Production-Testing of Embedded Systems with Aspects

IBM Verification Conference 2005

Jani Pesonen

Technology Platforms

Nokia Corporation

Mika Katara and Tommi Mikkonen

Institute of Software Systems

Tampere University of Technology

NOKIA
Connecting People

Outline

- Software Product-lines
- Aspect-orientation
- Production testing
- Expected benefits from aspect
- Practical experiment
- Evaluation results
- Conclusions and future work

Software Product-lines

- A product-line enforces planned and enabled large-scale re-use of implementations, requirements, and specifications
  - Thinking the product-line as whole, instead of individual products
  - Common asset base forms the basis for re-use acting as foundation for product varation
- Product-lines with plenty of products stress the asset base
  - Variety of different products broadens assets supply
  - Wide support complicates the product-line when involving thorough and generic features
- Requires a software architecture appropriately designed to the purpose
  - Adopting product-line principles is expensive task for existing systems
  - Expanding the core assets to support each and every feature is exhaustive

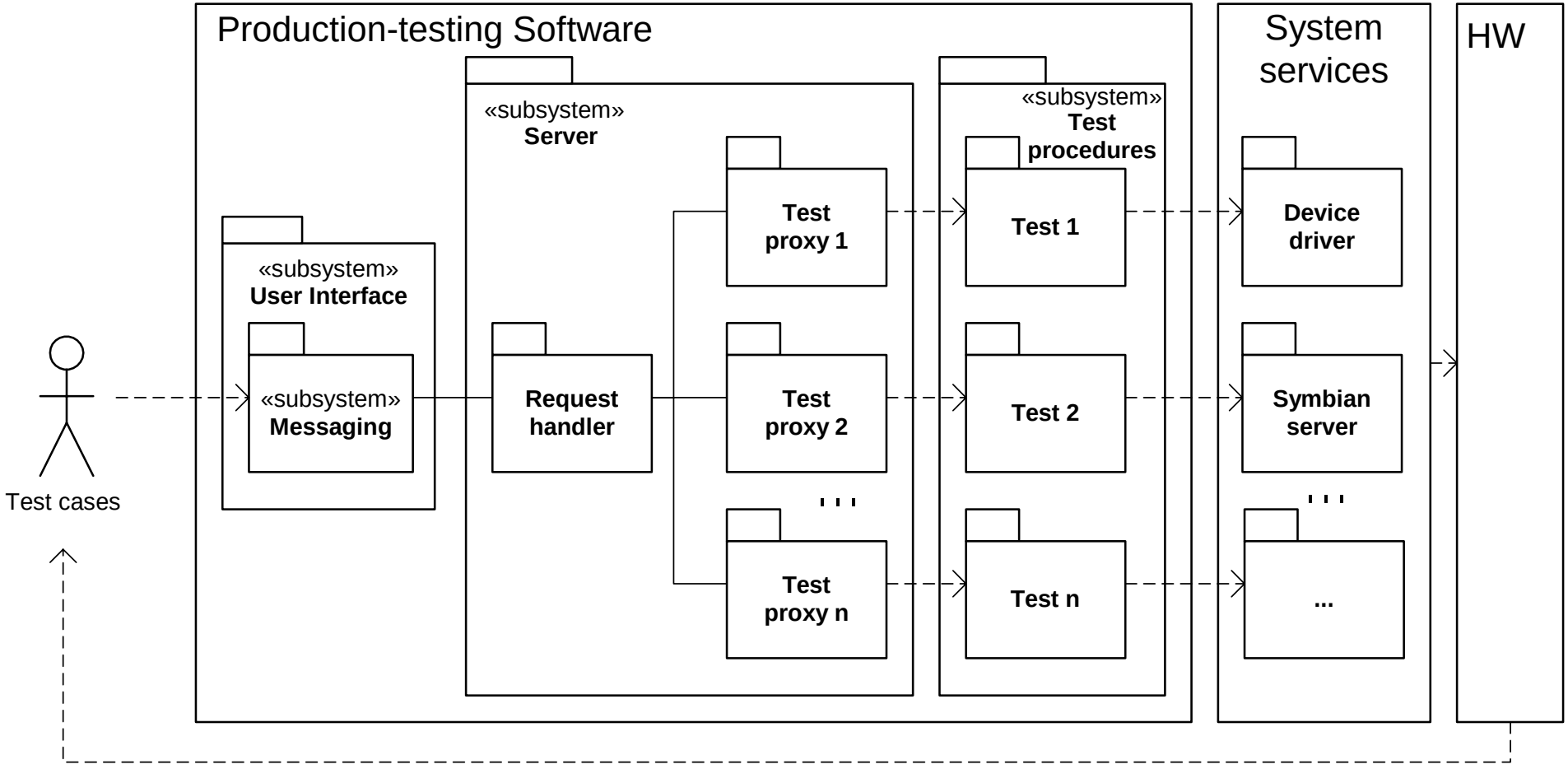
Aspect-orientation

- Aspect-oriented approaches provide facilities for sophisticated dealing with tangling and cross-cutting issues in programs
  - Modularizing cross-cutting concerns
- Aspect-oriented programming (AOP) makes it possible to weave new operations into already existing systems, thus creating new behaviors
  - Non-invasive nature makes it an attractive patching mechanism
- Also possible to override and expand methods
  - Provides means for manipulating the behaviors that already existed

Production-testing

- Production testing is validation of the devices' manufacturing correctness
  - Production-testing SW must be adapted to all different devices
  - Challenging variation task to support all possible HW configurations and combinations
- With conventional techniques
  - Attempts to group high variability implementations tend to increase the system's complexity
  - Tangled and scattered code is a burden in especially code blocks with long historical background and in case of one-shot deviations from the mainstream
- This paper studies solving these challenges with aspect-oriented techniques in a real industry scale system
  - Environmental constraints: Symbian OS, C++, embedded device

