A Generic Multi-purpose Conceptual Model Analysis Approach – Conceptual Specification and Application

Sebastian Herwig, Łukasz Lis, Matthias Steinhorst, Jörg Becker, Patrick Delfmann

University of Münster
European Research Center for Information Systems (ERCIS)
Leonardo-Campus 3
48149, Münster
Germany
{herwig | lis | steinhorst | becker | delfmann}@ercis.uni-muenster.de

Abstract: In this paper, we introduce a model analysis approach suitable for multiple analysis purposes. The approach is based on generic graph pattern matching and semantic standardization of model elements. Due to its generic nature, it is applicable to conceptual models of any graph-based modelling technique. Through semantic standardization, the approach relies on unambiguous model contents. Therefore, its focus is on analyses with regards to the structure and contents of conceptual models.

1 Introduction

The analysis of conceptual models addresses different goals, e. g., searching for corresponding model sections for model integration in distributed modelling projects or evaluating process compliance, amongst others. Manual model analysis can be costly, as the number of models to be analysed may rank in the thousands [YDG10]. We therefore argue that a semi-automated model analysis approach being suitable for different modelling languages and for different application scenarios is highly beneficial. Such an approach has to consider two basic aspects: First, it should be able to recognize structures occurring within a model. Second, the approach should recognize the labels of the model elements. It is crucial that the model elements' contents are unambiguous, thus the approach should incorporate a mechanism that relies on semantic standardization.

In this contribution, we present a semi-automated model analysis approach, which makes use of semantic standardization to assure comparability as a precondition for model analysis and allows for flexible pattern specification and matching. Our research follows the design science paradigm outlined by [He04]. This integrated approach is based on our previous research on naming conventions [DHL09] and generic pattern matching [De10]. The remainder of this contribution proceeds as follows: In Section 2, we present a literature overview of existing model analysis approaches. Section 3 introduces the conceptual specification of the approach. Section 4 provides an application example. We close with an outlook to future research in Section 5.

201

2 Related Work

Different analysis approaches proposed in the literature can be divided according to their primary goal into *quality evaluation*, *exploration*, and *comparison* approaches. Notable work concerned with evaluating the syntactical as well as semantic quality of conceptual models has, for example, been proposed by [Fr09], [LSM09], or [Go08]. In terms of model exploration, various analysis approaches have been proposed by [Az09] [Es09], or [TIR07]. To compare two or more models to one another, algorithms have been developed by [Di11], [YDG10], or [DDM08], amongst others. These approaches deal with particular analysis task and are therefore only suitable in their respective domains. We argue that it would be beneficial to have a generic analysis approach being suitable for any modelling language and in any application domain. The following section introduces such an approach. It is based on our previous work on enforcing naming conventions [DHL09] and identifying arbitrary structures in conceptual models [De10].

3 Conceptual Specification

The overall procedure of our analysis approach is subdivided into two main steps as depicted in Figure 1 (black-shaded elements are adapted from the pattern matching approach, grey-shaded elements are adapted from the semantic standardization approach, and non-shaded elements are new). First, the analysis is defined (cf. section Analysis Definition in Figure 1). Second, the analysis is applied to a set of models in order to generate a report as a result (cf. section Report Generation in Figure 1). An *Analysis* is composed of one or more *Sub-analyses*, each of them having a particular *Search Criterion*. The search criterion describes the properties of the expected analysis results (e.g., "find all receipt structures containing the term "invoice). Furthermore, it has to be defined whether the scope of an analysis is to explore a single set of models or to perform a comparison of two model sets (cf. attribute *scope*). To assure that all sub-analyses within an analysis can be applied to the same model base, an analysis can be composed only of those sub-analyses which have the same scope.

