Automated Semantic Analyses of Conceptual Models

Jörg Becker, Daniel Pfeiffer

European Research Center for Information Systems, Leonardo-Campus 3,
48149 Münster, Germany
{becker, pfeiffer}@ercis.de

Abstract. Conceptual models are an important repository for knowledge in companies and public institutions. The retrieval of this knowledge can prepare reorganisations projects and support IT investment decisions. However, so far this information source has hardly been utilized in automated analyses. We argue that if modelling grammars are endowed with specific characteristics the semantics of the resulting models can be analysed in an automated manner. Based on the conceptual modelling grammar PICTURE we demonstrate that knowledge retrieval with conceptual models is facilitated.

Keywords: Model Comparison, Domain Specific Grammars, Conceptual Modelling, PICTURE.

1 Conceptual Models and Analysis Operations

Conceptual models are an important knowledge source for managerial decisions. They can contain information about the business processes, the resulting products and services, the data structures, or involved organisational units [1-3]. A detailed analysis of conceptual models can, therefore, help to asses and improve the efficiency of an organisation. However, actually semantic analyses of conceptual models are mainly performed manually with high personnel and financial efforts.

Although, automated semantic analyses could significantly improve the value of conceptual models in practice they have not established yet. The reasons for this situation lay in the complexity of the endeavour. An automated semantic analysis of conceptual models faces the following problems:

1. *Conceptual models which are incorporated in an analysis must have been created by following the same modelling rules in order to minimize variations.* For a successful automated evaluation it is crucial that the models use the same domain vocabulary and exhibit an equal grade of abstraction as well as a comparable level of detail. Consequently, there must be detailed rules which ensure that modellers describe a certain domain in a similar form. However, existing modelling grammars do not sufficiently restrict the modeller [4].
2. *Semantic conflicts in the models must be resolved.* To perform semantic analyses in an automated manner the models must be searched for certain reoccurring structural patterns. The semantic aspects of these patterns must be considered in order to yield meaningful results. Thus, defects such as synonym and homonym

conflicts in the models must be addressed. This cannot be done in a fully automated form since conceptual models are not sufficiently formalized. A semi-automatic approach, however, leads to significant efforts [5].

3. *The modelling grammar must allow for the specification of semantically meaningful analysis operations.* It has to support a level of detail as well as a proximity to the domain which complies with a subsequent analysis. Hence, modelling grammars and the analysis mechanism must comply with each other. If for example the modelling grammar does not support the annotation of organisational units the identification of “Ping Pong”-processes (alternate between different organisational units multiple times) is not possible. Common modelling grammars are not designed to support analysis operations.

Semantic analyses of conceptual models in an automated manner require specific modelling grammar characteristics. To perform an analysis with models that exhibit an arbitrary structure hampers the identification of semantically meaningful results. The aim of this paper is to show which modelling grammar characteristics are required, in order to identify semantically relevant model elements in an automated manner. We will explain how these grammar properties foster the retrieval of significant knowledge for reorganisation and IT investment decisions.

2 Grammar Characteristics for an Automated Analysis

A modelling grammar and the corresponding modelling process must ensure the following characteristics in order to significantly simplify analytical operations on the resulting models [6]:

- D1 All constructs of the modelling grammar must be semantically disjoint.
- D2 The resulting models must not contain different domain statements with the same meaning as labels of model elements (no synonyms).
- D3 No construct of the modelling grammar is permitted to have more than one meaning (no homonyms).
- D4 The resulting models must not contain domain statements as labels of model elements that have more than one meaning (no homonyms).

Conditions D1 and D3 refer to the modelling grammar. As the modelling grammar is an artificial artefact created by a method engineer, it can be freely modified. Constraints D2 and D4, however, bear on the domain language. The domain language is naturally grown and cannot be easily adjusted as it is the shared property of a language community. The language community decides on how this language is used. One possibility to cope with this problem is to employ a domain ontology in which all homonyms and synonyms are eliminated [7]. Then, it is necessary to oblige the modeller by additional rules or tool support to apply the ontology to label the model elements and not to use any other domain vocabulary.

However, there is an alterative approach to meet the conditions D2 and D4. The relevant domain language statements can be transformed into constructs of the modelling grammar [8]. A domain specific grammar emerges. Now, the modeller must not use domain statements at all but is limited to the constructs of the modelling grammar. The drawback of this modification is that the modelling grammar loses the flexibility to be used in an arbitrary domain but is now rather specific to a particular knowledge area. However, this solution provides the following advantages:

