Towards more Reuse in Conceptual Modeling – A Combined Approach using Contexts

Jörg Becker, Christian Janiesch, Daniel Pfeiffer

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

Abstract. In this paper a novel reuse approach called context-based modeling is proposed. Context-based modeling combines the reuse mechanisms aggregation, restriction, and specialization. Context-based modeling aims at both, minimal preparation of reuse combined with a high degree of guidance to create suitable models. The proposal is conceptually explored.

Keywords: Meta modeling, conceptual modeling, component-based development, reuse, context.

1 Reuse in Conceptual Modeling

Reuse in modeling means to apply the experience of other modelers or experiences from former projects to approach an actual problem [8]. It implies that an existing knowledge base is utilized to avoid starting from scratch. For the reuse of knowledge within conceptual models different approaches have been developed: patterns (P), components (CO), reference models (RM), and views (V) [1, 2, 6, 7, 9].

These reuse approaches apply the following reuse mechanisms: analogy construction (AC), aggregation (A), configuration (C), specialization (S), instantiation (I), and restriction (R) [2]. Table 1 maps the different reuse mechanisms to the corresponding reuse approaches.

Table 1. Mapping of reuse mechanisms to reuse approaches.

	Pattern	Component	Reference Model	View
Analogy Construction	P / AC		RM / AC	
Aggregation		CO / A	RM / A	
Configuration			RM / CF	
Specialization	P / S		RM / S	
Instantiation			RM / I	
Restriction			RM / R	V / R

A comparison of reuse mechanisms can be based on a classification according to two dimensions: The first dimension is the degree of guidance; the second dimension is the degree of preparation necessary to apply the mechanisms.

The degree of preparation defines how much effort is necessary before a certain mechanism can be used. To be able to apply the mechanism of *configuration*, rules must be defined and the model elements must be annotated according to the rules. This process is very time-consuming. The domains of valid values must be specified for an *instantiation* of each placeholder. The mechanism of *restriction* requires the specification of model element types that are permitted or disallowed. For *aggregation* constraints can be defined which restrain the possible combinations of components, but such rules are not obligatory. *Specialization* can exclude certain sorts of modification and allows the general adaptation of models. *Analogy construction* can always be applied and does not require any preparation.

The degree of guidance explains how much the modeler is assisted when a certain mechanism is used. E.g. the guidance of *configuration* is very high. When the parameters are filled with values the model can be configured automatically. Interactions with the user are only necessary to resolve possible conflicts. *Restriction* provides also a high guidance as model element can automatically be removed if their corresponding types do not belong to the perspective which is applied to the model. *Instantiation* specifies the domain of possible values but gives no hints what values to choose in a certain situation. The guidance of *aggregations* and *specialization* depends on whether any restrictions have been specified. It is to assume that an increased degree of guidance requires in the same amount an increased degree of preparation so that no overall gain can be achieved. *Analogy construction* offers no instructions on how to proceed. In Fig. 1 the different reuse mechanisms are arranged in a portfolio.

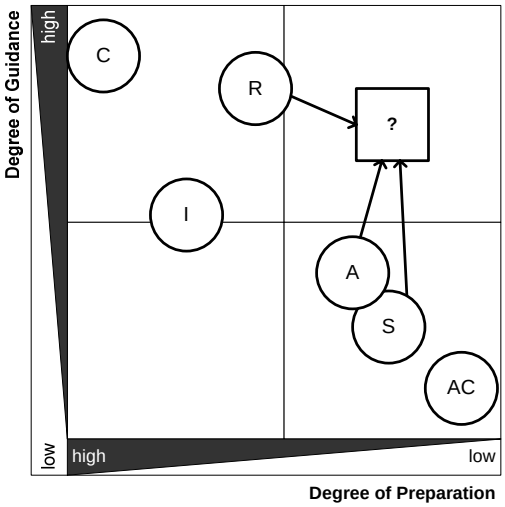

Fig. 1. Portfolio of reuse mechanisms.

The objective of this paper is to construct a reuse approach that can be applied without intensive preparation but that provides, nonetheless, a high degree of guidance. The mechanisms restriction, aggregation and specialization are combined since they seem to be the most promising ones to be used with a varying degree of preparation. The result is a novel reuse approach called context-based modeling.

2 Context-based Modeling

Core to context-based modeling is to model in redundant as well as complementary contexts which each have distinct semantics. In this case, a context describes or represents the environment in which the model or a component thereof has a certain meaning.

A context-based modeling language comprises several contexts. A context acts as a filter as well as a container. In its role as filter, it applies the mechanism of restriction (R) to constrain the available constructs and their relations that can be used within an instance of this context. A context can also be related to (subordinate or sibling) contexts and, thus, applies the mechanism of aggregation (A). By the creation of these relations, model constructs from different contexts are linked. (This entails on a technical level that the contexts are only linked implicitly, since the explicit relations are between the constructs, i. e. the model elements.) Specialization (S) can limit the display of the aggregated components. But more commonly it rather comes into place on a modeling level. If the aggregation and restriction of the model does not suffice the individual needs, it is permissible to specialize the model by addition, deletion, and modification of the existing elements.

