
Report on a Practical Application of ASMs in Software Design

The FALKO Project

Siemens (Munich)

March 1998 - May 1999

Egon Börger

Peter Päppinghaus

Joachim Schmid

Main purpose of FALKO: Construct & validate timetables for tram/sub-/railway lines

- Timetables constructed offline from raw data which
 - can be input manually via a GUI
 - can be read in from files written in established formats
- Validation done by simulation
 - 3 main components + 1 hidden discrete event simulation kernel controlling the simulation
- Component modeling the railway process has been formally specified and designed using ASMs
 - **integrating an existing library** for numerical computations (computing train velocities, trip times etc.) designed and hand coded conventionally

Goal of „FALKO“

Total Effort for Railway Process Model: 66 pw (person weeks)

- **Design Phase**

- Requirement specification based on predecessor system, developed in meetings of the design team, documented by minutes of the meetings (4 persons 2 weeks)
- Design of 1st draft of executable ASM model (1 person 8 weeks)
- Several cycles of testing and debugging using the ASM Workbench (developed by Giuseppe Del Castillo as part of his Doctoral Thesis at University of Paderborn) (1 person 8 weeks + 1 person 11 weeks)
- Review of 2nd draft of ASM model by design team plus external reviewers (6 persons 1 week)
- Several cycles of improving, testing and debugging (2 persons 5 weeks)

- **Implementation Phase**

- Development of ASM-SL to C++ Code Generator (developed by Joachim Schmid as part of his Doctoral Thesis at University of Ulm, 1 person 4 weeks)
- Specification and implementation of additional handwritten C++ code (1 person 2 weeks)
- Integration of FALKO system including testing and debugging (3 persons 3 weeks)
- Documentation of railway process model component (as collection of HTML documents with literate programming features, linked to ASM Workbench) and final polish (1 person 6 weeks)

Comparison: ca. 110% of estimated total effort for conventional software design

Effort for Component „Railway Process Model“

ASM Model (source of C++ code generation)

- **ca. 3 000 lines of ASM Workbench code**
- **120 rules**
- **315 functions and relations (240 functions, 75 relations)**
 - 71 dynamic
 - 69 external
 - 59 static
 - 116 derived

C++ Code

- **ca. 9 000 lines of generated C++ Code**
- **ca. 2 900 additional lines of handwritten C++ Code, consisting of**
 - ca. 400 lines wrapper code for interfacing to other components of FALKO
 - ca. 2 500 lines low-level library code
- **Railway process model of prototypical predecessor system:
ca. 20 000 lines of (handwritten) C++ code**

Size of ASM Model and C++ Code for „Railway Process Model“

Courtesy of P. Päppinghaus,
Siemens AG

Experiences

- **Design**

- Developers and reviewers had no problems to understand the formal specification (ASM model)
- Tests with ASM model uncovered, at an early stage, bugs also in other components of the package

- **Implementation**

- Coherence of specification and implementation by seamless tool support
- Performance loss of generated code tolerable even for product quality code

- **Maintenance**

- 4 installations at Subway Vienna Operator since March 1999, 1 in daily use
- Customer reported no bugs so far
- 2 bugs discovered in tests run by FALKO developers, temporary fixes for the 2 bugs (by handhacking the generated C++ code) later replaced by recompiling updated ASM model

- **Development Environment** built-up, supporting seamless flow from spec to code

Experiences with Use of ASMs for Software Design

References

- **Egon Börger, Peter Päppinghaus, Joachim Schmid:** Report on a Practical Application of ASMs in Software Design
 - » In: Y. Gurevich et al.(Eds.): Abstract State Machines. Theory and Applications. Springer LNCS 1912, 2002, 361-366
- **Giuseppe Del Castillo:** The ASM Workbench. A Tool Environment for Computer-Aided Analysis and Validation of Abstract State Machine Models
 - » Dissertation, Heinz Nixdorf Institut, Universität Paderborn, 2001, pp.iv + 212, ISBN 3 - 9311 466 – 82 – 5
- **J. Schmid:** Compiling Abstract State Machines to C.
 - In: Journal of Universal Computer Science 7 (11), 2001, 1069-1088

References