

Model-Driven Development for End Users, too!?

Gregor Engels, University of Paderborn
engels@upb.de

Own work

Software Engineering is the discipline which develops and evaluates concepts, languages, methods and tools for professional software development in order to yield high-quality software systems. During the last decade, the paradigm of a model-driven development (MDD) has become prominent and is nowadays the accepted method for industrial software development. The main idea is to work with intermediary models in order to bridge the semantic gap between high-level, abstract user requirements and low-level, concrete programs and to support a stepwise refinement process.

This development is supported by agreeing on a standardized unified modelling language (UML). In order to cope with different application domains, the UML provides also standardized means (termed stereotypes) to customize the language, ending up with domain-specific (modelling) languages (DSL).

The UML is strong in modelling the internal functionality of a software system, but weak in modelling user interface aspects of a system. Thus, with respect to the well-known MVC (Model-View-Controller) pattern, the UML supports the Model aspect, while neglecting the View and Controller aspect.

While the model and functionality aspect is of high interest for a software developer, the view and model aspects are of particular interest for end users. Thus, any support for end users in customizing or even changing a software system should first concentrate on means to support the adaption of user interface aspects.

Within in our research on multimedia software systems, we extended and customized the standard UML towards a domain-specific modelling language for multimedia systems ([4], [5]). In particular, we introduced language concepts to define concrete layout aspects as well as complex interactive behaviour. In [1], we developed sophisticated tool support for end-users in order to customize multimedia user interfaces.

In a research cooperation with M. Erwig at Oregon State University, we investigated an approach to introduce an object-oriented model level for the development of spreadsheets [3]. We showed how such a model level prevents end users from hidden, hard to detect errors within spreadsheet applications.

Besides facilitating the development due to abstract, domain-related modelling concepts a model-based development provides the additional advantage of analysing model properties. For instance, in [2], we developed concepts based on patterns to prove the conformity of models with constraints as they might be expressed in given (ISO) standards.

Currently, several PhD students are working on the topic of "Model Quality", in order to develop concepts and tools to understand, define and check the quality of any developed model.

Future Research Topics

While the model-driven development paradigm forms a well-accepted approach towards software development for professional software development, such an approach is in its infancy in end-user software development. It has to be investigated and evaluated, whether end-users at all are willing to develop high-level, abstract models instead of directly dealing with low-level programs. In addition, it has to be investigated what kind of modelling languages are appropriate for end-users. This has to be accompanied by setting up and evaluating case studies within real scenarios.

Furthermore, it has to be investigated whether and how standard model analysis techniques can be transferred to end user software development. In particular, appropriate explanation and help systems have to be designed which translate analysis results into a representation which is understandable by end users.

References

- [1] S. Sauer, G. Engels: *Easy Model-Driven Development of Multimedia User Interfaces with GUIBuilder*. In Proc. 4th International Conference on Universal Access in Human-Computer Interaction (UAHCI 2007), July 2007, Beijing, LNCS, Springer, 2007 (to appear).
- [2] A. Förster, T. Schattkowsky, G. Engels, R. Van Der Straeten: *A Pattern-driven Development Process for Quality Standard-conforming Business Process Models*. In Proc. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Brighton 2006.
- [3] G. Engels, M. Erwig: *ClassSheets: Automatic Generation of Spreadsheet Applications from Object-Oriented Specifications*. In Proc. 20th IEEE/ACM International Conference on Automated Software Engineering, Long Beach, California, USA, November 7-11, 2005, pp. 124-133.
- [4] G. Engels, S. Sauer: *Object-oriented Modeling of Multimedia Applications*. In S.K. Chang (ed.), *Handbook of Software Engineering and Knowledge Engineering*, vol. 2, pp. 21-53, World Scientific, Singapore, 2002.
- [5] S. Sauer, G. Engels: *UML-based Behavior Specification of Interactive Multimedia Applications*. In Proc. IEEE Symposia on Human-Centric Computing Languages and Environments (HCC'01), September 2001, Stresa, Italy, pp. 248-255. IEEE Computer Society Press.