Quality Criteria and an Analysis Framework for Self-Healing Systems

Sangeeta Neti Hausi A. Muller

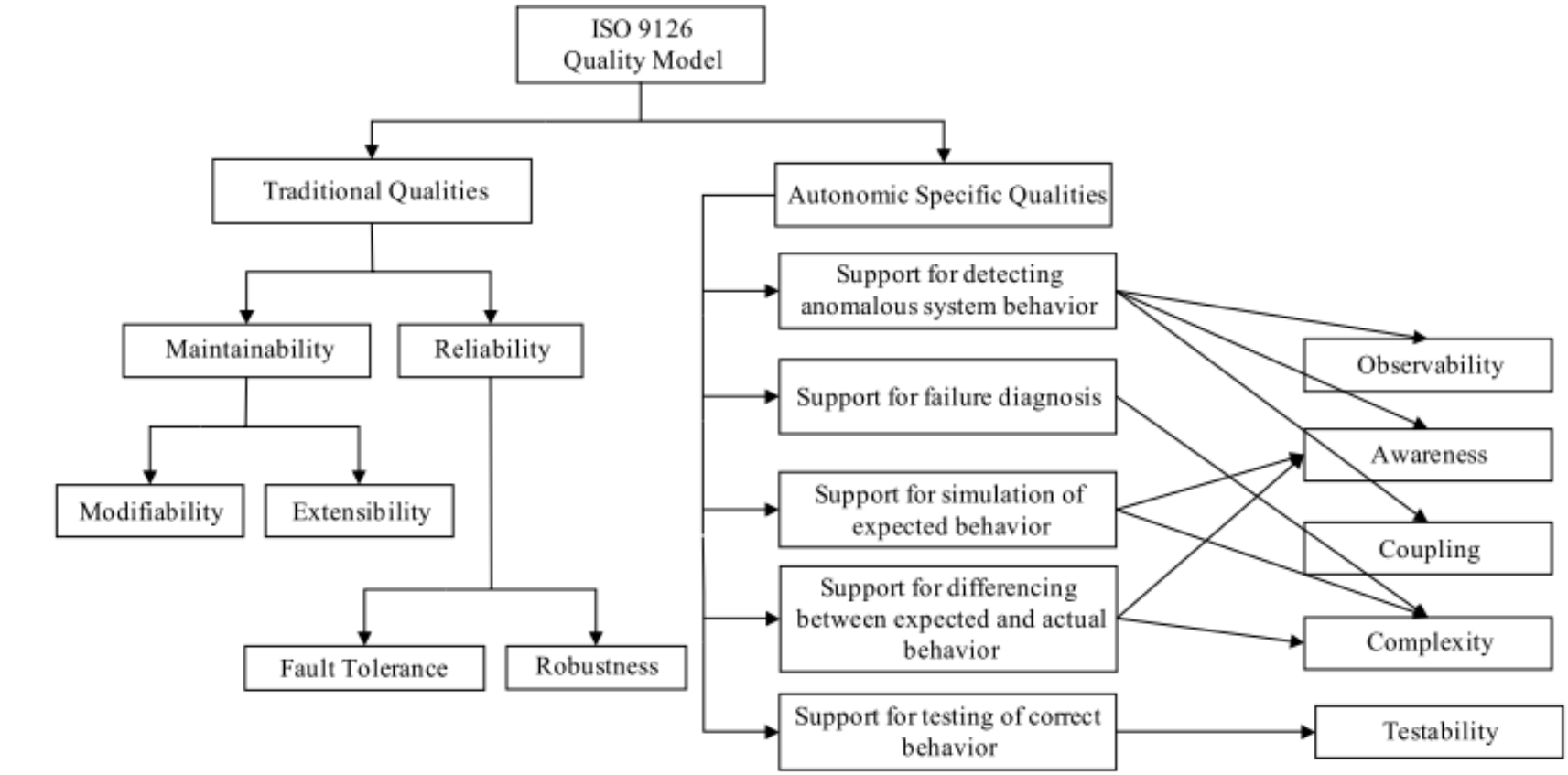
Proceedings of the 2007 International Workshop on Software Engineering g for Adaptive and Self-Managing Systems

