Towards Autonomic Computing

Ph.D. Thesis Defense

Alexander V. Konstantinou

DCC Laboratory
Computer Science Department
Columbia University
Autonomic Computing

Autonomic computing

- Important challenge of this decade (management over 70% TCO)
- Goal: self-configuring/optimizing/healing/protecting systems

Thesis contributions

- Peer-to-peer autonomic management architecture
- Language for embedding autonomic management functions at design time
- Change propagation model, language, and analysis
- Autonomic platform prototype implementation (released)
- Applications: security, service & user mobility, active networks
Autonomy as an Afterthought

Traditional management
- Man-in-the loop
- Knowledge is diffused
- Ad-hoc processes & architecture
- Unsafe & insecure

Challenges of Autonomy
- Knowledge distribution
- Processes & architecture to effect knowledge
- Safety & security
An Architecture for Autonomy
Peer-to-Peer Autonomic Mgmt Architecture

Two-layered approach:
- Data modeling layer
  - Unified object-relationship schema
  - Transactional access, event notification, persistence, security
- Autonomic management layer
  - Semantic schema extensions

Advantages:
- Scalability: unified model, multi-manager, publish-subscribe, cross-domain
- Reliability: safe access, synchronous policy enforcement, reduced complexity
A Language for Autonomy
Autonomic Element Instrumentation

Challenges:
- Element-agent sync., safe access, sync. control, efficient monitoring
- Adaptation to emerging software engineering approaches

**JSpoon:** extending Java with management features
- Management attributes, relationships, transactions & events
- Remote access, persistence & discovery
- Common element & manager language

![Diagram of JSpoon Runtime](image-url)
Autonomic Element Instrumentation

**JSpoon Runtime**
- Remote access, transactions, events, persistence
- Plug-in schema semantic extension

```
NtpServer
Socket sock
start()
short port
```

```
Java Element Obj.
```

```
NtpServer
getPort() : short
setPort(short)
```

```
Java View Object
```

```
Constraint
Knowledge
Plug-in
```

```
JSpoon Runtime
```

```
Remote Acc.
```

```
Persistence
Repository
```

```
JSpoon Compiler
```

```
JSpoon Object
```

```
port < 1024
```

```
Schema Ext.
```

```
Constraint
Knowledge
Plug-in
```

```
JSpoon Runtime
```

```
Java Element Obj.
```

```
Java View Object
```

```
JavaScript
```

```
Runnable
```

```
Remote Acc.
```

```
Persistence
Repository
```

```
Constraint
Knowledge
Plug-in
```

2003-09-25  Alexander V. Konstantinou
public class NtpServer extends Thread {

    protected DatagramSocket sock;

    config boolean enabled = true;

    instrument counter long reqCount = 0;

    relationship timeSource, TimeSource, serves;

    public void run() {
        while(enabled) {
            sock.receive(packet);
            reqCount++;
        }

        subscribe !srv.enabled {
            srv.reqCount = 0;
        }

        atomic(timeout) {
            if (!srv.enabled) ...
        }
    }
}
JSpoon Management Events

Challenge:
- How to extend element behavior

JSpoon synchronous events
- Notification in transaction context
- Generalized exception mechanism

JSpoon asynchronous events
- Efficient monitoring
- Ex.: utilization > 0.9 over 30000

JSpoon schema extensions
- Plug-in event handlers
- Constraints, change prop., event correl.

```java
try {
    setPort(321);
} catch (Exception e) {
    // recovery
}
```

Traditional Exceptions

```
port = 321;

subscribe NtpServer on port < 1024 {
    if (user != root) {
        abort;
    }
}
```

Generalized Exceptions
Effecting Change Propagation
Effecting Change Propagation

Challenges:
- Current approaches: scripting, constraint satisfaction
- Termination, safety, deterministic behavior, bounding change, composition

A spreadsheet model of change propagation
- Object attribute ~ cell, relationship ~ relative location
- Change rules are inherited attributes of objects
- Disallow cycles, ambiguities

Example:
- UDP-based web-radio application packet size configuration

```
packetSize := servedBy.linkedVia.mtu - 68
```
Spreadsheet model

- $s: y \leftarrow f(x_1, x_2, \ldots, x_n)$
- Target(s) = $y$, Trigger(s) = \{ $x_1, x_2, \ldots, x_n$\}
- Cycle: $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_1$

Spreadsheet rules

- Defined over schema
- Evaluated over instantiation

Static analysis

- Over schema graph

Execution model

- Attribute-set, relationship-set, object-create/remove

<table>
<thead>
<tr>
<th>0</th>
<th>1 IF($A3,$A2,1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 IF($A4,2,$A1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WebRadio</th>
<th>servedBy</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>short packetSize</td>
<td></td>
<td>name=&quot;www&quot;</td>
</tr>
<tr>
<td>WebRadio</td>
<td>packetSize=508</td>
<td>servedBy</td>
</tr>
<tr>
<td>string name</td>
<td></td>
<td>name=&quot;ftp&quot;</td>
</tr>
<tr>
<td>WebRadio</td>
<td>packetSize=128</td>
<td>servedBy</td>
</tr>
</tbody>
</table>
Object Spreadsheet Language (OSL)

- **Assignment**
  - `object-field := functional-expression`

- **Relationship navigation**
  - To-one → instance, to-many → collection

- **Operations**
  - Arithmetic, boolean, first-order
  - Missing: unbounded looping, recursion

