A Case Study in Software Adaptation

Giuseppe Valetto and Gail Kaiser
Telecom Italia Lab / Columbia University

WOSS '02

November 18, 2002

Dynamic adaptation of SW

- The ability to influence the structure, state and behavior of a running complex SW system
- Can be seen as a run-time extension of maintenance practices
  - Corrective or perfective
- Strongly automated

- Our targets: systems of (legacy) systems

Dynamic adaptation of an Internet service: a case study

- An industrial Internet application
  - Thousands of users
  - QoS is business-critical
- A complex distributed service
  - Multi-channel instant messaging
  - Including legacy / 3rd party components
  - Expensive to deploy, configure, monitor manage

Scope of the case study

Dynamic adaptation aimed at:

- Automated management
  - Automated deployment and instantiation
  - On-the-fly configuration
  - Continuous monitoring and feedback (tune, repair)
- Service optimization
  - Automated scalability
  - Component re-configuration according to monitored QoS parameters
  - Component fault detection
  - System-wide repair

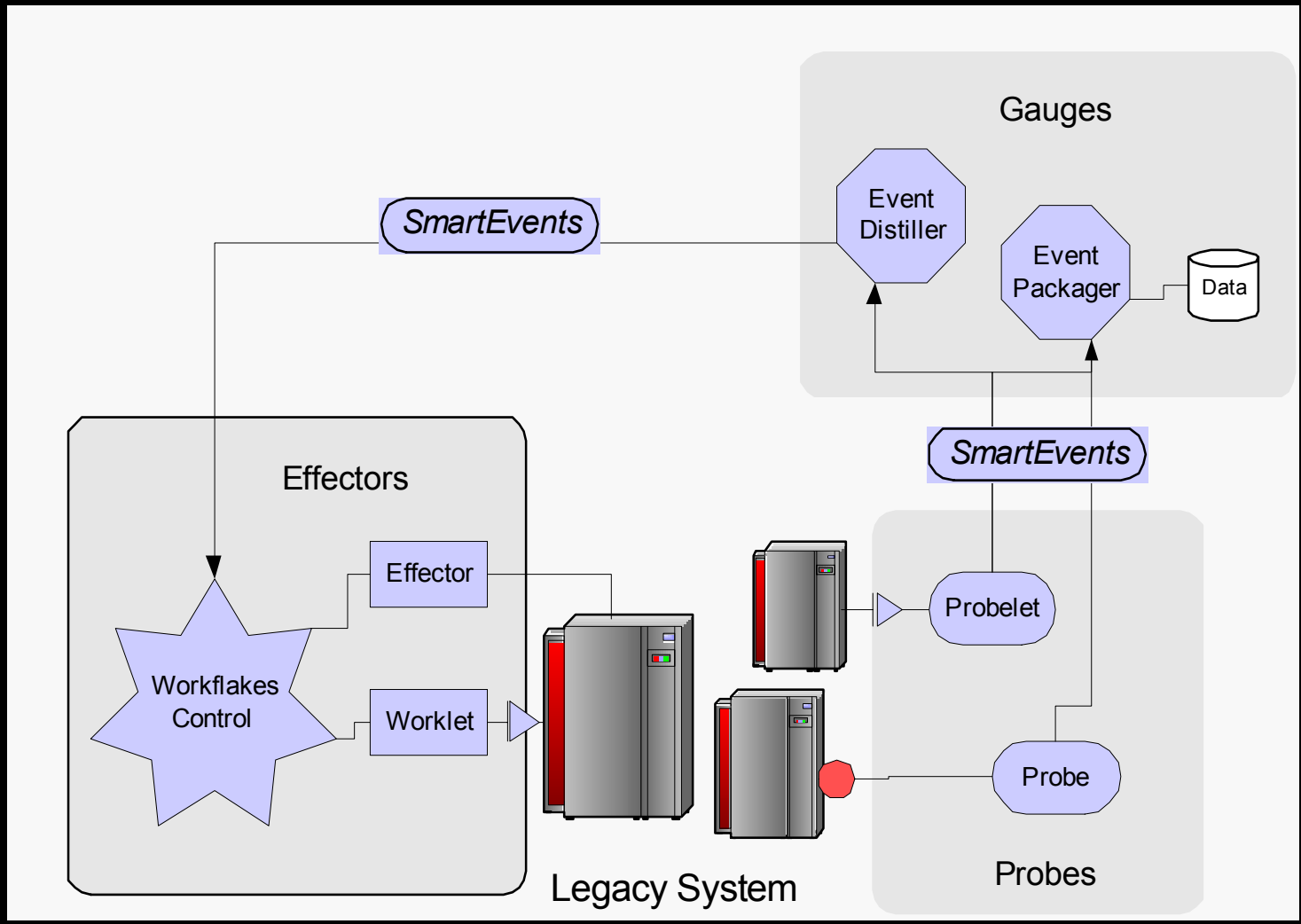
Results

- Beneficial impact on costs and responsiveness of service management
  - 50 to 90% optimizations
- Automation of adaptation decisions and actions provide tight control loop
  - eventually benefits perceived service quality
- Relatively little amount of code developed to adapt the system:
  - On top of the KX infrastructure code base

How we did it

- Our infrastrucure: KX (Kinesthetics eXtreme)
- Feedback control loop superimposed to the target system
  - External and orthogonal
  - To preserve independence and generality

KX Architecture



KX decision and coordination

- Decision on the basis of:
  - Gauges' reports
  - Codified model of the target system
- Upon decision – adaptation actions:
  - Multi-step process
  - Carried out by multiple effectors
  - That need coordination
- Effectors' coordination is automated by a workflow engine (Workflakes)

KX and Target System

- Two points of contact:
  - Probes
  - Effectors
- Require target instrumentation
- Numerous techniques possible
- Must be minimally intrusive

- The rest of the adaptation framework is detached from the target
  - Although needs to know a great deal about it

The issue of “self”

- As in “self”-healing
- Tension between built-in adaptation provisions and external adaptation infrastructure
- Both serve the same purpose
  - “self”-healing
- But carry numerous different conceptual and engineering implications

The case for an expanded “self”

- Applies better to legacy
- Promotes separation of concerns
- Retains generality
- Makes maintenance easier
- Can cooperate with and take advantage of any built-in techniques
- Can always be “built in” onto a new system

Relevant issues for externalized adaptation

- Requires formalization and explication of a system model
  - Can be complex and labour-intensive
  - Calls for "good SE practices"
- Repertoire and integration of probes and effectors are technological challenges
- Must reconcile heterogeneity
  - With "standard" protocols and APIs
- Calls for reliability guarantees on the external infrastructure itself

Final remarks

- What are trade-offs and limits of internal vs. external adaptation provisions?
- Which techniques are better suited for the internal vs. external approach?
- What categories of target systems can be optimally addressed by each, and what characterizes them?

enable (vt) : *to make possible, practical, or easy*

PROGRAMMING SYSTEMS LAB
COLUMBIA UNIVERSITY

http://www.psl.cs.columbia.edu/