Summarized by: Sean (Shahin)M. Ansari

Agenda

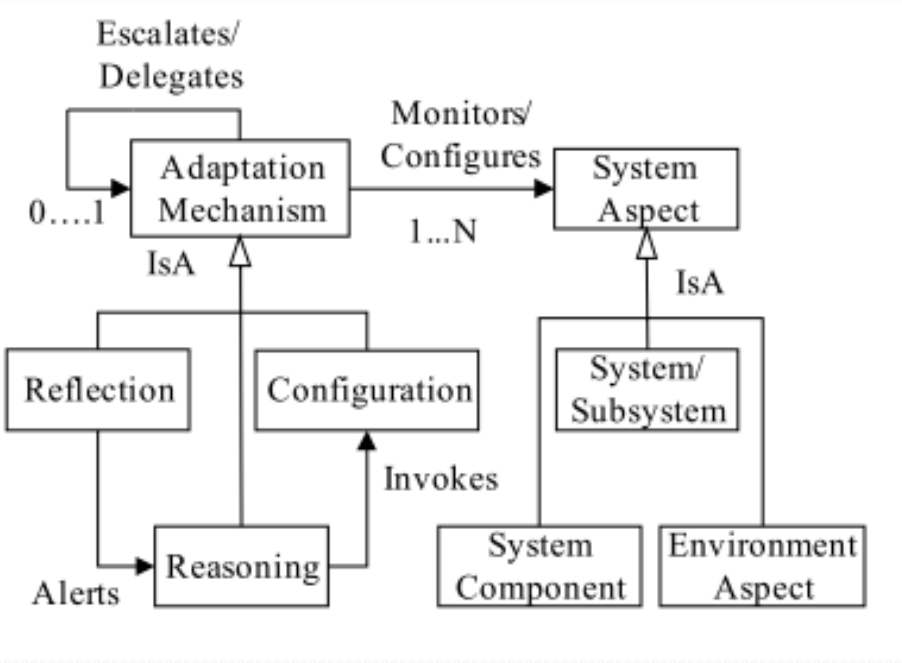
- Introduction
- Quality Criteria for self-healing systems
- Traditional quality attributes
- Autonomic-specific quality attributes
- Architectural Styles for self-healing systems
- Attribute-based architectural styles for self-healing systems
- Traditional quality ABAS
- Autonomic specific quality ABAS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