- **Object & relationship creation**

- **Scaling rule development**
  - Management functions

- **Syntax**
  - Smalltalk, UML Object Constraint Language

**Syntax**

- `OSL0`
  - *Arithmetic*
    - `=, +, -, *, /, <, <=, >, >=`

- `OSL0.5`
  - *Propositional*
    - `if-then-else, and, or, xor`

- `OSL1`
  - *First-Order Expressions*
    - `iterate, forAll, exists, select, allInstances`
**OSL at a Glance**

**Assignment**

**To-one navigation**

**To-many navigation**

**Relationship operations**

**Management functions**

**context Application:**

active := servedBy.active default false

**context WebRadio:**

packetSize := servedBy.connectedVia
  ->select(not loopback)
  ->collect(mtu)
  ->min(1500)

**context NetworkHost:**

defun isConnected() : boolean =
  connectedVia->select
  ( (not loopback) and
    (connectedVia.state = UP) )
  ->size() <> 0
OSL Triggering Graph

Triggering graph (directed)
- Nodes: attributes & relationships
- Edges: trigger → target

Propagator
- Edge label identifying dependency path

```
context WebRadio:
   packetSize := servedBy.linkedVia.mtu - 68 default 506
```

```
WebRadio. packetSize
   \arrow[thick, ->]{this} \arrow[thick, ->]{serves} WebRadio. servedBy / Host. serves
   \arrow[thick, ->]{serves} Host. linkedVia / LinkInterface. links
   \arrow[thick, ->]{linkedVia mtu} LinkInterface. mtu
```

“.” (dot)

servedBy

“.” (dot)

linkedVia

mtu
OSL₀ Rule-Set Evaluation

**OSL₀ Termination:**
- Set of rules contains cycle iff triggering graph contains cycle

**Rule rank:** Target(r) node order in topological sort
- Evaluation algorithm complexity $O(i)$

**Pending Rules**

$$z := y + x$$

$$y := x$$

**Changed Values**

$$x$$

$$y$$

$$z$$

**Instance selection:**
- Use propagator to select effected instances

**OSL₀.5 analysis**
- Cycle may not lead to infinite execution, if propagators not satisfiable
Cross-Domain Autonomy

**Challenges:**
- Detect & control cross-domain propagation
- Scale cross-domain rule analysis

**Summary triggering graph**
- Export border objects to summary domain
- Summarize triggering dependencies

Challenges:
- Detect & control cross-domain propagation
- Scale cross-domain rule analysis

Summary triggering graph:
- Export border objects to summary domain
- Summarize triggering dependencies
Prototype & Applications
NESTOR Browser Snapshots

NESTOR Topology 2.4.0 - Columbia University

Network: Nestor-Demo.cs.columbia.edu
- System Name: Nestor-Demo.cs.columbia.edu
- Bridge Type: false
- Contact: Alexander V. Konstantinou
- IP Chains: input@nestor-democ.s.columbia.edu
- IP Stack: IP@nestor-democ.s.columbia.edu
- Link Interfaces: eth0@nestor-democ.s.columbia.edu
- Location: View
- Memory Size: 261658272
- Network Interfaces: 129.59.22.223@nestor-democ.s.columbia.edu
- OS Name: Linux
- OS Release: 2.4.16-10mpm
- Packages: mailcap@nestor-democ.s.columbia.edu
- Processors: cpu[0]@nestor-democ.s.columbia.edu
- Request Reboot: View
- Request Shutdown: View
- Service Table: services@nestor-democ.s.columbia.edu

NESTOR Browser v2.4.0 - Columbia University
Browser Repository Transaction Topology Help

- Network: Network Interface
- Service Naming: BindDNS Zone
- Service Naming: Domain Name Server
- Service Naming: DNS Address RR
- Service Naming: DNS Common Name RR
- Service Naming: DNS Host Name RR
- Service Naming: DNS Mail Exchange RR
- Service Naming: DNS Name Server RR
- Service Naming: DNS Resource Record
- Service Naming: DNS Zone
- Service Naming: Domain Name Server
- Service Naming: Nestor Constraint
- Service Name: Nestor Demo Containment
- Service Name: Nestor Demo Containment
- Service Name: Nestor Demo Containment
- System Host: Nestor-Demo.cs.columbia.edu
- System Host: www.nestor.cs.columbia.edu
- System Host: Host CPU
- System Host: Host Process

2003-09-25 Alexander V. Konstantinou
Enforcing Security Policies (Telcordia)

**Challenge:** Expressing & enforcing domain-wide security policies

- Example: don’t allow telnet from Internet

**NESTOR-based solution**

- Express security policies using declarative language (Prolog)
- Compile policies into a MODEL configuration model
- Policy monitoring & enforcement using NESTOR
Impact

Publications
- USENIX’99, JSAC’00, DANCE’02, AMS’03

Applications
- DNS/DHCP integration (DARPA 1997)
- Dynamic security (USENIX 1999)
- Active multimedia QoS (DARPA 2000)
- Distributed firewall (Telcordia 2001)
- Active Networks management (DARPA 2001)
- Web-server mobility (DARPA 2002)

Technology Transfer
- Telcordia Technologies: Smart Firewalls
- UCLA/UCB/Utah (DARPA ANETS): Adaptive multimedia
Conclusions & Future Work

What will it take to create autonomic systems?
- Standardization of instrumentation technologies
- Analyzable change propagation
- New operational procedures

Thesis contribution
- Cut-through approach to the technology issues to prove feasibility

Future work
- Scaling development of change propagation models
- Handling the dynamics of change propagation
- Managing the autonomic management layer