1. *Abstraction conflicts are avoided.* Condition D1 is stricter than constraint D2, as D2 just demands for the elimination of synonyms within the domain language but D1 additionally requires the modelling constructs to be semantically disjoint. Thus, it is not possible to have more general and more specific modelling constructs. There cannot be two differently abstract constructs as they must be semantically disjoint. Therefore, the constraint D1 prevents abstraction conflicts. If the modelling grammar is declared as mandatory in a certain project, modellers are forced to represent the reality in an identically abstract manner.
2. *Semantic analysis operations can be defined at the design time of the CMG.* When domain statements become constructs of the modelling grammar, the modelling grammar does not only provide measures to structure the domain but also it is sufficient to describe it. The use of additional statements from a domain language is no longer required. With a multi-purpose grammar the domain specific terms are not part of the modelling grammar but are added when the conceptual model is constructed. Thus, multi-purpose grammars allow for the definition of semantic analysis operations after the models have been constructed. However, with domain specific grammars this can already be done at the design time of the grammar as the domain statements are part of it. Although, domain specific modelling grammars are overall less general than multi-purpose grammars, from the perspective of their semantic operations, they are more widely reusable. For example, an UML diagram can be examined for how many activities it comprises. This is no domain specific analysis though. Alternatively, suppose a domain specific grammar with the construct “Enter data into IT”. With this grammar it is possible to count how often paper documents are digitised. Hence, consequences for the introduction of a document management system can be derived. With UML such an analysis could be defined based on a set of existing models but not with the grammar alone. Moreover, contrary to the domain specific grammar with UML the conflicts C1-C4 had to be addressed. A domain specific grammar that exhibits the characteristics D1 and D3 addresses the before mentioned problems and thus allows for an analysis of the resulting conceptual models in an automated form [6].

3. Evaluation and Future Research

PICTURE is a domain specific grammar for the efficient representation of the process landscape in public administrations [9]. With PICTURE processes are modelled as a sequential flow of domain specific process building blocks. A process building block represents a predefined set of activities within an administrational process. The semantics of a process building block is specified by a corresponding domain statement which is part of the modelling grammar. The PICTURE-grammar has been constructed in consideration of the conditions D1 and D3. It has been strived for a set of modelling constructs whose members are semantically disjoint (cf. D1) and do not comprise homonyms (cf. D3). The constructs have been chosen based on an analysis of existing process models from the public administration domain and an evaluation of electronic government literature. As all PICTURE-constructs correspond with language statements from the public sector, they are domain specific.

So far in two large case studies 21 modellers have collected more than 330 processes with PICTURE. In these two projects, the acquisition of the processes took significantly less time than with traditional modelling approaches and reorganisation proposals could be derived in an automated manner [9].

The perspective of the paper is not to take conceptual models as given when they are analysed. Rather, we have argued that if the modelling grammar complies with certain rules then a semantic analysis process can be noticeably simplified. The consequence is that the models are not created for a single purpose anymore but have a lifecycle in which they are modified and extended to keep up with the changes in the environment. The definition of operations on conceptual models like transformation, integration or search helps to address this issue [10]. It is due to further research to evaluate how the proposed grammar characteristics influence these semantic operations.

Acknowledgments. The work published in this paper is partly funded by the European Commission through the STREP PICTURE. It does not represent the view of European Commission or the PICTURE consortium and the authors are solely responsible for the paper's content.

References

[1] Green, P. F., Rosemann, M.: Integrated Process Modeling: An Ontological Evaluation. Information Systems 25 (2000) 73-87

[2] Shanks, G., Tansley, E., Weber, R.: Using ontology to validate conceptual models. Communications of the ACM 46 (2003) 85-89

[3] Becker, J., Kugeler, M., Rosemann, M.: Process Management - A Guide for the Design of Business Processes. Springer, Berlin et al. (2003)

[4] Hadar, I., Soffer, P.: Variations in Conceptual Modeling: Classification and Ontological Analysis. Journal of the AIS 7 (2006) 568-592

[5] Pfeiffer, D., Gehlert, A.: A framework for comparing conceptual models. In: Proc. Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2005) (2005) 108-122

[6] Pfeiffer, D.: Constructing comparable conceptual models with domain specific languages. In: Proc. 15th European Conference on Information Systems (ECIS2007) (2007)

[7] Mena, E., Kashyap, V., Sheth, A., Illarramendi, A.: Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. In: Proc. First International Conference on Formal Ontologies in Information Systems (1998)

[8] Guizzardi, G., Pires, L. F., Sinderen, M. J. v.: On the role of Domain Ontologies in the design of Domain-Specific Visual Modeling Languages. In: Proc. 17th ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2002) (2002)

[9] Becker, J., Algermissen, L., Falk, T., Pfeiffer, D., Fuchs, P.: Model Based Identification and Measurement of Reorganization Potential in Public Administrations – the PICTURE-Approach. In: Proc. Tenth Pacific Asia Conference on Information Systems (PACIS 2006) (2006) 860-875

[10] Rahm, E., Bernstein, P. A.: A survey of approaches to automatic schema matching. The VLDB Journal 10 (2001) 334-350