Traditional and Autonomic Specific Quality Attributes



Architectural Styles for self-healing systems

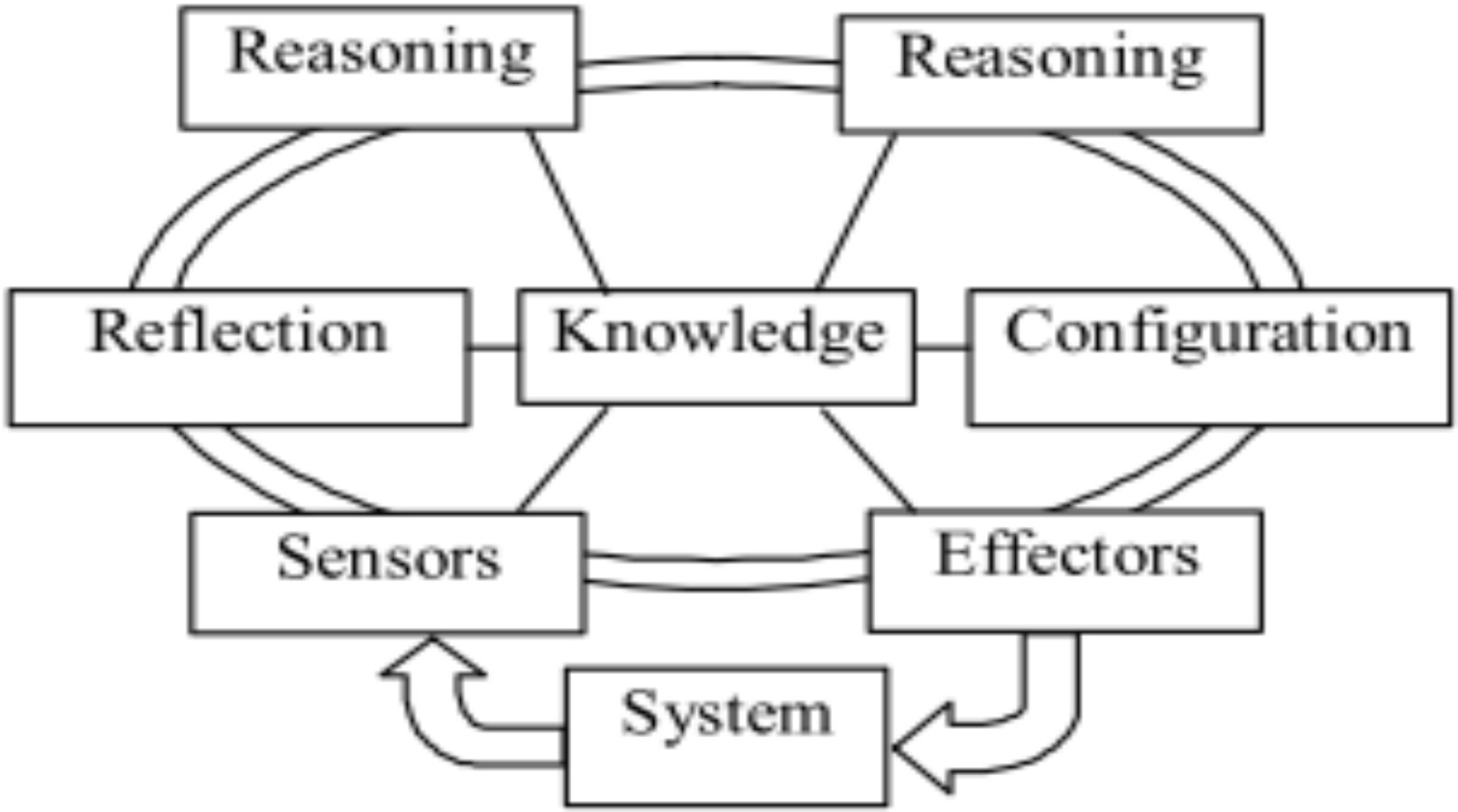
- Basic Requirements:
  - Reflection Mechanism
  - Reasoning Mechanism
  - Configuration Mechanism



