Guest Editorial: Special Section on Interaction and State-Based Modeling

Sebastian Uchitel, Member, IEEE Computer Society, Manfred Broy, Member, IEEE Computer Society, Ingolf H. Krüger, Member, IEEE, and Jon Whittle

1 INTRODUCTION

Behavior models play a key role in the engineering of software-based systems. They are the basis for systematic approaches to requirements elicitation, specification, architecture design, simulation, code generation, and verification and validation. A range of notations, techniques, and tools supporting behavior modeling for these development tasks have been suggested. Underlying these notations, techniques, and tools, two complementary approaches to modeling behavior can be identified: interaction-based and state-based modeling.

Interaction-based modeling focuses on the interactions between actors and components in a system. Consequently, communication between such entities is viewed as the principal modeling construct. Interaction modeling, commonly realized using scenario and use case techniques, provides an overall view of a system which is particularly suited for supporting communication between project stakeholders.

State-based modeling, on the other hand, concentrates on the internal behavior of components. State-based formalisms have a distinguished history in software engineering, particularly in object-oriented, distributed, and real-time system design. In many cases, state-based formalisms support rigorous automated analysis, such as model checking.

Approaches for both interaction and state-based modeling are in common use in industry. In addition, they have been studied extensively by researchers. However, the relationship between these modeling approaches has yet to be fully understood and exploited. There has been a significant recent effort in this direction. This special section of the IEEE Transactions on Software Engineering brings together a collection of high quality papers that show the advances in the area. The call for papers for this special issue attracted 40 submissions, all of which were reviewed to the high standards of the journal. The review process resulted in the selection of five excellent articles representing a cross-section of the state-of-the-art in understanding and leveraging the relationship between interaction-based and state-based modeling.

2 THE PAPERS

The papers that make up this section illustrate the breadth of the area in domains, formalisms, and techniques. The papers focus mainly on synthesis and verification. However, novel modeling approaches bringing together interaction and state-based behavior are also covered. The papers address a variety of application domains, from business processes and Web services to reactive and real-time systems, and include a variety of formalisms, from Message Sequence Charts, UML Sequence Diagrams, and Live Sequence Charts to agent strategies, π-Calculus, and Timed Automata.

The first paper, “From Live Sequence Charts to State Machine and Back: A Guided Tour” by Y. Bontemps, P. Heymans, and P.Y. Schobbens addresses the theoretical and practical limitations, as well as the challenges, of the scenario-development approach. The paper focuses on complexity and decidability of verification and synthesis within the context of Live Sequence Charts.

In “Interaction Protocols as Design Abstractions for Business Processes,” N. Desai, A.U. Mallya, A.K. Chopra, and M.P. Singh present a novel modeling approach for business processes that brings together interaction-based and state-based modeling. Interaction models are extended to capture exchange of commitments, which provides the context for a notion of completeness of an interaction. Flexible composition mechanisms are given for combining MSCs with diverse role, data, and commitment dependencies. Finally, interactions are refined with local policies to produce concrete, executable protocols.

In the third paper, “A Scenario-Matching Approach to the Description and Model Checking of Real-Time Properties,” V. Braberman, N. Kicillof, and A. Olivero address the specification and checking of real-time properties. A novel graphical scenario language is presented which permits the description of complex properties not expressible with sequence chart-like notations. In addition, results on decidability of verification are provided.
In “Synchronizability of Conversations among Web Services,” X. Fu, T. Bultan, and J. Su study the interplay between global interaction modeling and local behavior modeling in the web services domain. The authors show how top-down and bottom-up development of Web-services requires careful understanding of issues such as reliability and show that the notion of synchronizability can enable tractable analyses to support such development. The authors also provide insights into the relation between conversation protocols and message sequence charts for which issues such as realizability have been studied extensively.

Finally, the fifth paper, “Generating Annotated Behavior Models from End-User Scenarios” by C. Damas, B. Lambeau, P. Dupont, and A. van Lamsweerde, builds on and extends work on grammar induction to provide an iterative method for synthesizing state-based models from message sequence charts. The algorithm exploits the notion of language kernel to guarantee convergence to an approximation of the intended system behavior and uses some heuristics to reduce the number of questions that the modelers must answer in order for the process to converge.

The guest editors would like to thank all authors who have submitted papers and, in particular, the reviewers for this special issue for their hard work and the high quality of the reviews.

Sebastian Uchitel
Manfred Broy
Ingolf H. Krüger
Jon Whittle
Guest Editors

Sebastian Uchitel received a degree in computer science from the University of Buenos Aires, Argentina, and the PhD degree in computing from Imperial College London, where he also worked as a research associate. He is a lecturer at Imperial College London. His research is in software engineering and, particularly, in behavior modeling and analysis of requirements and design for complex software-intensive systems. His research currently focuses on scenario-based specifications, behavior model synthesis, and partial behavior models such as modal transitions systems. He has been active in the IEEE and the ACM since 1998, serving on numerous program committees. He is currently program cochair of the 21st IEEE/ACM International Conference on Automated Software Engineering. He is a member of the IEEE Computer Society.

Manfred Broy is a professor in the Department of Informatics at the Technical University of Munich, Germany. His research interests are software and systems engineering, comprised of both theoretical and practical aspects. He leads a research group working on a number of industrial projects that apply mathematically-based, reactive systems, software architectures, description techniques, verification, and validation. The CASE tool AutoFocus was developed in his group. He is a member of the IEEE Computer Society.

Ingolf H. Krüger received the PhD degree from Technische Universität München, Germany in 2000 and the MA degree from the University of Texas at Austin in 1996. He is an assistant professor in residence in the Computer Science and Engineering (CSE) Department of the University of California San Diego’s (UCSD) Jacobs School of Engineering, leading the “Service-Oriented Software and Systems Engineering Laboratory”; he also directs the “Software & Systems Architecture & Integration” functional area within the California Institute for Telecommunications and Information Technology (Calt2). He is a member of the UCSD Divisional Council of Calt2. His major research interests are service-oriented software & systems engineering for distributed, reactive systems, software architectures, description techniques, verification & validation, and development processes. The application domains to which he applies his research results span the entire range from networked embedded systems to Internet-wide business information architectures. He is a member of the IEEE.

Jon Whittle received the degree in mathematics from the University of Oxford and the master’s and PhD degrees in artificial intelligence from the University of Edinburgh. He is an associate professor of software engineering at George Mason University. He worked for six years in the Automated Software Engineering Group at NASA Ames Research Center, where he developed novel software engineering techniques and applied them to NASA applications. He is on the program committee and steering committee of the ACM/IEEE MODELS conference series and served as conference chair of the International Conference on UML in 2003. He has also taught as a guest professor at IISc Bangalore and IIT Kanpur in India and at TU Braunschweig in Germany.