Extending Software Integration Testing Using Aspects in Symbian OS

Jani Pesonen

Technology Platforms

Nokia Corporation

NOKIA
Connecting People

Outline

- Software product-lines
- Issues in conventional testing techniques
- Aspect-orientation
- Practical experiment
- Evaluation results
- Conclusions and future work

Software Product-lines

- A product-line enforces planned and enabled large-scale re-use
  - Thinking the product-line as whole
  - Common asset base
- Product-lines with plenty of products stress the asset base
  - Variety of different products and features
  - Wide support complicates the product-line when involving thorough and generic features
- Testing product-lines is somewhat easier
  - Verified asset base
  - However, other testing related problems remain…

Issues in Conventional Testing Techniques

- Comprehensive testing of large systems is impossible in practice
  - Excessive amount of test cases
  - Testing requires considerable investments
  - Project schedules and cost-effectiveness cut time and money
- Inability to capture cross-cutting issues under test
  - Inability to isolate tangling code for testing
  - Laborious and expensive, easily neglected
- Focusing on the most important issues
  - Selection between desired and unwanted test cases is difficult
  - Based on experience and best practices

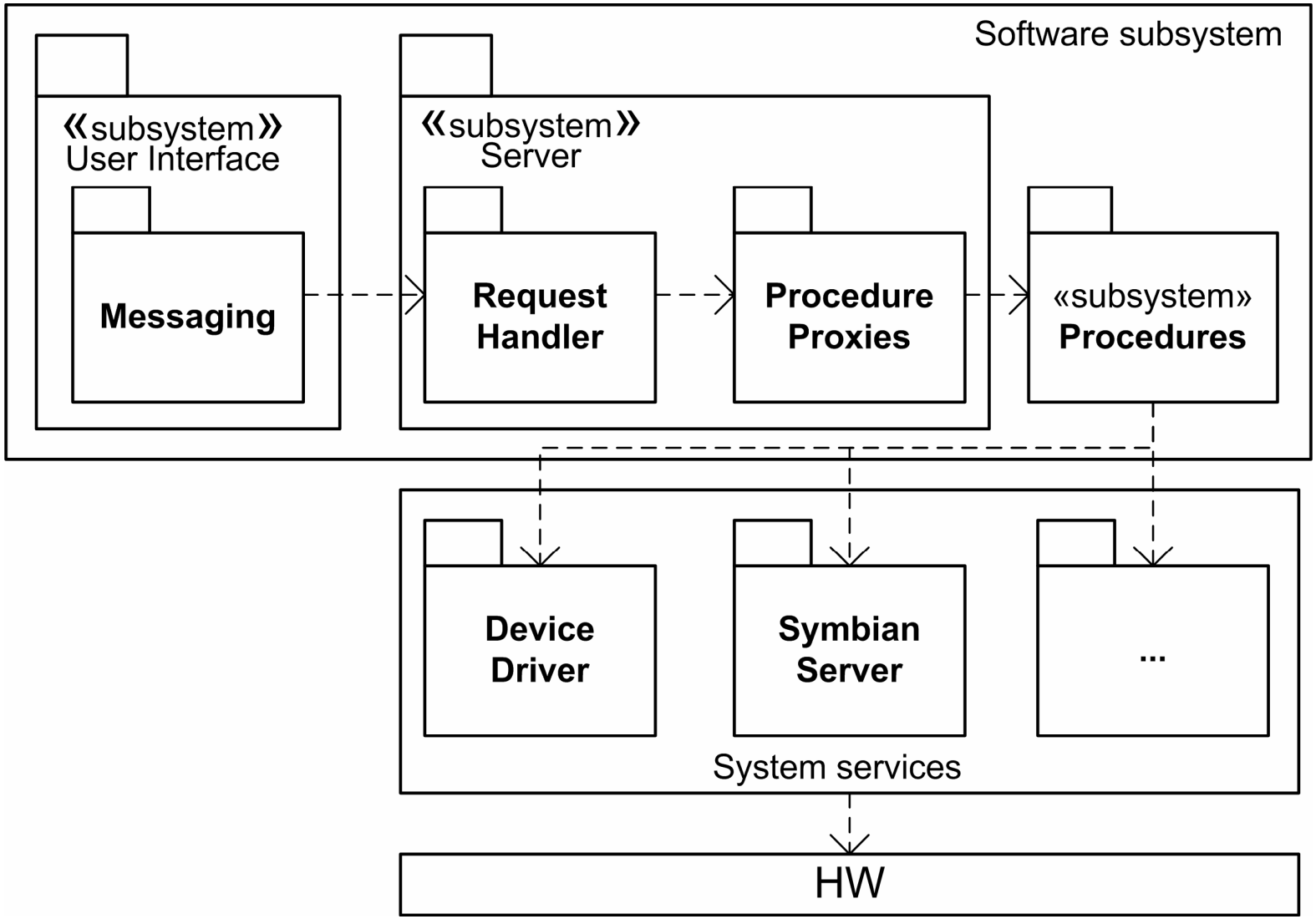
Aspect-orientation

- Aspect-orientation provides means for capturing tangling and cross-cutting issues in programs
  - Modularizing them as manageable units.
- Aspect-oriented Programming (AOP) makes it possible to weave new operations into already existing systems
  - This non-invasive nature makes it an impressive tool for capturing testability concerns
  - An attractive patching mechanism
- Possible to extend and overwrite methods
  - Provides means for manipulating already existing behaviors