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

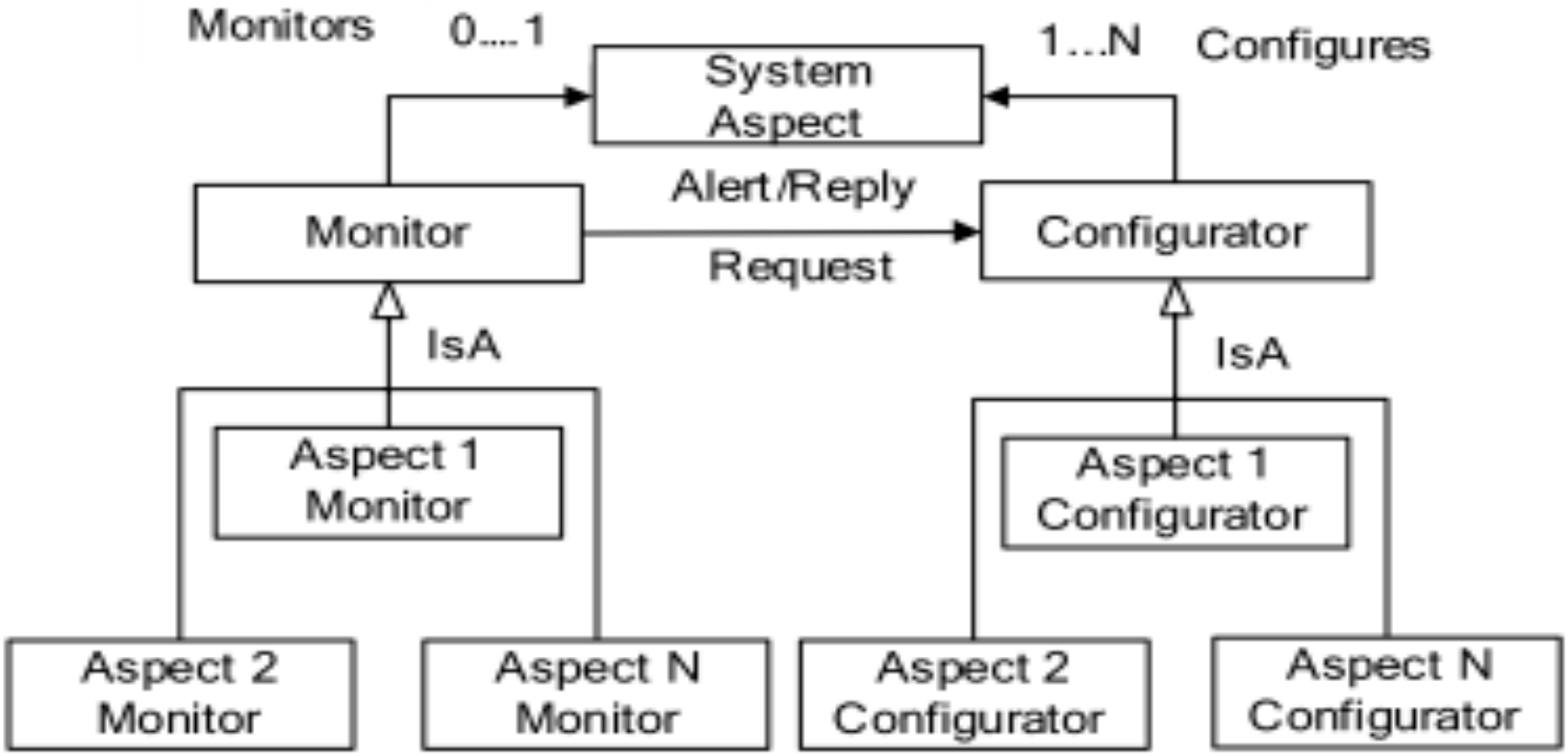


Figure 4. Aspect peer-to-peer architectural style

H.P. Aggregator-escalator-peer

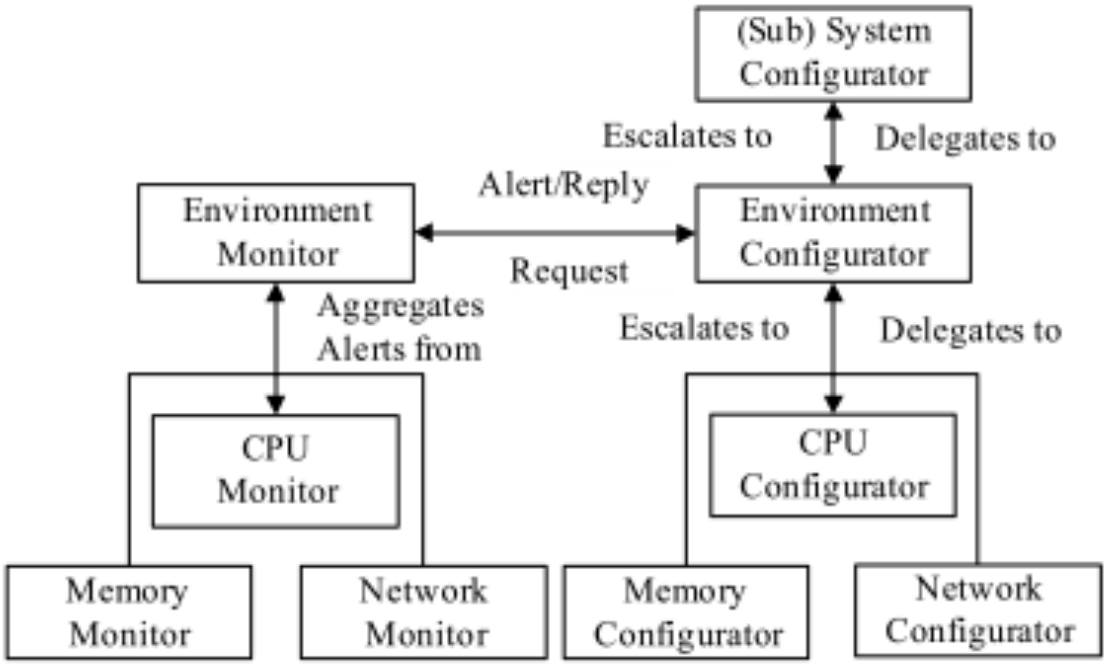


