A Conflict Resolution Control Architecture for Self-Adaptive Software

Prof. A. Taleb-Bendiab
School of Computing and Mathematical Sciences
Liverpool John Moores University
{cmsnbadr, d.reilly, a.talebbendiab}@livjm.ac.uk
http://www.cms.livjm.ac.uk/except
Dependable software

• Autonomic computing: a recent trend
  – Devolving software management, maintenance to software
    • Self-organising, self-healing, sentient, self-adaptive, self-aware, etc.
  – Requiring meta-systems and meta-reasoning to;
    • Continuous measurement and/or reflection on operational systems

• High-assurance: high-{integrity, availability, etc.}
  – Complexity and uncertainty hiding through;
    • adaptive capability to respond to changes including: fault & intrusion-tolerance, thus masking errors, failures, etc.
    • Dynamic architecture transformation and reconfiguration strategy;
      – This requires reasoning and consideration of a set of concerns;
        » software architecture model including; components and their interactions, the properties and policies,
        » Style and composition rules and/or norms that limit the allowable systems adaptation operations.
Integrity Management

• Dynamic architecture transformation often lead to inconsistencies and conflicts
  – Systems integrity
  – Quality of service, etc.

• Requirement for a software adaptation engine with;
  – Conflict detection and identification
  – Conflict resolution
    • Solution generation, negotiation
    • Change plan enactment, etc.
  – Control strategies defining;
    • Transformation rules, regulations, patterns, etc.

• Our approach is a middleware to support for self-adaptive software conflict management.
Related Work

• Self-Adaptive Software
  – Can be defined as software with computational reasoning capabilities to monitor and change its own structure and/or behaviour to adapt to its operating environment and recover from errors.

• Reflective middleware

• Dynamic configuration control and management

• Conflict resolution
  – Negotiation Protocol
  – Exception handling
# Reference Model #1

<table>
<thead>
<tr>
<th>System Identifier</th>
<th>System Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>System One (S1) Operations</td>
<td>System One performs the productive operations of the organization. An organization may be composed of a number of S1s, each providing a distinct product or service. Each S1 consists of an operational element controlled by a management process and in contact with the operational environment and is in some respects similar to the plant/management arrangement adopted by control system theory.</td>
</tr>
<tr>
<td>System Two (S2) Coordination</td>
<td>System Two is concerned with the coordination of activities within this system which will be involved in the production, although still not involved with the environment in any significant way. The management processes contained within the control system will be concerned with activities of a different kind than in the normally accepted sense.</td>
</tr>
<tr>
<td>System Three (S3) Audit</td>
<td>System Three (read also as System Four) is concerned with planning the way ahead in the light of external environmental changes and internal organizational capabilities. To this end, S4 ‘scans’ the environment for trends that may be either beneficial or detrimental to the organization and constructs developmental organizational plans accordingly. To ensure that such plans are grounded in an accurate appreciation of the current organization, the intelligence function contains an up-to-date model of organizational capability.</td>
</tr>
<tr>
<td>System Four (S4) Intelligence</td>
<td>System Four is concerned with planning the way ahead in the light of external environmental changes and internal organizational capabilities. To this end, S4 ‘scans’ the environment for trends that may be either beneficial or detrimental to the organization and constructs developmental organizational plans accordingly. To ensure that such plans are grounded in an accurate appreciation of the current organization, the intelligence function contains an up-to-date model of organizational capability.</td>
</tr>
<tr>
<td>System Five (S5) Policy</td>
<td>System Five determines the overall purpose of the organization i.e. defines the activities that are performed by S1s. As such S5 represents the policy-formulation or normative planning function. Policy formulation is informed by a “world-view” provided by S4 and representing the current beliefs and assumptions held by the system about the environment and models of current organizational capability populated by data flowing from the lower level systems in the organization.</td>
</tr>
</tbody>
</table>

**Key**

- $V$ = Variety
- $\wedge$ = Attenuation
- $\triangle$ = Amplification

---

A. Taleb-Bendiab, ICSE’02, Workshop on Architecture for Dependable Systems, Orlando, 200999, Pages: 5
An Architectural Model
An Example: E-Fire Services
Example Programming Model
Demonstrator

Ad-hoc service assembly tool

Architecture transformation tool

Software instrumentation tool
Conclusions & Future Work

• Presented an architecture for conflict resolution and management for
  – Self-adaptive software
  – Supplied as a middleware service

• Presented an example illustrating;
  – Propose programming model
  – Usage model

• Further work
  – Resolution session control and management
  – Evaluation.
That’s the end – so I’m off!