Constructs from one context can be aggregated in other contexts. However, the semantics and syntactics that specify the relations of the different constructs to each other is unique to a context. For example, construct A, which is defined in context A', might also be allowed in context B'. However, C might be a subordinate construct to A only in context A'. When linking construct A with constructs D of context D', it is possible to restrict this to A originating only from A'. A from context B' cannot be reused in context D'. Linking constructs – i.e. reusing or merely connecting them – is enabled by so-called context rules. Cf. Fig. 2 for an overview.

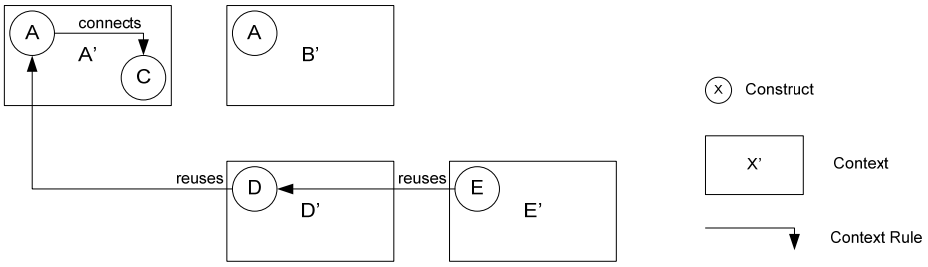

Fig. 2. Exemplary definition of context rules between constructs in different contexts.

This leads to a distinction of different sorts of contexts. While A' provides the basis for all other contexts and does not reuse constructs of other contexts, it acts as a *foundation context*. D' is a context with multiple relations, i. e. it reuses components from other contexts (A') as well as its components are reused (E'). It acts as a *link context*. Context E' in comparison is not intended for reuse but acts as an *aggregation context* which assembles components from other contexts as the highest node in a model. Context B' is uncommon since it is not integrated with other contexts; it acts as a *single context*.

3 Summary and Outlook

Core to context-based modeling proposal is to integrate the mechanisms of restriction, aggregation and specialization. Aggregation is used in a very granular way since model components from within contexts are assembled and contexts are only linked implicitly. The interpretation of restrictions is less restrictive and their semantic consistency lies within the requirements and competence of the modeling language engineer. Furthermore, the mechanism of specialization is incorporated since model component aggregation with explicit context rules allows for adaptations of the resulting model. Contexts are supposed to offer adequate guidance on the reuse of models with a lesser amount of model preparation than current isolated adaptation approaches.

Context-based modeling has proven to be useful for the description of management information systems [3] as well as web information systems [5]. Research suggests that it can be a basis for the design of further situational modeling languages [4] and that it can act as an innovative reuse mechanism.

Practically any existing modeling language can be transformed into a context-based modeling language. Its meaningfulness, however, is strongly depending on the purpose of the modeling endeavor as well as the original design of the modeling language. As with all reuse mechanisms, certain types of modeling languages are more suitable for a problem at hand than others.

References

[1] Alexander, C.: A Pattern Language: Towns, Buildings, Constructions. Oxford Univ. Press, New York (1977)

[2] Becker, J., Delfmann, P., Knackstedt, R.: Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models. In: Proc. Reference Modeling Conference (RefMod) (2006)

[3] Becker, J., Janiesch, C., Pfeiffer, D., Seidel, S.: Evolutionary Method Engineering: Towards a Method for the Analysis and Conception of Management Information Systems. In: Proc. 12th Americas Conference on Information Systems (AMCIS) (2006) 3686-3697

[4] Becker, J., Janiesch, C., Seidel, S., Brelage, C.: A Framework for Situational and Evolutionary Language Adaptation in Information Systems Development. In: Knapp, G., Wojtkowski, G., Zupancic, J., Wrycza, S. (eds.): Advances in Information System Development. Springer (2007)

[5] Brelage, C.: Web Information System Development: Conceptual Modelling of Navigation for Satisfying Information Needs. Dissertation. University of Münster, Berlin (2006)

[6] Brinkkemper, S., Saeki, M., Harmsen, F.: Meta-modelling Based Assembly Techniques for Situational Method Engineering. Information Systems 24 (1999) 209-228

[7] Gupta, D., Prakash, N.: Engineering Methods from Method Requirements Specifications. Requirements Engineering 6 (2001) 135-160

[8] Mylopoulos, J.: Information Modeling in the Time of the Revolution. Information Systems 23 (1998) 127-155

[9] Zachman, J. A.: A Framework for Information Systems Architecture. IBM Systems Journal 26 (1987) 277-293