Workshop on Self-managed Systems (WOSS’04)

October 31 - November, 2004

Newport Beach, CA
Workshop Logistics

- Sessions organized by affinity groups
  - 3-4 papers per 1.5 hour session
- Session chairs, as noted on program
  - See your chair before your session
- Emphasis on discussion
  - 15-minute research overviews from each paper
- Half-hour breaks between sessions
- Joint lunches
Software Engineering Today

- Common assumptions
  - Known and stable system requirements
  - Known and stable operating environment
  - Control over the development of assembled parts
  - Development time and run time are separate
  - Systems can be taken “down” for “maintenance”

- Consequences
  - Focus on improving development time processes, techniques, notations
  - Provide high assurance through testing, rigorous specification, modeling, verification, etc.
  - Manual installation and upgrades
Isn’t This Good Enough?

- Increasingly, systems
  - are composed of parts built by many organizations
  - must run continuously
  - operate in environments where resources change frequently
  - are used by mobile users

- For such systems, traditional methods break down
  - Exhaustive verification and testing not possible
  - Manual reconfiguration does not scale
  - Off-line repair and enhancement is not an option
What Has to Change?

- Goal: systems automatically and optimally adapt to handle
  - changes in user needs
  - variable resources
  - faults
  - mobility

But how?
A Unifying View?

Development Time

The System

Run Time

The System

Environment

The System

Environment
Many Approaches

- Architecture-based adaptation
- Programming language support
- Algorithms (e.g., self-stabilizing, machine learning)
- Operating systems support
- Domain-specific techniques (e.g., distributed databases, pub-sub architectures, …)
- Adaptable middleware
- Support for user mobility
- Fault tolerant system design (e.g., graceful degradation)
- Biologically-inspired models
Why Have a Workshop?

- Understand the relationships between these different approaches
- Identify the *software engineering* challenges and opportunities
- Create a common vocabulary (or possibly a reference model)
Affinity Groups

- Architecture-based adaptation
- Systems: operating systems, distributed systems, databases
- Techniques
- Everyday Systems
- Reasoning
- Patterns
- System-level adaptations
Common Questions

- What does "self-managed" mean to you?
- What aspects of the self-management problem are you dealing with?
- What aspects are you NOT dealing with?
- What applications are you targeting?
- What are the top two/three new technical ideas/approaches that you are pursuing in this work?
Common themes/issues from last time

- Internal versus external
- Distributed control versus centralized
- Generic versus application/domain-specific
- Level of adaptation
- Knowing that you have it “right”
- Compositionality and multi-dimensionality
Today’s Schedule

- **Architecture-based Adaptation**
  - Moderator: Alex Wolf

- **Distributed Systems**
  - Moderator: Jeff Kramer

- **System-level Adaptations**
  - Moderator: Rogerio de Lemos

- **Techniques**
  - Moderator: Jeff Magee