

Test-driven Tailorable UML-based Software Documentation

Thorsten Haendler, Stefan Sobernig, and Mark Strembeck, WU Vienna

Motivation

Context

- UML: *de facto* standard for modeling/documenting the structure and behavior of software systems

Problems

- Manual specification and maintenance of UML models is tedious and often error-prone
- Reverse-engineering UML models from system-execution traces suffers from the problems of *model-size explosion* and *high detail level*

Deriving UML Models from Runtime Tests

- Model-driven approach for the interactive automated derivation of UML-based documentation from runtime tests
- Resulting models human-tailorable for software-maintenance tasks

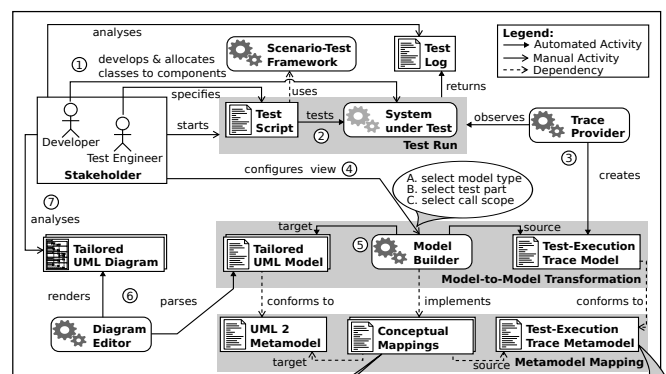


Fig. 1: Conceptual overview.

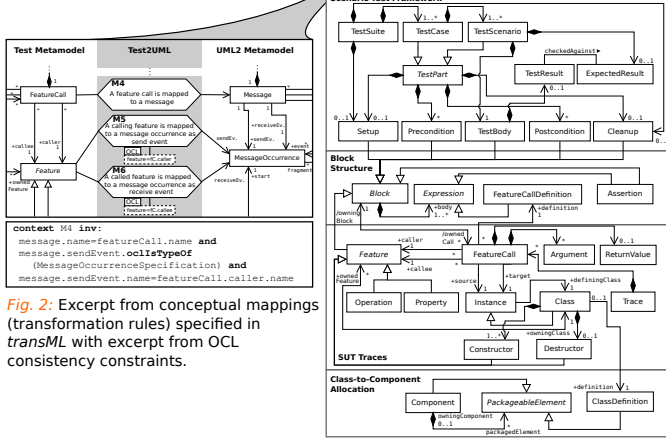


Fig. 2: Excerpt from conceptual mappings (transformation rules) specified in transML with excerpt from OCL consistency constraints.

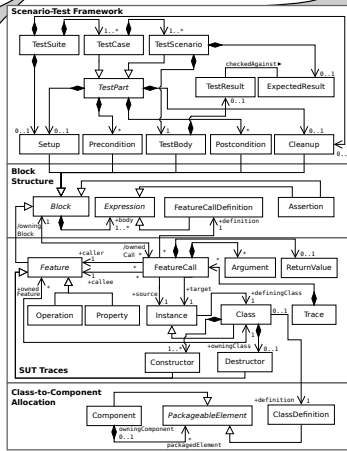


Fig. 3: Test-execution trace metamodel.

Tailoring the Resulting UML Models

- Test viewpoint for tailoring the models to provide information relevant for performing software-maintenance tasks

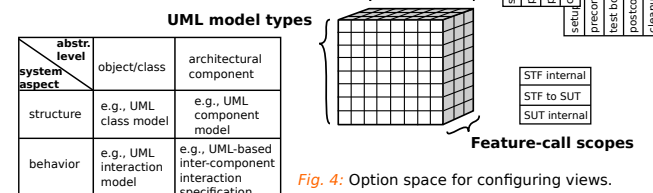


Fig. 4: Option space for configuring views.

KaleidoScope

- Our **prototype implementation** integrates with the testing framework *STORM*
- Trace provider: instruments the test execution applying *NX/Tcl* introspection techniques (e.g., message interceptors, callstack introspection)
- Model builder: *QVT Operational* for transforming view and trace models into tailored UML models (handled in XMI)

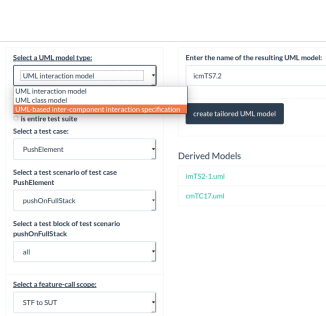


Fig. 5: GUI for configuring views.

