Agenda

- Introduction
- Quality Criteria for self-healing systems
- Traditional quality attributes
- Autonomic-specific quality attributes
- Architectural Styles for self-healing systems
- Attribute-based architectural styles for self-healing systems
- Traditional quality ABAS
- Autonomic specific quality ABAS
- Use of ABASs in analysis
Traditional and Autonomic Specific Quality Attributes
Architectural Styles for self-healing systems

- Basic Requirements:
  - Reflection Mechanism
  - Reasoning Mechanism
  - Configuration Mechanism
Autonomic element

- Building Block
- Autonomic manager and components
- Control Loop – Monitor, analyzer, planner, executor
Self-Healing in terms of AC
H.P. Aspect peer-to-peer

Figure 4. Aspect peer-to-peer architectural style
Figure 5. Aggregator-escalator-peer architectural style
H.P. Chain-of-configuration

Figure 6. Chain-of-configurators architecture style
Attribute-based architectural styles (ABAS)

- Reasoning framework for evaluation
- Problem description
- Stimulus Response
- Architectural Style
Modifiability ABAS

Figure 7. Characterization of the modifiability ABAS
## Modifiability ABAS Table

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Peer-to-peer</th>
<th>Aggregator-escalator</th>
<th>Chain-of-Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Hidden Implementation details</td>
<td>Some Visibility</td>
<td>Some Visibility</td>
</tr>
<tr>
<td>Stimulus / Response</td>
<td>Peer layer change / # of peers, components, connectors</td>
<td>1:1 monitor change 1:1 configurator</td>
<td>1:1 monitor change Configurator change no impact</td>
</tr>
<tr>
<td>Architectural Style</td>
<td>Independent</td>
<td>dependent</td>
<td>Somewhat dependent</td>
</tr>
<tr>
<td>Parameters</td>
<td>No connectivity</td>
<td>Connected</td>
<td>Meddle Ground</td>
</tr>
<tr>
<td>Analysis</td>
<td>Add/delete Monitor / Configurator</td>
<td>Add / Delete -&gt; Change in Monitor / Configurator</td>
<td>Only monitor requires change in other monitors</td>
</tr>
</tbody>
</table>

2/12/13
## Modifiability ABAS Table (Cont.)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Peer-to-peer</th>
<th>Aggregator-escalator</th>
<th>Chain-of-Configuration</th>
</tr>
</thead>
</table>
| Design Heuristics        | • Not scalable  
• Single monitor visibility  
• Low coupling  
• Highly Modifiable  
• Low Efficiency  
• No Runtime Modification | • Full monitor output  
• Coupling -> low modifiability  
• Added layer -> performance penalty  
• No Runtime Modification | • Supports Runtime Modification from a set of candidates        |
Autonomic-specific ABAS

Figure 8. Characterization of support for detecting anomalous system behaviour ABAS
## Anomalous System Detection ABAS Table

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Peer-to-peer</th>
<th>Aggregator-escalator</th>
<th>Chain-of-Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Architecture Support?</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td>Stimulus /Response</td>
<td>Fault -&gt; Detection Rate, time, coupling, Awareness, Observability, fault model</td>
<td>Same</td>
<td>Same, Omitted</td>
</tr>
<tr>
<td>Architectural Style</td>
<td>Peer-to-peer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Analysis</td>
<td>- Coupling</td>
<td>+ Coupling</td>
<td>+ Enhanced Coupling</td>
</tr>
<tr>
<td></td>
<td>- Complexity</td>
<td>+ Complexity</td>
<td>- Complexity</td>
</tr>
<tr>
<td></td>
<td>- Awareness ( local )</td>
<td>+ Awareness</td>
<td>+ Awareness</td>
</tr>
<tr>
<td></td>
<td>- Observability</td>
<td>- Observability</td>
<td>+ Observability</td>
</tr>
<tr>
<td>Attribute</td>
<td>Peer-to-peer</td>
<td>Aggregator-escalator</td>
<td>Chain-of-Configuration</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------</td>
<td>----------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Design Heuristics</td>
<td>• Instrumenting: Static Code Dynamic Probes</td>
<td>Same</td>
<td>Same</td>
</tr>
<tr>
<td></td>
<td>• Testability</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Hard-wired requirements</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ABAS evaluation of Non-stopping Java Server

- Self-configuration, Self-Healing, Self-Managed
- Components: analyzers, policy manager, healing manager, healers, and policies
- Evaluation of: Modifiability, anomalous event detection, & failure diagnostics
- Style: Chain of Configurators
- Applicable ABAS: Chain-of-Configurators support for modifiability and detecting anomalous behavior
- ABAS used as template
Comments

• Assertions based on this model are based on abstractions vulnerable to discussion.