Production-testing Harness

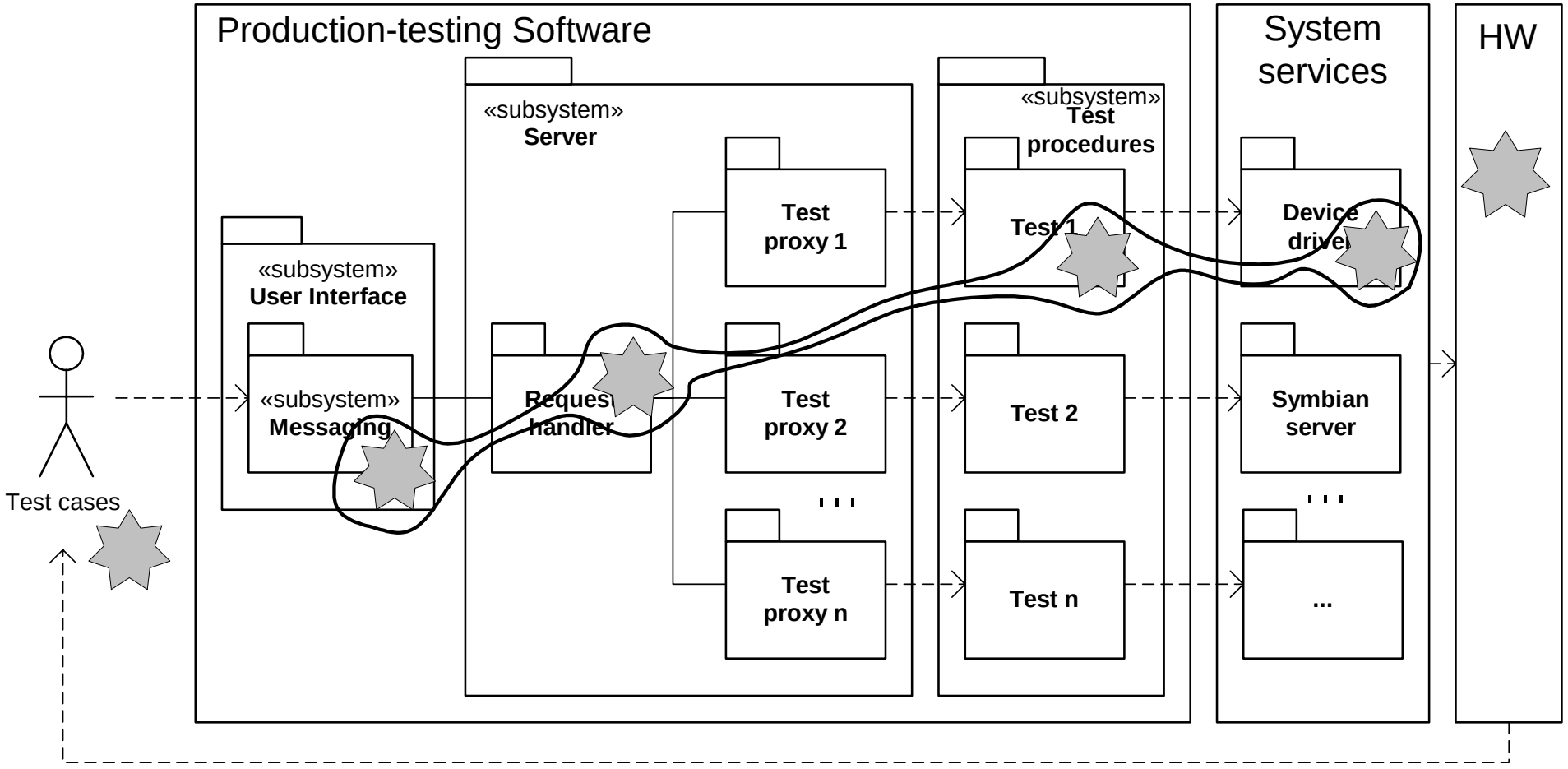


NOKIA
Connecting People

Expected Benefits from Aspects

- One-shot implementations are easy to weave into the system without any need for restructuring the system architecture and intervening the implementation
- Tracing etc. similar ”extra” tasks can be introduced into the system easily without touching the actual implementation
- Conventional techniques tend to add code overhead, often experienced as gratuitous repetition, which could be avoided with aspects

Example: Aspects in Production-testing



NOKIA
Connecting People

Practical Experiment

- Foresightedness is required in factoring implementations between aspect-oriented techniques and conventional techniques
- Proposed Solution:
  - Common parts are included in the object-oriented base implementation and device-specific ones are woven into that implementation as aspects
  - Aspects instrument product level specializations into the common assets
  - Common extra tasks should be instrumented as common core aspects
- Two groups of aspects:
  - Test specialization aspects (hardware dependent)
    - One-shot camera example
  - General purpose core aspects (product level instrumentation)
    - Enabling tracing support using aspects
- AspectC++ used for weaving aspects

Evaluation Results

- AOP-techniques are convenient for low-level extensions, but seem to lack the potential for large-scale implementations
  - Cooperation of conventional and aspect implementation can be difficult to follow and debug
  - Aspects were considered superior in weaving in small implementations not interfering with the conventional parts
    - Domain-specific rules needed for applying AOP
- Practical problems were mainly related to immaturity of the tools and the environment
  - Tight relationship between tools is likely to paralyze development in product-family
  - AspectC++ and Symbian OS together form a challenging combination
- Thorough evaluation of development tools is crucial in demonstrating effectiveness of a new technology in industrial use

Conclusions and Future Work

- Conclusions
  - AOP in production-testing looks promising on paper, but requires enhancements prior to industrial scale use
  - Experiments indicate challenges in integrating several specialized tools
  - Practical experiments in an embedded setting question the practicability of the available aspect-techniques in size-delicate appliances
- Future work
  - Investigations of embedded product-family architectures with aspect-oriented techniques
  - Configuration management of systems with a combination of conventional and aspect-oriented technologies
  - Applying aspects as tools for non-invasive system and integration testing

