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

Fig. 2: Software Documentation Test-driven Tailorable UML-based consistency constraints (transformation rules) specified in Feature References

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

Exemplary Application

Tailoring the Resulting UML Models
• Test viewpoint for tailoring the models to provide information relevant for performing software-maintenance tasks

References
• Haendler, Thorsten: KaleidoScope 2015. http://nm.wu.ac.at/nenhaendler

KaleidoScope
• Our prototype implementation integrates with the testing framework STORM
• Trace provider: instruments the test execution applying NX/TCI 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)

Exemplary Application

Tailoring the Resulting UML Models
• Test viewpoint for tailoring the models to provide information relevant for performing software-maintenance tasks

References
• Haendler, Thorsten: KaleidoScope 2015. http://nm.wu.ac.at/nenhaendler