Each sub-analysis is based on a *Search Criterion* specifying the fact to be searched for. Any search criterion consists of either a single *Atomic Search Criterion* or a *Composed Search Criterion*. Atomic search criteria are further specialized as:

- *Pattern Equivalence Class (PEC)* to search for different structural patterns regarded as equivalent (e.g., two different data structures being recognized as similar)
- *Structural Pattern* to search for a specific model structure (e.g., a pattern representing activities in process models that are related to an application system)
- *Element Types* (e.g., all "application systems")
- *Phrase Syntax* to search for model elements whose labels follow a certain syntax (e.g., model elements whose labels follow the syntax <*verb, imperative*> <*noun, singular*> in order to name process activities)
- *Word Class* to search for all words of a certain type within model elements' labels (e.g., all nouns to identify business process objects)

202

- *Word* to search for a particular word within the label of model elements (e.g., the particular word “invoice” in order to identify all data models containing invoice data)
- *Comparison Type* to define a criterion for the comparison of two model sets. A comparison type can take the values “structural pattern”, “pattern equivalence class”, “element type”, “word class”, “word”, or “phrase syntax”. If an analysis has the scope “comparative”, a result found in the one model set is compared to all results found in the other model set. If both results are equivalent concerning the comparison type, they are returned as a result pair marked as “equivalent”. For example, finding similar structures in data models requires defining a pattern equivalence class containing these structures as structural patterns. Then, a comparative analysis can use this class as a search criterion. In addition, it uses the comparison type “pattern equivalence class” to relate every pattern match in the one model to every match in the other model, and possibly a further comparison type “word” to only return structure pairs containing the same term (e.g., “invoice”).

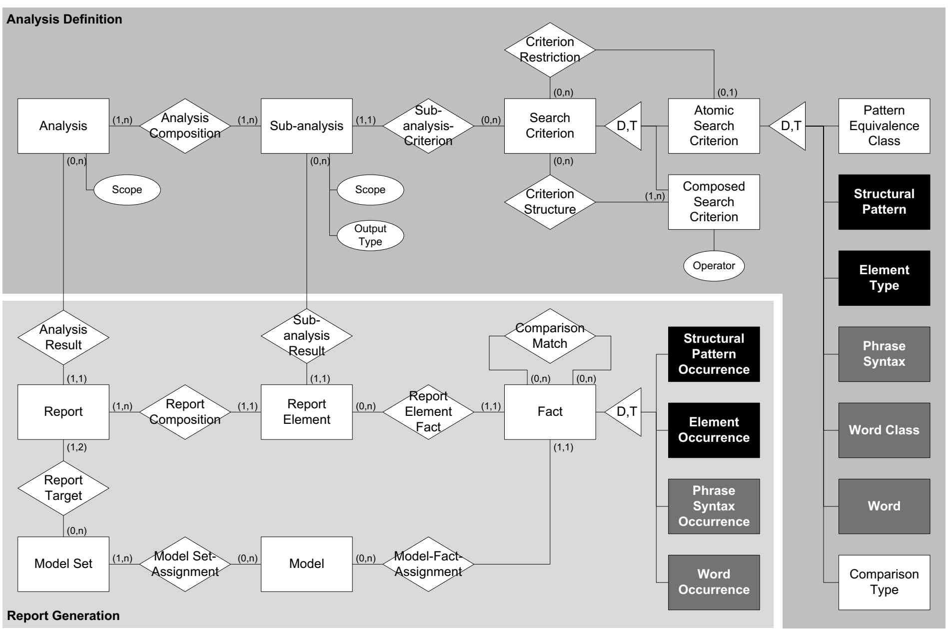

Figure 1. Conceptual Specification of the Analysis Approach

As a comparison type defines only a criterion for the comparison of two model sets, this criteria is only available for comparative sub-analysis (i.e., the attribute scope is set to “comparative”). All the other atomic criteria can be used regardless of the specified scope. A search criterion that contains more than one atomic search criterion is defined as *Composed Search Criterion*. It is composed of further complex or atomic search criteria, which are connected through a logical operator. The search criteria to be combined are specified by the *Criterion Structure*. To define the logical operator to be used for the combination of these search criteria, we introduce the attribute *Operator*, which can hold

203

one of the values “AND”, “OR”, “XOR” or “NOT”. A *Criterion Restriction* is used to refine an atomic search criterion. Here, a search criterion – either an atomic or a complex one – serves as a constraint for an atomic search criterion restricting the resulting set of model fragments. Therefore, the restricting search criterion is directly assigned to the atomic search criterion to be restricted.

