TEST MODELING FOR CONTEXT-AWARE UBIQUITOUS APPLICATIONS WITH FEATURE PETRI NETS

Georg Püschel, Ronny Seiger, Thomas Schlegel

Copenhagen, June 25, 2012
OUTLINE

- Introduction/Motivation
- Main Questions
- Technologies
  - Model Based Testing
  - Feature Models
  - Dynamic Feature Petri Nets
  - Context Rules
- Workflow for Test Case Generation
- Tool Support
- Conclusion
MOTIVATION – MOBILE APPLICATIONS (1)

- Static variability

Platform A.1

Platform A.2

Platform A.3

Test Object (App)

Platform B.1

Platform B.2
MOTIVATION – MOBILE APPLICATIONS (2)

- Dynamic variability

- Context changes: “external data, that may influence the application” [1]
TESTING OF UBIQUITOUS APPS – MAIN QUESTIONS

- Taming the feature and configuration space
- Validation of a ubiq. app against a common test model (despite multiple platforms)
- Modeling of dynamic changes (changing context)
- Description of test case coverage (equivalence classes)
- Automation of test case generation and reuse

EXAMPLE “TRANSAPP”

- Automatic language selection (location-based)
- Internet or local dictionary look-up (connection-based)
TECHNOLOGIES – MBT

“Model-based Testing is the automation of the design of black box tests.” [2]

- Higher productivity through automation
- Reusability
- Measurable coverage
- Traceability
Test Modeling for Context-aware Ubiquitous Applications with Feature Petri Nets

**TECHNOLOGIES – FEATURE MODELS**

```
FS_{sample} = \{GPS \text{ Localization}, \text{Local Dictionary Translation}, \text{Core}, \text{GPS}\}
```
TECHNOLOGIES – DFPN (1)

Dynamic Feature Petri Nets [3]
- Transitions, places
- Tokens, arcs
- Control feature binding at runtime
- Transitions extended by: application condition, update expression

Application condition: defines for which configurations a transition may fire

\[ \varphi ::= a \mid (\varphi \land \varphi) \mid (\varphi \lor \varphi) \mid \neg \varphi \mid true \text{, with } a \in F \]

Update expression: manipulates the feature configuration

\[ u ::= noop \mid a \text{ on} \mid a \text{ off} \mid u; u \quad \text{[3]} \]

\[ u' ::= u \mid \text{action}(x) \mid \text{verify}(x, v) \mid u'; u' \quad v \quad \text{.. Verdict} \]

\[ x \quad \text{.. Term} \]

Example:

\[ \text{Internet Translation} \]
\[ \land \text{GPS Localization} \]
\[ \text{action(enter(PHRASE))}; \quad \text{action(translate)} \]
TECHNOLOGIES – DFPN (2)

Internet Translation
\[ \land \quad \text{GPS Localization} \]
\[ \text{action(enter(PHRASE))}; \]
\[ \text{action(translate)} \]

true
\[ \text{verify(translation(GPS, Internet), FAIL)} \]

Dictionary Translation
\[ \land \quad \text{GPS Localization} \]
\[ \text{action(enter(PHRASE))}; \]
\[ \text{action(translate)} \]

true
\[ \text{verify(translation(Net, Dictionary), FAIL)} \]
TECHNOLOGIES – CONTEXT RULES

\[ \varphi(\text{on} | \text{off}) \Rightarrow (\text{action}(x) | \text{verify}(x, v)) \]

Example: \[ \text{GPS Localization on} \Rightarrow \text{action('enable GPS')} \]

Context feature model

\[
\begin{align*}
\text{true} & \quad \text{Net Localization off;} \\
\text{true} & \quad \text{GPS Localization on;} \\
\text{true} & \quad \text{Dictionary Translation on;} \\
\text{true} & \quad \text{Internet Translation off;}
\end{align*}
\]

\[
\begin{align*}
\text{action(updateCtx('enable gps'))} & \\
\text{action(updateCtx('disable internet'))} & \\
\text{noop} & \\
\end{align*}
\]
WORKFLOW FOR TEST CASE GENERATION

- feature modeled configuration space
- product generator (SAT)
- context rules
- sequence generator
- products/test configurations
- SUT feature-aware behavioral model

SAT .. SATisfiability
SUT .. System Under Test
TOOL SUPPORT

- MATE – Mobile Application Test Environment (http://www.quality-mate.org)

1. DFPN
2. Context rules
3. Feature model
CONCLUSION

- **Model-based testing** for automation of black box tests
- **Feature models** for coping with platform and feature complexity, esp. in ubiq. apps
- **Petri net-based models** for describing dynamic and static behavior
- **Context rules** for considering context changes
- **Workflow** for deriving test cases / test suites

Advantages:
- Extended formal description of test cases
- Leveraging properties of Petri nets
- Automation, reusability
- Traceability, measurable coverage

- Modeling tool in ongoing development

**TESTING IS FUN!**
REFERENCES


CONTACT

- Dipl.-Inf. Georg Püschel, Georg.Pueschel1@mailbox.tu-dresden.de
- Dipl.-Inf. Ronny Seiger, Ronny.Seiger@tu-dresden.de
- Jun.-Prof. Dr.-Ing. Thomas Schlegel, Thomas.Schlegel@tu-dresden.de

- http://www.seus.inf.tu-dresden.de
- http://www.quality-mate.org

CREDITS

- Kudos to Georg Püschel for conducting the majority of this research and providing contents, graphics and slides for this presentation.