Practical Experiment

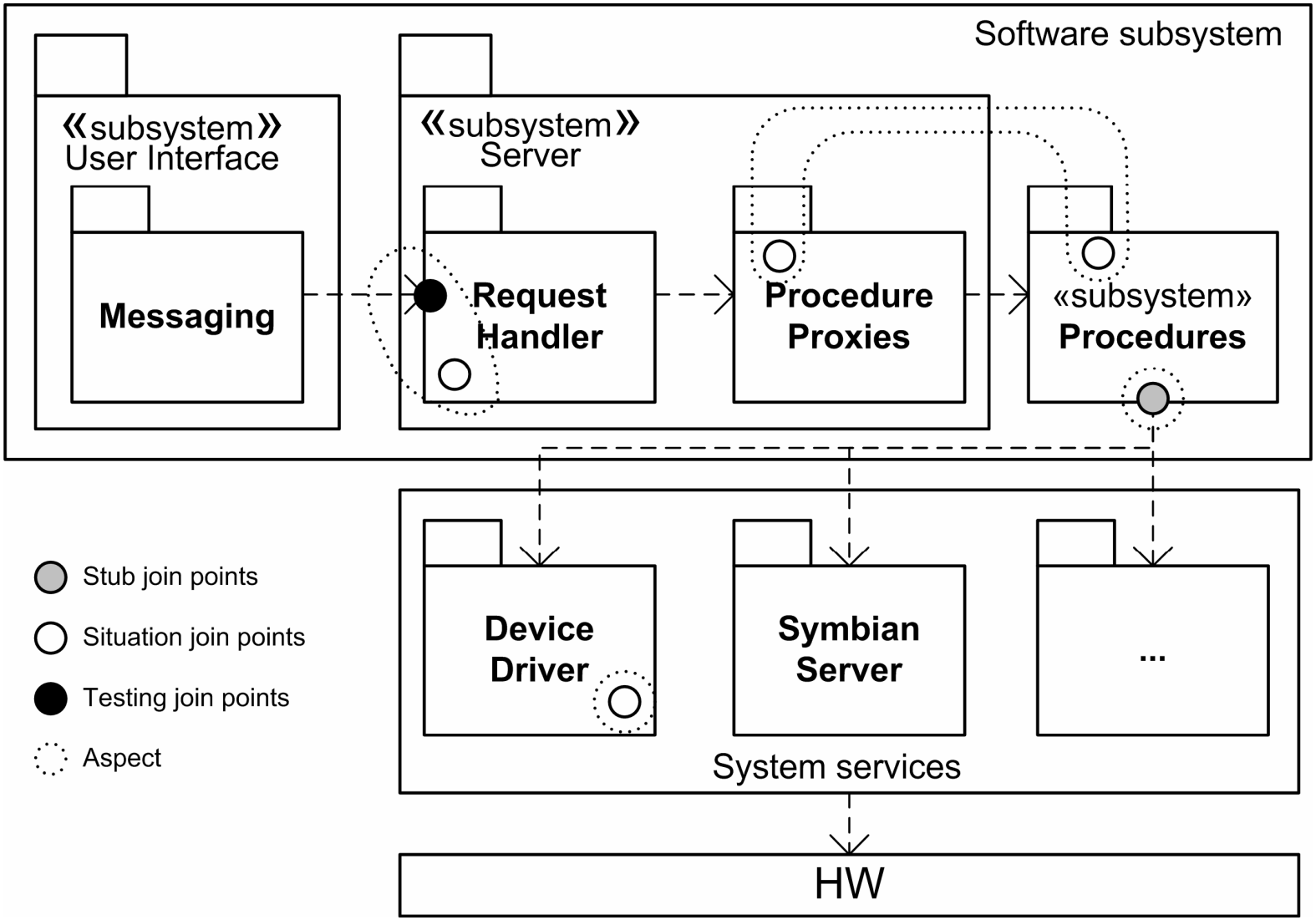
- A test harness using AspectC++ in Symbian OS context
  - A product-family example
- Three different kinds of aspects:
  - Aspects as stub drivers and use cases
  - Aspects for identifying certain error situations
  - Generic aspects for tracking system-wide behaviors

Example: System Under Test



NOKIA
Connecting People

Example: System Under Test



NOKIA
Connecting People

Evaluation Results

- Aspect-oriented programming is practical in capturing intrinsic functionalities
  - Recalls for insight on which parts of the system benefit from aspects
  - Using only aspects to implement all the testing effort is not reasonable
- Difficult to identify system invariants to be formulated as aspects
  - Stubs tend to become complex also when using aspects
- Practical problems were mainly related to the problematic tool chain
  - AspectC++ and Symbian OS together form a challenging combination
  - Source-to-source weaver limits the possibilities

Conclusions and Future Work

- Aspects provide interesting means to enhance integration testing
- Potential lies on regression testing
  - Easy to prevent errors from recurring
  - Tracking of bad behaviors
  - Measuring system resources
- Future work:
  - Methods for effectively capturing the key assets as aspects
  - Ability to inject test cases to already compiled software
  - Generating testing aspects from high-level descriptions

Contact Information

Jani Pesonen

Nokia Corporation
Technology Platforms
P.O.Box 88
FI-33721 Tampere
FINLAND

E-mail: jani.p.pesonen@nokia.com

NOKIA
Connecting People