Synthesizing Modal Transition Systems from Triggered Scenarios

German Emir Sibay, Victor Brberman, Sebastian Uchitel, Member, IEEE Computer Society, and Jeff Kramer, Member, IEEE Computer Society

Abstract—Synthesis of operational behavior models from scenario-based specifications has been extensively studied. The focus has been mainly on either existential or universal interpretations. One noteworthy exception is Live Sequence Charts (LSCs), which provides expressive constructs for conditional universal scenarios and some limited support for nonconditional existential scenarios. In this paper, we propose a scenario-based language that supports both existential and universal interpretations for conditional scenarios. Existing model synthesis techniques use traditional two-valued behavior models, such as Labeled Transition Systems. These are not sufficiently expressive to accommodate specification languages with both existential and universal scenarios. We therefore shift the target of synthesis to Modal Transition Systems (MTS), an extension of labeled Transition Systems that can distinguish between required, unknown, and prohibited behavior to capture the semantics of existential and universal scenarios. Modal Transition Systems support elaboration of behavior models through refinement, which complements an incremental elaboration process suitable for specifying behavior with scenario-based notations. The synthesis algorithm that we define constructs a Modal Transition System that uses refinement to characterize all the Labeled Transition Systems models that satisfy a mixed, conditional existential and universal scenario-based specification. We show how this combination of scenario language, synthesis, and Modal Transition Systems supports behavior model elaboration.

Index Terms—Scenarios, MTS, synthesis, partial behavior models

1 INTRODUCTION

Operational behavioral models such as labeled Transition Systems (LTSs) are convenient formalisms for modeling and reasoning about system behavior at the architectural level. These models provide a basis for a wide range of automated (and semi-automatic) analysis techniques, such as model-checking, simulation, and animation. One of the limitations of operational behavior modeling is the complexity of building the models in the first place. Operational behavioral model construction remains a difficult, labor-intensive task that requires considerable expertise. To address this, a wide range of techniques for supporting (semi-)automated synthesis of operational behavior models has been investigated. In particular, synthesis from scenarios and use cases has been studied extensively [1], [2], [3], [4], [5].

Scenario-based specifications such as Message Sequence Charts (MSCs) [6] describe how system components, the environment, and users interact in order to provide system level functionality. Their simplicity and intuitive graphical representation facilitate stakeholder involvement, making them popular for requirements elicitation. Model synthesis from scenario-based specifications facilitates early analysis, validation, and incremental elaboration of behavior models. A range of scenario description languages and associated behavior model synthesis algorithms have been developed (e.g., [1], [7], [8]). Although they differ in many aspects, a noteworthy semantic distinction is whether scenarios are interpreted as existential or universal statements. An existential scenario provides an example of system behavior, one that the system-to-be is required to provide. A universal scenario provides a rule that all system behavior is expected to satisfy. Although each approach is typically geared to one interpretation or the other, some languages, notably Live Sequence Charts (LSCs) [3], provide syntactic and semantic support for both interpretations. The motivation is that during the requirements process, there is a progressive shift from existential statements in the form of examples and use cases to universal statements in the form of declarative properties. A scenario-based language that supports both interpretations is better equipped to support this shift.

1.1 Existing Triggered Scenarios (eTSs)

Despite the variety of existing approaches, no language and associated synthesis algorithm is suitable for describing conditional existential scenarios. Consider the statement “if the user inserts a valid card into the ATM, and then enters the correct password, she/he shall be able to request cash and have it dispensed by the ATM.” This statement is existential in that it provides an example of system execution. It is also conditional in the sense that requesting
7 CONCLUSION

In this paper, we have defined a scenario-specification language which includes support for describing triggered existential and universal scenarios. We have also defined a synthesis algorithm that constructs MTS models which characterize via refinement all LTS models that conform both to the existential and universal aspects of the scenario-based description.

A novel aspect of the approach is the use of triggered existential scenarios which have a branching semantics. This is in line with existing informal scenario-based and use case-based approaches to requirements engineering exploiting the expressive power of MTS in an operational behavior model.

The approach supports behavior elaboration through the analysis and refinement of underspecified system behavior using MTS merging, model checking, inspection, and animation, moving from examples to comprehensive descriptions during the behavior elaboration process.

In future work, we intend to continue to develop and integrate support for elicitation and elaboration of behavior models using MTS. In particular, we are investigating the use of learning, in the form of Inductive Logic Programming [40], to aid the elaboration process. We aim to develop techniques and tools to support identifying, providing feedback, and resolving inconsistencies in the process of merging MTS that result from scenario-based specifications.

ACKNOWLEDGMENTS

The work reported herein was partially supported by ERC STG PBM-FIMBSE, CONICET PIP11-200801-00955KA4, UBACYT W0813, PICT 1774, PICT PAE 227, PICT PAE 2278, and MEALS 295261. Finally, the authors would like to thank the anonymous reviewers for their careful and insightful reviews that greatly improved the quality of this paper.

REFERENCES