Finally, an analyst has to specify the granularity of the analysis results to be displayed. Therefore, we introduce the attribute *Output Type*. For example, an analysis is defined to search for structural pattern occurrences. By defining patterns as output type, the pattern occurrences contained in the analysis' result are included in the report. By setting the value to “element”, the report is straightened to visualize occurrences of single model elements, which are contained in the returned pattern occurrences. By defining “phrase syntax” as output type only phrase syntax occurrences contained in the returned pattern occurrences are visualized, and so on. To avoid empty reports, the output type has to be defined in respect to specified search criteria.

The results of an analysis are visualized as a *Report*. Depending on the scope of an analysis, a report is targeted at either one *Model Set* for an explorative analysis or two model sets for a comparative analysis. Each model set consists of one or more *Models* to be analysed. A report is composed of one or more *Report Elements*, each of them resulting from one sub-analysis. A report element represents a single row in a report and shows the particular *Facts* returned as search results from the corresponding sub-analysis. A fact defines a particular match of the sub-analysis' search criterion, which is shown in the report. In respect to the possible output types of a sub-analysis, a fact can be a *Pattern Occurrence*, an *Element Occurrence*, a *Phrase Syntax Occurrence,* or a *Word Occurrence*. For one particular sub-analysis of the aforementioned example, all process activities that are related with a specific application system are included in the report element as particular element occurrences. Facts resulting from a comparative analysis are linked to each other as *Comparison Matches*. This way, equivalent result pairs are defined by relating a fact occurrence in the one model set to the matching fact occurrence in the other one.

4 Application

To demonstrate the feasibility of our model analysis approach, we developed a prototypical implementation combining both structural pattern search and linguistic standardization features. As an implementation basis, we used a meta-modelling tool which was available from a previous research project. Since our model analysis approach is generic in nature, it is applicable to a variety of business scenarios. To evaluate its general feasibility, however, we decided to provide an application example in an area where we believe the approach is most beneficial, namely in case of mergers and acquisitions. A major challenge in merging two or more companies is integrating the respective IT landscapes [MB07]. A first step toward IT integration in merger scenarios is identifying the different application systems that support a particular business activity in all involved companies.

204

To identify such structures, we define a simple analysis with the output type pattern. The report will show us all model structures that match the predefined search criteria. For the example of application systems supporting specific business activities we therefore defined a structural pattern called “Function->Application System” for EPCs. This pattern returns all functions that are directly related to an application system. As far as the linguistic features are concerned, we define the words “customer” and “CRM” to comprise the necessary vocabulary for this domain. Given this pattern and these terms we can construct the search criterion that allows for searching occurrences of the “Function->Application System” pattern. The search is further restricted to include only those pattern occurrences containing either the word “customer” or “CRM” at least once.

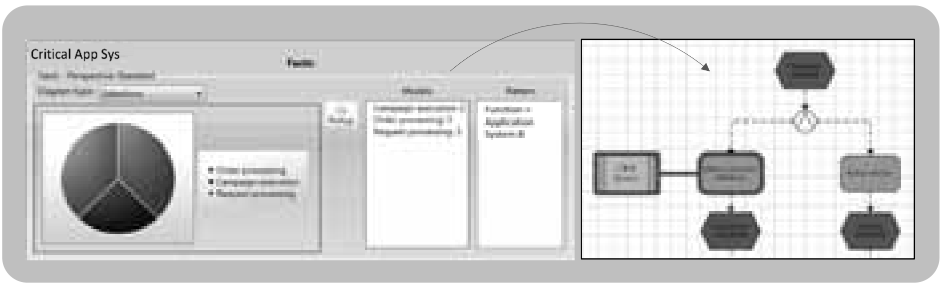

Figure 2. Exemplary Analysis

As a model basis, assume that we have two fictional companies. For each of these companies we modelled the three business processes “campaign execution”, “order processing”, and “request processing” as EPCs. Running the analysis as described above results in the report depicted in Figure 2. In this case, the business process “campaign execution” contains two occurrences of the specified search criterion. The processes “order processing” and “request processing” each contain three occurrences. Clicking on one model allows for navigating to it. Each fact is then highlighted, so the analyst can easily locate the different fact occurrences. In the example, all functions that are directly connected to an application system and contain the terms “customer” or “CRM” are highlighted.

