Harald Störrle:
Models of Software Architecture
An Approach based on UML and Petri-Nets

Book-on-Demand, München, 2001; 306+xiv Pages, 253 Figures, 17×22 cm, ISBN 3-8311-1330-0, 35 Euro

Erstgutachter: Prof. Dr. Martin Wirsing
Zweitgutachter: Prof. Dr. Gregor Engels
Datum der Prfung: 20.12.2000

Motivation

How can the design and subsequent analysis of models of software architectures be supported to achieve better quality and productivity? For instance, how can the integration of multiple views be supported by automatic checking pocedures? And how can the communication among stakeholders be supported? Finally, how can reuse and automatic analysis of (parts of) archtiectures be fostered? There have been two kinds of approaches to this subject.

On the one hand, there are Architecture Description Languages (ADLs) like SARA and Wright. Originating in academic settings, they emphasize formal semantics, and complete coverage of all possible concepts. While producing a range of interesting results, ADLs have never really been adopted by industry.

On the other hand, industrial approaches like UML or the Business Object concept have proved practically useful. There are methods and tools, but often these approaches are somewhat restricted, lacking generality and powerful formal methods tool support—exactly what academic ADLs provide.

So, the prime goal of this thesis has been to bridge the gap between the academic and industrial approaches to software architecture communities.

Approach

This is done by creating a detailed and comprehensive logical metamodel, that carefully and conservatively extends the UML 1.3 metamodel. Then, a very detailed mapping from the metamodel to the semantic domain of Petri-nets is provided, so as to allow for automatic translation of dynamic models. Thus, the "frontend" designers are used to (ROOM/UML) is transparently equipped with a formal semantics that may be used for automatically checking formal properties such as the consistency among different views.

Results

The following results are documented in the book:
(a) a family of notations for architectural modeling based on ROOM;
(b) firmly based on a complete and detailed logical metamodel, extending the UML 1.3 metamodel (maps easily to UML 1.4);
(c) transparent formal semantics for dynamic and static views of architectural models, and almost 30 generic consistency conditions for view integration;
(d) extended example of a industrial application and a set of process patterns;
(e) a thorough review and evaluation of previous research, extensive index and glossary.

Since the completion of this work, several aspects have additionally been published at several conferences and workshops. A tool has been implemented as a proof of concept, and a complete CASE environment is currently being developed. Further work based on this approach deals with code-generation and performance prediction aspects of architectural models.

Contents