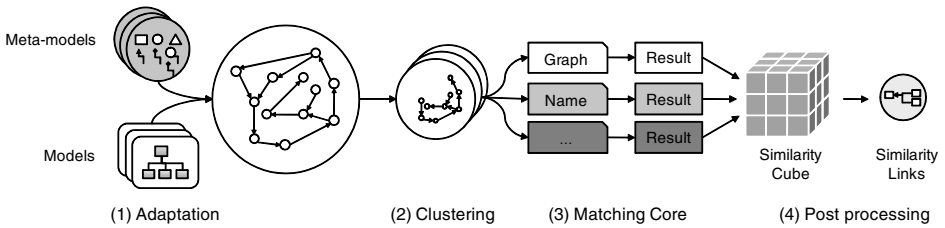

Figure 2: Conceptual overview on the process, internal model and matching in MatchBox

ones. For further details please refer to [VIR10]. Finally the post-processing component (4) uses created mappings (links) to remove contradicting or duplicate mappings. Put simply, MatchBox takes two models as input and produces mappings as output. In the following, we will argue how we tackle the challenges given in Section 2.

Scalability The challenge of scalability is tackled by following a divide-and-conquer approach, thus reducing the actual problem dimension. This is achieved by clustering the input graph into smaller sub graphs (clusters) which are matched with each other. These clusters will be determined by results given from meta-model matching, thus arranging elements by their corresponding mapped type and strongly connected components. Applying isomorphism algorithms on graphs in general results in NP-complete problems. However, algorithms based on certain graph characteristics as planarity or reducibility are promising to determine matchings in reasonable time, thus increasing scalability.

Matching Quality The matching on both layers makes use of common schema matching techniques such as name edit distance and parent-child similarity propagation as described in [VIR10] and additional graph matchers using structural information and graph characteristics. Again isomorphism or graph edit distance algorithms for general graphs are NP-complete, but, for instance planarity provides a linear algorithm for the calculation of an error-tolerant approximate graph edit distance on attributed graphs [NB04]. In first experiments on meta-model level, we achieved promising results [2]. Our first prototype of our meta-model matching system MatchBox is based on EMF which provides a common space for integration of models. Therefore, we do not need to struggle with the extraction of information and integration.

Evaluation of MatchBox Our evaluation has to consider meta-model matching and model matching. The *matching quality* is commonly determined by using the statistical measures precision, recall and F-Measure from the field of information retrieval [Rij79]. These measures are calculated by comparing the result obtained by a matching system to a so-called gold-standard which is the perfect match. For meta-model matching, we propose to use model transformations as gold-standards which enable an automatic extraction

[2] An improvement up to 25% compared to traditional matching techniques (name, name path and children [VIR10])

of gold-standards and an extensible evaluation base. One example of a source for gold-standards is the ATL-Zoo[3] consisting of more than hundred model transformations. For our evaluation of model matching we will make use of Service Engineering [CVW08] and several service descriptions and models in the context of the TEXO project each being made of 8 models with 10 up to 15,000 elements. We will further investigate the mappings contained in the SAP Enterprise Service Repository (ESR) [4] as a base for our evaluation, which will grant access to at least 30 real world gold standards for services interface mappings. Regarding *scalability* we will apply our approach to a variety of synthetic and real-world models with differing size and complexity as given in the previous paragraph. Thereby, we will take measurements such as runtime and memory consumption. Finally, we propose to compare MatchBox with other matching approaches.

4 Related Work

Model transformation engines in general allow for trace link generation by default. However, we especially tackle the problem of having no transformation engine present. The problem of model element similarity calculation is dealt with by semi-automatic matching. However, current systems lack quality in terms of matching results obtained, scalability, and a comprehensive evaluation. There are three systems supporting layer independent matching, i. e. semi-automatic matching of meta-model and models.

EMF Compare [BP08] is the most prominent one possessing a broad user base since it is an EMF-related Eclipse project. We applied experiments with more than 120 meta-models[5] comparing it with MatchBox and noted a quality improvement when using MatchBox in terms of precision and recall regarding the matches by factor 3-4. This is due to the fact that EMF Compare uses a simple algorithm which computes label and string edit distances. Furthermore, EMF Compare present a weak concepts for scaleability by only considering tree-based neighbours. DSMdiff [LGJ07] uses type information of meta-models to encode characteristics in strings being compared. The structural information used is as simple as the number of children or references. This leaves room for improvement in matching quality. Again they do not present a concept to cope with the dimensions of model matching and thus scalability. SiDiff [SG08] presents an algorithm traversing a tree bottom up and top down similar to our parent-child similarity propagation. However, in contrast to them we consider the graph structure as a whole. Finally, all approaches consider model differencing, thus adding potentially misleading information by match result interpretation. In contrast, we propose matching only. All of the matching approaches evaluate their systems by a limited number of simple artificial examples e. g. two similar meta-models (e. g. UML and MiniJava [FHLN08]) or two trivial models (UML-Class diagrams [BP08, LGJ07]) which indeed yield good results due to many identical elements. In contrast, we target a comprehensive real-world evaluation.

[3] http://www.eclipse.org/atl/zoo
[4] http://esworkplace.sap.com
[5] ATL-Zoo, ModelCVS and Ontology Alignment Contest

5 Conclusion

We have presented MatchBox, a system for layer independent matching allowing for metamodel and model matching based on a combining approach making use of graph characteristics for matching. We have given an overview of the current state of the art in matching and have identified challenges in link creation that we address with MatchBox. Thereby, we cover the complete matching process, by contributing improved matching techniques, using clustering for increased scalability and a real-world evaluation based on service descriptions and implementations. The work on the graph characteristics and use of metamodel matching information is currently ongoing. In parallel, we implement and extend the architecture and infrastructure step-wise to a complete version of MatchBox.

References

[ARNRSG06] N. Aizenbud-Reshef, B.T. Nolan, J. Rubin, and Y. Shaham-Gafni. Model traceability. *IBM Systems J.*, 45(3):515–526, 2006.

[BP08] C. Brun and A. Pierantonio. Model differences in the eclipse modeling framework. *Upgrade, Special Issue on Model-Driven Software Development IX*, 2008.

[CVW08] Jorge Cardoso, Konrad Voigt, and Matthias Winkler. Service Engineering for The Internet of Services. In *Enterprise Information Systems X*. Springer, 2008.

[FHLN08] Jean-Rémy Falleri, Marianne Huchard, Mathieu Lafourcade, and Clémentine Nebut. Metamodel Matching for Automatic Model Transformation Generation. In *Proc. of MoDELS '08*, pages 326–340, 2008.

[GV09] Birgit Grammel and Konrad Voigt. Foundations for a Generic Traceability Framework in Model-Driven Software Engineering. In *Proc. of the ECMDA Traceability Workshop'09*, 2009.

[LGJ07] Y. Lin, J. Gray, and F. Jouault. DSMDiff: a differentiation tool for domain-specific models. *European Journal of Information Systems*, 16(4):349–361, 2007.

[NB04] M. Neuhaus and H. Bunke. An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification. *LNCS*, pages 180–189, 2004.

[Rij79] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 1979.

[Sel03] B. Selic. The pragmatics of model-driven development. *IEEE software*, 20(5):19–25, 2003.

[SG08] Maik Schmidt and Tilman Gloetzner. Constructing difference tools for models using the SiDiff framework. In *ICSE Companion '08*, pages 947–948. ACM, 2008.

[VIR10] Konrad Voigt, Petko Ivanov, and Andreas Rummler. MatchBox: Combined Metamodel Matching for Semi-automatic Mapping Generation. In *Proc. of ACM SAC '10*, 2010.