5 Outlook

The contribution of this paper is two-fold: on the one hand, it provides a convenient tool to practitioners and professionals that can be used in a multitude of application scenarios, regardless of what modelling languages they prefer. On the other hand, we provide an innovative approach to the model analysis body of knowledge. It takes into account both the intensively discussed requirements of semantic interoperability and structural pattern matching. It is applicable, reusable and thus evaluable in multiple scientific, educational and professional scenarios. In future research, we aim at developing an automated or at least semi-automated approach to integrate a number of conceptual models to one consolidated model. Furthermore, we intend to develop meaningful metrics for the analysed models.

205

References

[Az09] Azevedo, L. G.; Santoro, F.; Baião, F.; Souza, J.; Revoredo, K.; Pereira, V.; Herlain, I.: A Method for Service Identification from Business Process Models in a SOA Approach. In (van der Aalst, W.M.P.; Mylopoulos, J.; Sadeh, N. M.; Shaw, M. J.; Szyperski, C.; Halpin, T. et al. Eds.): Enterprise, Business-Process and Information Systems Modeling, LNBIP, vol. 29. Springer, Heidelberg, 2009; pp. 99-112.

[DDM08] van Dongen, B. F.; Dijkman, R.; Mendling, J.: Measuring similarity between business process models. In (Bellahsene, Z.; Léonard, M. Eds.): Proceedings of the 20th International Conference on Advanced Information Systems Engineering, Montpellier, 2008, pp. 450-464.

[De10] Delfmann, P.; Herwig, S.; Lis, L.; Stein, A.; Tent, K.; Becker, J.: Pattern Specification and Matching in Conceptual Models. A Generic Approach Based on Set Operations. In: Enterprise Modelling and Information Systems Architecture 5 (2010) 3, pp. 24-43.

[Di11] Dijkman, R.; Dumas, M.; van Dongen, B.; Käärik, R.; Mendling, J.: Similarity of business process models: Metrics and evaluation. In: Information Systems 36 (2011) 2; pp. 498-516.

[DHL09] Delfmann, P.; Herwig, S.; Lis, L.: Unified Enterprise Knowledge Representation with Conceptual Models – Capturing Corporate Language in Naming Conventions. In: Proceeding of the 30th International Conference on Information Systems (ICIS 2009), Phoenix, 2009, pp. 13-26.

[Es09] Esswein, W.; Weller, J.; Stark, J.; Juhrisch, M.: Identifikation von Services aus Geschäftsprozessmodellen durch automatisierte Modellanalyse. In (Hansen, H. R.; Karagiannis, D.; Fill, H.-G. Eds.): Proceedings der 9. Internationalen Tagung Wirtschaftsinformatik, Vienna, 2009; pp. 513-522.

[Fr09] Friedrich, F.: Measuring semantic label quality using WordNet. In: Proceedings des 8. GI-Workshops Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK). CEUR-WS.org, 2009; pp. 7-21.

[Go08] Governatori, G.; Hoffmann, J.; Sadiq, S.; Weber, I.: Detecting Regulatory Compliance for Business Process Models through Semantic Annotations. In (van der Aalst, W.M.P.; Mylopoulos, J.; Sadeh, N. M. et al. Eds.): Business Process Management Workshops. Springer, Heidelberg, 2008; pp. 5-17.

[He04] Hevner, A. R. et. al.: Design Science in Information Systems Research. MIS Quarterly 28, 2004; pp. 75-105.

[LSM09] Leopold, H.; Smirnov, S.; Mendling, J.: On labeling quality in business process models. In: Proceedings des 8. GI-Workshop Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK). CEUR-WS.org, 2009; pp. 42-57.

[MB07] Mirklitz, T.; Buxmann, P.: IT Standardization and Integration in Mergers and Acquisitions. A Decision Model for the Selection of Application Systems. In: Proceedings of the 15th European Conference on Information Systems (ECIS 2007), St. Gallen, 2007, pp. 1041-1051.

[TIR07] Thom, L. H.; Iochpe, C.; Reichert, M.: Workflow patterns for business process modeling. In: Proceedings of Workshops and Doctoral Consortium of the 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007), Trondheim, 2007; pp. 349-358.

[YDG10] Yan, Z.; Dijkman, R.; Grefen, P.: Fast business process similarity search with feature-based similarity estimation. In: Proc. of the 2010 international conference on the move to meaningful internet systems, Hersonissos, 2010; pp. 60-77.

206