Figure 5. Aggregator-escalator-peer architectural style

H.P. Chain-of-configuration

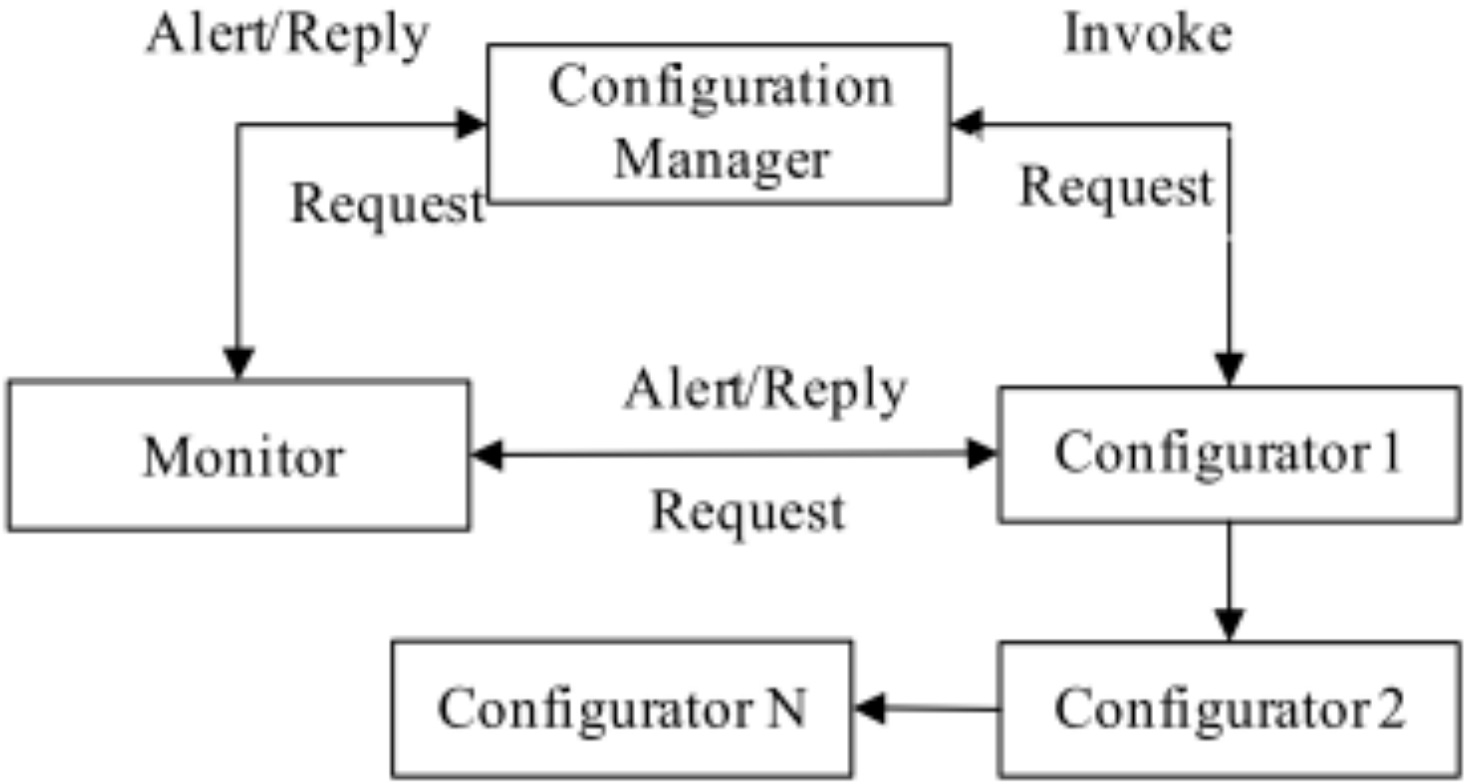


Figure 6. Chain-of-configurators architecture style

Attribute-based architectural styles (ABAS)

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

