Workshop Description of 5th Intl. Workshop on Scenarios and State Machines: Models–Algorithms–and Tools (SCESM)

Jon Whittle
Department of Information & Software Engineering
George Mason University
Fairfax, VA 22030, USA
jwhittle@ise.gmu.edu

Leif Geiger
Software Engineering Research Group
University of Kassel
Wilhelmshöher Allee 73,
34121 Kassel, Germany
leif.geiger@uni-kassel.de

Michael Meisinger
Institut für Informatik
Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany
meisinge@in.tum.de

ABSTRACT

SCESM’06 is the 5th International Workshop on Scenarios and State Machines: Models, Algorithms and Tools. It is a one day ICSE’06 workshop. Details about SCESM’06 may be found at http://ise.gmu.edu/scesm06/.

Categories and Subject Descriptors
D.2.2 [Software Engineering]: Design Tools and Techniques – State Diagrams; D.2.1 [Software Engineering]: Requirements/Specifications

General Terms
Languages, Design, Theory, Algorithms

Keywords
Scenarios, State Machines, Message Sequence Charts, UML

1. DESCRIPTION

Scenarios and state machines are two of the fundamental modeling perspectives for developing behavioral abstractions of complex, reactive software. Scenarios represent a partial view on the interactions between multiple system components; state machines typically model the complete behavior of individual components. Both perspectives have advantages—scenarios are easily understood by stakeholders at all levels, lead naturally to tests and focus on inter-component communication; state machines, on the other hand, provide precise descriptions of component behavior and can be used for generating implementations (either manually or automatically). In particular, UML supports both notations and in UML2.0 [8], scenarios can be modeled in a much more expressive way than with previous versions.

Scenarios and state machines offer complementary views of a software-intensive system. Scenarios show global interactions; state machines show local behavior. For successful software development, both views should be brought together. This workshop aims to investigate the connection between scenarios and state machines, assess how this connection can be exploited to improve software development, support the evaluation of techniques that exploit the connection and support the showcasing of developer tools based on both views.

There are many ways to combine scenarios and state machines. For example:

1. scenarios can form the starting point for the synthesis of prototypical 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;
2. scenarios representing traces of system execution help in understanding interacting state machines in a reengineering effort;
3. both scenarios and state machines can be applied in specifying component interfaces;
4. simulation tools can use scenarios and/or animated state machines in the validation of software requirements, software architecture or software design;
5. and scenarios and state machines can be used together in model-based testing to generate tests.

The SCESM series of workshops dates back to a 2000 OOPSLA workshop [5] and has been followed since by four successful ICSE workshops ([6], [7], [2], [1]) as well as a Dagstuhl Seminar [3]. During this time, there has been a progression from theoretical studies of algorithms for connecting scenarios and state machines, to application of algorithms on real-world case studies, through to the development of advanced methodologies that exploit the algorithms. Since 2002, participants are invited to use case studies [4], which are part of real-world industrial projects (NASA’s CTAS system and Paderborn New Railway System). They serve as a common ground for discussing and comparing the various approaches, algorithms and tools.

Copyright is held by the author/owner.
ICSE’06 May 20–28, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.
In 2006 [9], the workshop continued its emphasis on development of fundamental algorithms, application and methodology, but also looked to evaluate and showcase software development tools that connect scenarios and state machines. The workshop also broadened its scope to cover topics only partially addressed previously in the workshop series, such as the relation of scenarios/state machines to software architecture, aspects and model-based testing.

2. SUBMISSIONS AND FORMAT

Three types of submission were solicited. Research papers describe novel, unpublished results. Tool papers describe implementations related to the workshop’s goals. Experience reports describe applications of related techniques in industry. All papers underwent a thorough review process by a highly regarded international program committee. Each paper received at least three reviews. A total of eighteen papers were submitted—fifteen research papers, six tool papers and one experience report (including overlaps). Around 50% of the submissions were selected for presentation at the workshop (in no particular order):

- “Assert and Negate Revisited: Modal Semantics for UML Sequence Diagrams,” by Harel and Maoz
- “Aspect-Oriented Software Design with a Variant of UML/STD,” by Nakajima and Tamai
- “A Scenario Based Notation for Specifying Temporal Properties,” by Marco, Inverardi and Pelliccione
- “A Schema Language for Coordinating Construction and Composition of Partial Behavior Descriptions,” by Grieskamp and Kicillof
- “Nobodys perfect: Interactive Synthesis from Parametrized RealTime Scenarios,” by Giese, Henkler, Hirsch and Klein
- “A Collaboration-based Approach to Service Specification and Detection of Implied Scenarios,” by Castejón and Braek
- “A Comparative Survey of Scenario-based to State-based Model Synthesis,” by Liang, Dingel and Diskin
- “Inferring Operational requirements from Scenarios and Goal Models Using Inductive Learning,” by Alrajeh, Russo and Uchitel
- “Automating Software Architecture Exploration with M2Aspects,” by Krüger, Lee and Meisinger

Two additional papers were accepted as part of a special tool demonstration session:

- “Test Ready UML Statechart Models,” by Murthy, Anitha, Mahesh and Subramanyan
- “Scenario-Driven Modeling and Validation of Requirements Models,” by Seybold, Meier and Glinz

3. IMPACT

The SCESM series of workshops has been a highly successful forum for the discussion of issues related to scenarios and state machines. Papers on the topic have appeared in the proceedings of ICSE and other major conferences as well as in top journals such as IEEE Transactions on Software Engineering, ACM Transactions on Software Engineering and Methodology and IEEE Software. Invited talks at previous SCESM workshops have been given by well-known figures in the field: Axel van Lamsweerde (ICSE’02), David Harel (ICSE’03), Michael von der Beeck (ICSE’04) and Hans Dieter-Ehrich (ICSE’05).

4. PROGRAM COMMITTEE

The SCESM’06 program committee consisted of:

- Yves Bontemps (University of Namur),
- Alexander Egyed (Teknowledge Corp.),
- Holger Giese (University of Paderborn),
- Martin Glinz (University of Zurich),
- Jean-Marc Jézéquel (IRISA, France),
- Joy Chakraborty (Motorola India),
- Ingolf Krüger (University of California, San Diego),
- Hillel Kugler (New York University),
- Stefan Leue (University of Konstanz),
- Wilhelm Schäfer (University of Paderborn),
- Stéphane Somé (University of Ottawa),
- Tarja Systä (Tampere University of Technology),
- Sebastian Uchitel (Imperial College, London),
- Albert Zündorf (University of Kassel)

and the organizers

5. REFERENCES