Class 6
Model-based SHS

Shang-Wen Cheng
zensoul@cs.cmu.edu
Apr 2, 2003

Papers

- Gross, Gupta, Kaiser, Kc, Parekh (CDSA'01)
  - An Active Events Model for Systems Monitoring.
- Combs, Vagel (WOSS'02)
  - Adaptive Mirroring of System of Systems Architectures.
- Bond, Sud (CDSA'01)
  - Service Composition for Enterprise Programming.
    - How is this related to SHS?

Model-based Self-repair

- Main idea
  - Use some abstraction as basis for problem detection and adaptation
- Variations
  - "Layer" of system
  - Models used?
  - Externalized or internalized?
  - Support multiple control models?

Taxonomy?

- Application Domains
  - Networks, Distributed Systems
  - Mobile Systems
  - Ubiquitous Computing
  - Biology Simulation
  - User Interfaces
  - Collaborative Computing
  - Games
- Tools, Mechanisms, Techniques
  - Architectural models
  - Algorithms/code-based
  - Formal models
  - Genetic algorithms/alternative models
  - Agents
  - Economic theory
- Goals:
  - Improve system performance/Resource usage
  - Improve user experience; reduce user distractions
  - Improve dependability

Model-based adaptation

Typically, distributed collaborative applications

Models: events, agents, contracts, policy

System quality attributes such as performance, reliability, etc.

*Four General Requirements for SHS

- Monitor:
  Observe the running system and abstract observed behavior.
- Detection:
  Continuously check constraints via explicit(?) run-time models.
- Resolution:
  Determine the cause of constraint violation and choose a adaptation strategy.
- Adaptation:
  Adapt the system using verified change strategies.

ActEvents Approach

- Application domain
  - Distributed system with focus on "monitoring"
- What's the model?
  - "Active events"
  - Notion of anomaly and time window
- Goals
  - Flexibility and Efficiency
- Mechanism
  - XML & mobile code

- Monitor:
  - Use system of "probes" to get raw information out
  - Abstract that information using "gauges"
- Detection:
  - SmartEvent showing anomalous condition
- Resolution:
  - Is there any?
- Adaptation:
  - Deploy worklets

Adaptive Mirroring Approach

- Application domain
  - Enterprise, distributed applications
- What's the model?
  - "Adaptive mirror" & contracts
    - How is this different from an architectural model?
- Goals
  - Reliability and scalability
- Mechanism
  - Agents
- Monitor:
  - Probes & gauges
- Detection:
  - Constraint directive violation
  - Does "hint" play a part?
- Resolution:
  - Search of eligible service substitutes
- Adaptation:
  - Service substitution
    - How does it occur?

ODSI Approach

- Application domain
  - Enterprise applications, with focus on middleware
- What's the model?
  - Enterprise policy
  - Dynamic directory?
- Goals
  - Reliable & flexible interoperation
- Mechanism
  - Peers
- Monitor:
  - Implicit, when peers get service requests
- Detection:
  - A requested service is unavailable or violates enterprise policy
- Resolution:
  - Determine where to reroute or request service
- Adaptation:
  - Reroute service or request service provision

Shortcomings?

- Are models explicit and well-defined?
- How "reusable" are the components of the various approaches?

Comments?

- The End