Modifiability ABAS

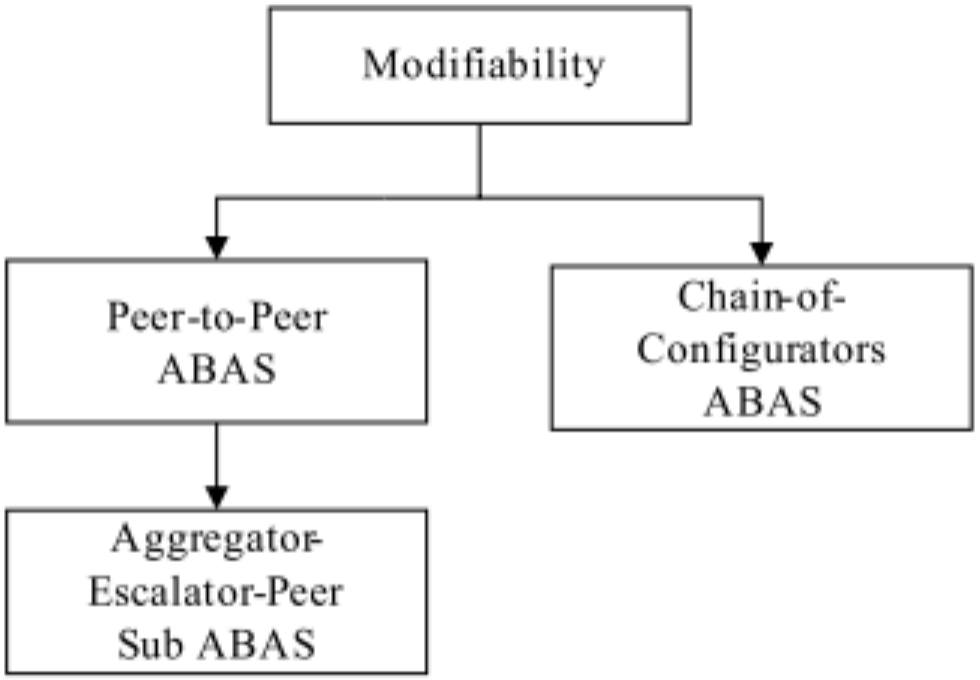


Figure 7. Characterization of the modifiability ABAS

Modifiability ABAS Table

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

Modifiability ABAS Table (Cont.)