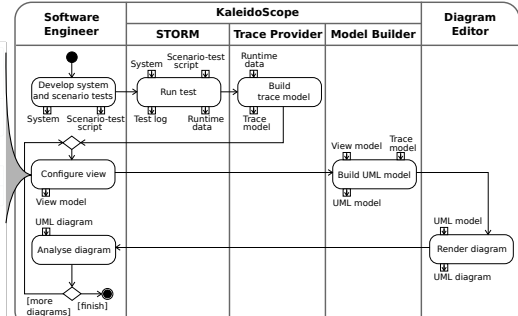


Fig. 6: Deriving tailored UML-based documentation with KaleidoScope.

- STORM-based tests check the correctness of the derivation process
- KaleidoScope is available for download from <http://nm.wu.ac.at/nm/haendler>

Exemplary Application

Task/Stakeholder Role	View Configuration	Model
Test-script review by test developer	<ul style="list-style-type: none"> UML interaction model (object level) Specific test scenario (pushOnFullStack) Calls running from STF to SUT 	(A)
Source-code modification by software developer	<ul style="list-style-type: none"> UML interaction model (object level) Spec. test block (test body of pushOnFullStack) Calls running from STF to SUT and SUT internal 	(B)
Re-using/modifying an architectural component by software architect or component developer resp.	<ul style="list-style-type: none"> UML based inter-component interaction spec. Specific test scenario (scenarioX) Calls running from STF to SUT and SUT internal 	(C)

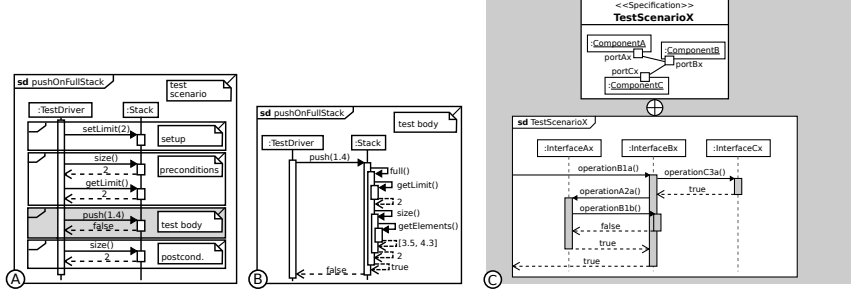


Fig. 7: Examples of different tasks/stakeholders with corresponding view configurations and resulting derived UML diagrams.

Conclusion

- Test viewpoint for the interactive automated derivation of UML models
- Approach generic by defining metamodel mappings (transformation rules)
- Prototype implementation (proof of concept)

Requirements/Limitations

- Test framework which allows for instrumenting the test execution
- Corresponding test suite structured in test parts
- System under test (SUT) having an object-oriented design/implementation

Future Work

- Case studies with large-scale industry projects and controlled experiments

References

- Haendler, Thorsten, Sobernig, Stefan, and Strembeck, Mark: **Deriving UML-based Specifications of Inter-Component Interactions from Runtime Tests**. In Proc. 31st ACM Symposium on Applied Computing (SAC 2016), Software Engineering Track, 1573-1575. Pisa, Italy: ACM Press.
- Haendler, Thorsten, Sobernig, Stefan, and Strembeck, Mark: **Deriving Tailored UML Interaction Models from Scenario-Based Runtime Tests**. In: Software Technologies (Revised Selected Papers of ICSE 2015), Communications in Computer and Information Science (CCIS), Volume 586, 326-348. Switzerland: Springer International Publishing.
- Haendler, Thorsten, Sobernig, Stefan, and Strembeck, Mark: **An Approach for the Semi-automated Derivation of UML Interaction Models from Scenario-based Runtime Tests**. In Proc. 10th International Conference on Software Engineering and Applications (ICSE/AE 2015), 229-240. Colmar, France: SciTePress.
- Haendler, Thorsten: **KaleidoScope**. 2015. <http://nm.wu.ac.at/nm/haendler>
- Neumann, Gustaf and Sobernig, Stefan: **Next-Scripting Framework**. 2015. <http://next-scripting.org>
- Strembeck, Mark: **Testing Policy-based Systems with Scenarios**. In Proc. 10th IASTED International Conference on Software Engineering (SE 2011), 64-71. Innsbruck, Austria: ACTA Press.