A synopsis of
Self-Managed Systems: an Architectural Challenge

Jeff Kramer and Jeff Magee
FOSE '07: 2007 Future of Software Engineering
Talk Agenda

• Overview
• Current Relevant Work
• Why Architectural Approach?
• Inspiration from Robotics
• Architectural Model Layers
• Research Issues
• Contribution and Conclusion
Overview

• Paper provides background for Autonomic Self-* systems
  – Self Configuring system should be able to re-configure or report if it can’t be done
  – Self-adaptation and Self-healing systems attempt to adapt to changed specification; much harder problem to solve

• Required specification includes functional as well as non-functional properties

• Many areas of research are included as reference
Current Relevant Work

• SelfMan workshop to discuss and analyze potential of Self-* systems
• Self-healing by dynamic fault awareness
• Self-adaptation for dealing with flash workloads
• Autonomic control loop – Collect, Analyze, Decide and Act
• Many conferences and workshops in the last several years
Why Architectural Approach?

- Generality - Applicable to large number of domains
- Abstraction – Use of components, binding rather than algorithm
- Scalability - Large scale complex applications
- Existing work – Provides rigorous approach
  - Provides a list of work in this field
Inspiration from Robotics

• Sense, Plan, Act (SPA) – for Robot designs – 80s architecture
  – Self-managed system is an autonomous system in exactly the same way as a Robot is.

• Now they follow GAT’s architecture since last 15 years or so
  – Control
  – Sequencing
  – Deliberation
Architectural Model – Control Layer

Control Layer – Bottom most layer

- Sensors, Actuators and Control Loops
- Reports current status to upper layer
- Creates, deletes and interconnects components
- Adjusts the operating parameters of components
- When confronted with situation not designed to handle, must report this condition to upper layers
Change Management Layer

Reacts to changes reported from bottom layer

Selects new control behaviors and new operational parameters

Can introduce new components, change interconnections, recreate failed components

Pre-specified plans activated in response to change from below

Situation for which no plan exists, has to be reported to the upper layer
Architectural Model – Goal Management

**Goal Management Layer**

Deliberation layer – Time consuming computations to plan to achieve a goal

Most complex layer – many research issues remain to be solved

Need to decide how general or domain specific this layer

Re-planning takes place when situation encountered is unknown

Introduction of new goals can produce change management which is the layer below
Research Issues – Control Layer

Safe application during algorithm change – avoid sharp transient behavior

Isolated components are easier to reconfigure than active components

Scalable Algorithms that minimize disruption

Verifying that safety property is not violated

One component could have multiple inner components – must deal with hierarchies of this nature
Research Issues – Change Mgmt. Layer

• Major research challenge is dealing with distribution and decentralization
  – Issues with latency, concurrency and partial failures
  – Local autonomy plus Global consistency

• Completely decentralized Architecture
  – Complete view of system
  – Total order broadcast required
  – Not scalable

• Relaxation of global structure constraints may open up some avenues
Research Issues – Goal Management Layer

• Precise specification of goals
  – Application and System goals
  – Both comprehensible by human and machine readable

• In the absence of precise specification, goal can be described
  – Search space is huge
  – Search time is unacceptably slow

• Offline problem planning and online verification
  – Online Planning is the harder one to solve – SASSY??
Contributions and Conclusion

• Architectural Level Self-Management
  – Components re-configure their interaction to be compatible with overall architecture spec.
  – Authors believe that this provides required level of abstraction and generality

• 3 layer reference model presented
  – Component Control
  – Change Management
  – Goal Management

• Authors want comprehensive solution lending itself to rigorous software approach