Attribute	Peer-to-peer	Aggregator-escalator	Chain-of-Configuration
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

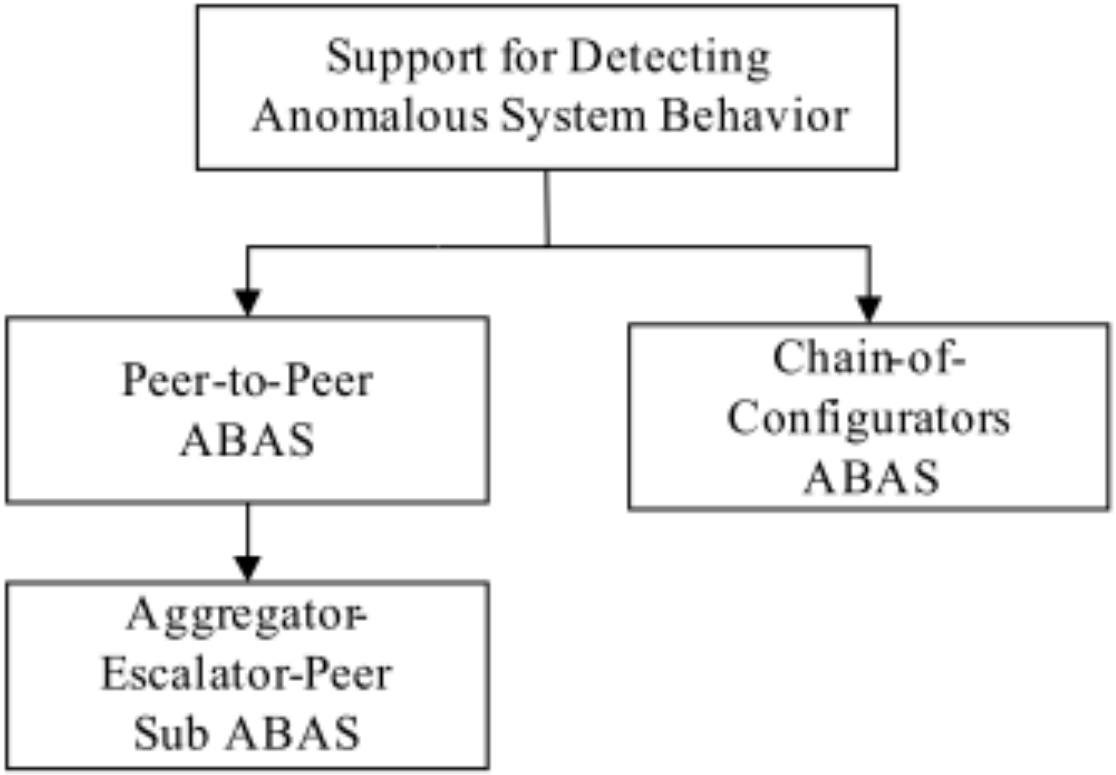


Figure 8. Characterization of support for detecting anomalous system behaviour ABAS

Anomalous System Detection ABAS Table

Attribute	Peer-to-peer	Aggregator-escalator	Chain-of-Configuration
Description	Architecture Support?	Same	Same
Stimulus /Response	Fault -> Detection Rate, time, coupling, Awareness, Observability, fault model	Same	Same, Omitted
Architectural Style	Peer-to-peer	Aggregator	Chain-of-Config
Parameters	-	-	-
Analysis	- Coupling - Complexity - Awareness (local) - Observability	+ Coupling +Complexity + Awareness - Observability	+ Enhanced Coupling - Complexity + Awareness + Observability

Anomalous System Detection ABAS Table

Attribute	Peer-to-peer	Aggregator-escalator	Chain-of-Configuration
Design Heuristics	• Instrumenting: Static Code Dynamic Probes • Testability • Hard-wired requirements	Same	Same

ABAS evaluation of Non-stopping Java Server

- Self-configuration, Self-Healing, Self-Managed
- Components: analyzers, policy manger, 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.