Domain-specific Customization of Modeling Languages

Michael Eisenbarth and Klaus Schmid

Fraunhofer Institut für Experimentelles Software Engineering,
Sauerwiesen 6, D-67661 Kaiserslautern, Germany
+49 (0) 6301 707 – 181 / – 158
{Michael.Eisenbarth,Klaus.Schmid}@iese.fraunhofer.de

1 Motivation

Most, if not all, specification approaches are more appropriate for modeling in one domain than in another. At this point, there does not exist an approach for selecting a modeling technique for a specific application scenario, or even to adapt a modeling technique to a specific context. For example, SCR was developed as a specification notation for embedded systems, but the specific modeling concepts that makes it appropriate for that task are not made explicit. Similarly, UML/RT is a special modification of the Unified Modeling Language for real time applications but no systematic adaptation process was performed [Do99]. It would be preferable to be able to systematically derive the required modeling concepts from the characteristics of the domain.

In this paper, we introduce a sketch for a systematic approach for customizations of modeling languages and to support the question which modeling notation is applicable for developing a software system within a specific domain. Our long term goal is to provide a structured approach and generic extension mechanism for the development of domain-specific modeling notations, especially UML profiles.

2 Approach

Our approach for a systematic customization of modeling notations consists of three parts [Ei03]: a metamodel containing modeling concepts, domain attributes for system classifications, and a set of selection rules.

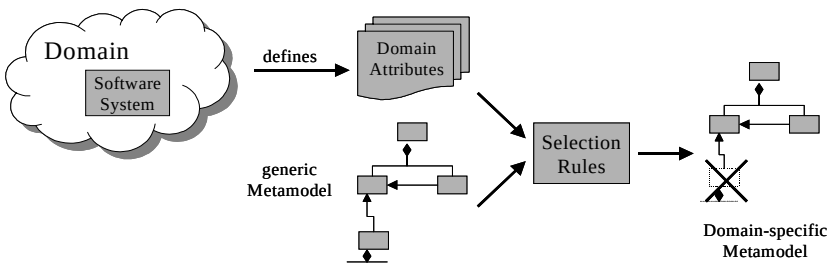

Fig. 1. Customization Approach

Our approach deals with the issue of system classification with a set of domain attributes, which characterize the domain, resp. application to be modeled. We

extended an initial set of attributes presented in [Da93], based on a wide analysis of possible systems and the domain characteristics that make adaptations of the modeling techniques necessary for them. While no guarantee for the completeness of the domain attributes can be given, we found our set so far sufficient in all analysed cases.

We built a metamodel that tries to capture requirements information types relevant to any known domain. The concepts have been grouped into system context related packages, i.e. data description, behaviour description, context description, function description, and a product line package that makes our approach applicable for product line modeling.

The third part consists of a set of selection rules, which are used to identify the concepts in the metamodel that are necessary for modeling an application in a specific domain as given by the domain attributes. On the other hand, unnecessary modeling concepts are removed by the selection rules. The resulting metamodel represents the structural concept framework of an applicable modeling language.

3 Results

We validated our metamodel and the approach with an analysis of three software systems from different domains, namely an ABS system, a car radio, and a telecommunication system. Our case studies showed that our domain attributes greatly simplify the task of identifying relevant domain properties. Especially, a performed case study with the real time modeling language ROOM provided confidence that our approach is capable of identifying domain-specific modeling concepts and instantiating the generic metamodel. The metamodel is not yet finished. We still work on the improvement of the representation and scope of the information types in the metamodel.

4 Conclusions and Future Work

We presented a systematic approach for the domain-specific customization of modeling notations. We developed a generic metamodel for modeling concepts and we derived a set of domain attributes for classifying systems. The selection rules are still an open issue in our approach and will be adressed in future work.

Current work that is strongly related to our research is the development of UML profiles, i.e., generating subsets of the UML that are particularly adapted to a specific problem domain. We plan to further extend our research towards a disciplined approach supporting the derivation of a domain-specific modeling notations. This would also support the systematic development of UML profiles [UML].

REFERENCES

[Do99] B. Douglass. *Real-Time UML Second Edition - Developing Efficient Objects for Embedd ed Systems.* Addison-Wesley, 1999.

[Ei03] M. Eisenbarth. *Domain-specific customizations for Product Line Modeling.* Master Thesis, University of Kaiserslautern, 2003.

[Da93] A. Davis. *Software Requirements - Objects, Functions, & States.* Prentice Hall PTR, 1993.

[UML] UML Profiles, http://www.omg.org/uml.