Panel #2: Can Management Systems Leverage Self-Organization?

Alexander Keller, alexk@us.ibm.com

The Autonomic Control Loop

Autonomic Manager

Diagnose
Delegate
Symptom

Assess
Delegate
Impact

Plan
Delegate
Action

Take
Delegate
Action

Sense
Delegate
Symptom

Analyze
Plan
Execute

Knowledge

Data
Action
Distributed Management vs. Self-organization

- "Smart" management system executes control/feedback loop
  - Interacts with "dumb" managed resources
  - If sub-optimal, configures them so that global optimum is achieved
  - e.g., Manager/Agent paradigm
  - How systems management is done today
- Every individual system runs its own MAPE-K loop
  - and, somehow, optimal behavior of the overall system emerges
  - e.g., Routing algorithms, event-driven I/O concurrency models
  - Smart resources, no management system needed (in principle)

These are indeed 2 fundamentally different viewpoints...

…and these 2 Viewpoints have been articulated long ago

- "What we see are merely reflections of perfect archetypes" (Plato)
  - Basically, the world is in equilibrium
  - If a system gets pushed outside of the equilibrium, push it back again
- "The world is in a constant state of flux" (Heraclitus)
  - The world is in a process of flow and change
  - Systems are interacting in a variety of combinations
- Variations of these viewpoints reflected in many CS paradigms
  - Manager/Agent model vs. peer-to-peer interactions
  - Implementing concurrent I/O with threads vs. events
  - Workflows vs. business objects
- 1985: The Santa Fe Institute - Complexity
  - Autocatalytic systems, Positive feedback and increasing returns ("lock-in"), Artificial Life
Demonstrating Value

- **Scientific community’s main focus areas:**
  - Optimization techniques
  - (Mathematical) system modeling and design
  - Rigorous evaluation of results through experiments
  - Achieving automation
- **Major headaches in practical IT service management:**
  - Availability and problem determination
  - Dealing with human error
    - More than 50% of service outages due to misconfiguration
    - Repeatable procedures for changing systems
    - More than 50% of service outages happen during maintenance windows
  - Labor Costs
  - “Good enough” solutions
  - Reluctance to immediately implement automation, based on past negative experience

Plenty of opportunities for the scientific community!
Challenges (1): Service Level Agreements

- **We know:**
  - how to express them (templates, SLA languages, e.g., WS-Agreement)
  - how to address key performance indicators (KPIs): Availability, response time, throughput, bandwidth...
  - how to monitor compliance: provider / customer / 3rd party measurement services
  - how to assign different cost/profit functions to different time periods

- **We don’t know:**
  - how to break down KPIs into measurable parameters
  - how to consolidate SLA parameters with other constraints
    - Cost/profit functions
  - how to address legal implications on where (corporate) data is stored
    - servers in many data centers are in 192.168 subnets

Challenges (2): Codify Best Practices

- **Development organizations:**
  - management costs viewed as support, not development costs
    - Core business: adding new functionality to products
    - Not: Making products easier to administer
  - Fortunately, this begins to change

- **IT organizations:**
  - Estimating the impact of a change is extremely difficult
  - Results in:
    - Setup of dedicated staging environment
    - Manually try out what works best (or at all)
    - Create IT run books detailing the procedures
  - Huge costs (equipment, personnel)
  - Takes up to 90 days for complex changes
  - Still no guarantee that the procedure will really work in production
  - Every IT organization reinvents the wheel

- **2 approaches to codifying best practices:**
  - as workflows with explicit control and data flows
  - as individual FSMs that communicate via message queues
Can we leverage Self-Organization? Takeaway Points

- Yes, BUT:
  - I doubt one can make self-organizing systems deterministic
    - These are diametrically opposed approaches
    - Can't be "a little bit pregnant"
  - Many customers don't care if a system is self-organizing or not
    - as long as it brings them *quantifiable* value ($$$)
    - as long as the system behavior remains predictable
    - as long as local optimum leads to global optimum
  - Automation will only happen gradually over time
    - Deal with humans, e.g., in system configuration
  - Need to put the parts together, think in a bigger context
    - Address the major pain points of real-life environments
    - Study the problems that yield the most "bang for the buck" first
  - Some focus areas:
    - IT Processes as a way to codify best practices
    - Service Level Agreements as prerequisite for automation