Third Workshop on Scenarios and State Machines: Models, Algorithms, and Tools (SCESM'04)

Holger Giese
Software Engineering Group
University of Paderborn
Warburger Straße 100
D-33098 Paderborn, Germany
hg@upb.de

Ingolf Krüger
Department of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive 0114
La Jolla, CA 92093-0114,USA
ikrueger@cs.ucsd.edu

1. Introduction

This third workshop in the SCESM (SCEnarios and State Machines) series is motivated by the very successful preceding workshops on this topic. This effort started with a workshop on scenario-based round-trip engineering at OOPSLA 2000. Two SCESM workshops at ICSE 2002 [1] and ICSE 2003 [2], and a Dagstuhl Seminar in September 2003 [3] followed. SCESM'04 continues this tradition by bringing together researchers and practitioners interested in advancing models, algorithms and tools for scenario- and state-oriented approaches to software and systems engineering.

2. Motivation

Scenarios and state machines have emerged as two important modeling perspectives on the reactive behavior of complex systems. Scenarios typically represent a partial view on the interactions between multiple components; state machines typically represent the complete behavior of individual components. The methodological potentials of the combination of partial and complete behavior perspectives have yet to be fully exploited in the development process for complex, reactive systems. Automated tool support – based on algorithms relating scenarios and state machines for analysis, design, implementation or validation – offers great promise for improving the current practice of software engineering.

Scenarios and state machines complement each other in multiple ways: Scenario modeling focuses on the interactions between system components or actors; communication between such entities is viewed as the principal modeling construct. Interaction modeling has proven useful for getting an overview of the interplay between components to fulfill a certain task, and sharing this insight among project stakeholders. In this sense, scenarios represent projections of the overall system behavior onto individual tasks. State-based modeling, on the other hand, usually concentrates on the internal states of individual components. This yields projections of the overall system behavior onto individual components. State-based formalisms have a distinguished history in software engineering, particularly in object-oriented, distributed and real-time system design. In many cases, the formalisms are amenable to rigorous automated analysis, such as model checking.

The UML [4] is a prominent example of a modeling language providing both scenario- and state-based description techniques for object- and component behavior. Scenarios are captured by means of use cases, or sequence and collaboration diagrams, where component interaction is in the center of concern. Statecharts capture state-based behavior, typically on a per-component basis. In the telecommunications domain, the ITU standards of MSC (Message Sequence Charts) [5] and SDL (Specification and Description Language) provide a similar distinction between exemplary and complete behavior descriptions. In both cases, there exists little methodological or tool support for transitions between scenarios and state machines, let alone means for refining or validating one against the other.

The relation between scenarios and state machines, however, is important and can be exploited throughout the development process. Scenarios can form the starting point for synthesis of prototypic state machines, and thus serve as an element in forward engineering. Scenarios and state machines together foster system comprehension when analyzing existing or planned systems. Scenarios representing traces of system execution help in understanding interacting state machines in a reengineering effort. Both interaction scenarios and state machines can be applied in specifying component interfaces. Simulation tools can use scenarios or animated state machines to make their relationships explicit.

To fully unleash their methodological potentials in practice, the relation between scenarios and state machine

Proceedings of the 26th International Conference on Software Engineering (ICSE'04)
0270-5257/04 $20.00 © 2004 **IEEE**

models needs to be supported by corresponding algorithms and tools. For instance, conformance between scenarios and state machine models can be analyzed using model checking techniques. The information available in a given set of scenarios can be used to automatically synthesize a state machine or a set of state machines implementing the specified interaction behavior. In each of these cases, the underlying notion of conformance has to bridge the gap between partial scenarios and complete, operational state machines; heuristics typically play an important role in this process.

3. Topic of Interest

The main area of interest for this third ICSE workshop is the application of proposed techniques for relating scenarios and state machines, their proper integration in the overall software development process, and the benefits and limitations of the current proposals in practice. Thus, practical problems (such as scalability) will be discussed by means of case studies provided on the workshop homepage (*http://scesm04.upb.de/index.html*) or other real-world examples. Specific areas of interests include, but are not limited to the following four main discussion topics:

1) Experience reports (on the case studies or other real-world examples)
2) Methodology (how to analyze/capture key scenarios; how to refine partial towards complete behavior specifications; when best to apply scenarios as opposed to state machines, and vice versa; what is the relationship between scenarios, state machines, and aspect-oriented modeling and implementation) .
3) Algorithms (e.g., checking consistency, synthesizing tests from scenarios, synthesizing state machines from scenarios, implied scenarios, generating scenarios from state machines) and corresponding tools.
4) Models and notations (requirements for different application areas, shortcomings in current notations, new suggestions for models or notations, categorizations)

4. Workshop Organization

A call for papers was widely distributed and made available at *http://scesm04.upb.de/index.html*. Prospective participants were requested to submit a position paper of up to five pages. In order to focus contributions and provide a common basis for discussion, we encouraged participants to make use of one of the two case studies which are provided on the workshop homepage. However, the use of the case study was not mandatory. Authors were still allowed to use their own examples or focus on specific or related aspects of the case study.

The program committee reviewed submissions. Each submission was reviewed by at least three different program committee members. The program committee selected the position papers based on relevance, soundness and novelty.

The workshop is open and will have a maximum number of 30 participants; however, priority registration shall be given to those with accepted papers.

The workshop will be divided into sessions. The opening session will include a keynote by Dr. Michael von der Beeck from BMW AG. Topics of the working sessions will be determined based on the distribution of accepted position papers. Each session will cluster short presentations where authors will have an opportunity to present the main ideas of their position papers. The presentations shall serve as an opening statement of the session, after which there will be at least half an hour reserved for in-depth discussion of the presentations, related issues, and the implications for future research.

In a final summary session, the discussions which arose during the working sessions will be summarized and further activities and required future research will be discussed.

5. Workshop Organizers

Francis Bordeleau, University of Carleton, Canada.
Holger Giese, University of Paderborn, Germany.
Martin Glinz, University of Zurich, Switzerland.
Jeff Kramer, Imperial College, UK.
Ingolf Krüger, University of California, San Diego, USA.
Stefan Leue, University of Freiburg, Germany.
Wilhelm Schäfer, University of Paderborn, Germany.
Tarja Systä, Tampere University of Technology, Finland.
Sebastian Uchitel, Imperial College, UK.
Jon Whittle, QSS Inc. NASA Ames, USA.
Albert Zündorf, University of Kassel, Germany.

References

[1] First ICSE Workshop on Scenarios and State Machines: Model, Algorithms and Tools (SCESM'02), 2002.
[2] Second ICSE Workshop on Scenarios and State Machines: Model, Algorithms and Tools (SCESM'03), 2002. http://www.doc.ic.ac.uk/~su2/SCESM/
[3] Dagstuhl Seminar Scenarios: Models, Transformations and Tools. Schloss Dagstuhl, Wadern, Germany, 2003. http://www.dagstuhl.de/03371/
[4] OMG, The Unified Modeling Language, 2003. http://www.omg.org
[5] ITU-TS, Message Sequence Charts, Recommendation Z.120, Geneva, 1996.