Christian Bunse: **Pattern-Based Refinement and Translation of Object-Oriented Models to Code**

Promotion: Universität Kaiserslautern
Erstgutachter: Prof. Dr. H. Dieter Rombach (Universität Kaiserslautern)
Zweitgutachter: Prof. Dr. Colin Atkinson (Universität Kaiserslautern)
Datum der Prüfung: 15.09.2000
Veröffentlichung: Stuttgart: Fraunhofer IRB Verlag, 2001 (PhD Theses in Experimental Software Engineering 2). Zugl.: Kaiserslautern, Univ., Diss., 2000; ISBN 3-8167-5613-1

Kurzfassung:

This thesis addresses the problem of systematically implementing object-oriented models by identifying the basic nature of the transformation activities involved in the implementation of software systems (i.e., refinement and translation), and defining a method which separates and explicitly distinguishes between refinement and translation, known as SORT. Refinement is viewed as the process of moving from a model presented in some language (e.g. UML) to a more detailed model presented in the same language, while translation is viewed as the process of moving from a description presented in some language (e.g., UML) to a description at the same level of detail presented in another language (e.g., C++). To help developers move from design to code, the SORT method, defined in this thesis, provides a set of patterns (both refinement and translation patterns) organized according to a multi-dimension classification scheme.

In general, the SORT method defines a process for the stepwise refinement of models to a level of abstraction, known as the implementation level, from which models can be translated to source-code. The method also supports continuous quality assurance by means of a systematic visualization and verification technique, as well as by a tool developed in the context of the thesis. In specific projects, developers choose the refinement patterns that best fit their needs (e.g., non-functional requirements) and apply them repeatedly to their models until the models are at the implementation level. Translation patterns are then applied to translate the refined models to code. To exactly define when refinement should end and translation can start, this thesis introduces the concept of the implementation set. An implementation set is composed of modeling constructs which are semantically close to code, and which can thus be directly mapped onto code. In other words, the implementation set defines when a model provides a sufficient level of detail for implementation.

As experienced developers are well aware, the underlying concepts of object-oriented programming are basically the same whatever language or implementation vehicle is used to apply them. Therefore, this thesis defines a specific implementation set, known as the Normal Object Form (NOF), aimed at exploiting that commonality to provide a language independent subset of object-oriented constructs. NOF constructs are semantically close to source-code but are independent of a specific language, thus allowing the translation of NOF models into various target languages.

The effectiveness of the SORT approach has been tested in the context of this thesis by a controlled student experiment. The experiment results show that SORT technology is beneficial and has a positive impact on the traceability, understandability, and verifiability of a design due to the establishment of precise links between models and source-code.

In summary, the SORT method addresses a weak link in modern object-oriented development methods, namely the problem of moving from design to code in a systematic way. In this sense, SORT follows the tradition of one of the most systematic processes defined to date: Cleanroom Software Engineering.