

Self-Adaptive Software: Landscape and Research Challenges

Salehie, Mazeiar and Tahvildari, Ladan, ACM Trans.
Auton. Adapt. Syst., Vol. 2, No. 4, 2009, pp. 1-42

Summarized by:

Mohan Krishnamoorthy

Overview

- Concept
- Principles
- Taxonomy
- Research Landscape
- Research Challenges

Self-Adaptive Systems

- Closed-loop systems with feedback from itself and context
- Evaluates its own behavior
- Changes or modifies behavior
 - Runtime
 - Functional
 - Non-functional

Hierarchy of the self-* Properties

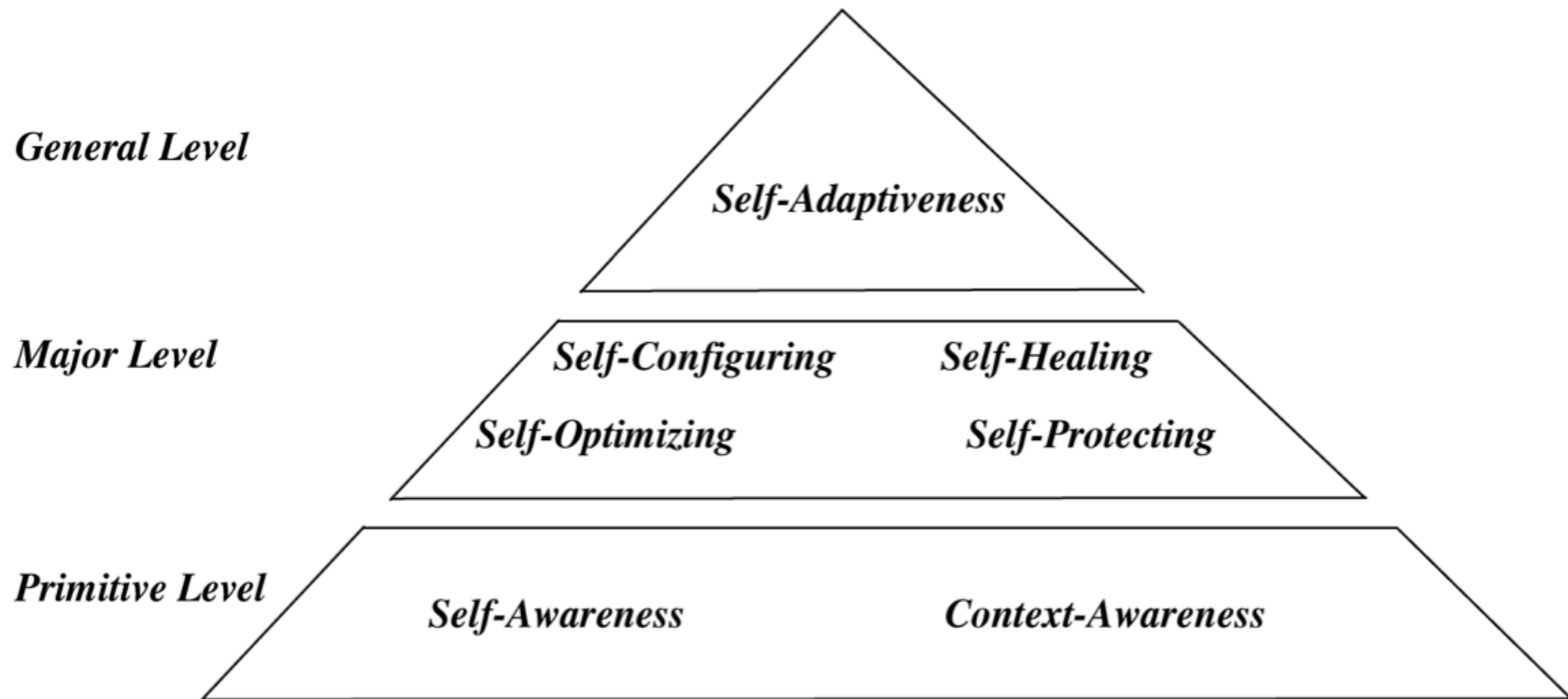


Figure 1 of Salehie et al. Paper

Adaptation Requirements Capturing

- Where
- When
- What
- Why
- Who
- How

An Example

- Client Server system
 - Each client belongs to a particular group of servers
 - Each server group has some bandwidth
 - Response time dependent on load and bandwidth
 - Maximize response time
- Adaptation Requirement capturing
 - Where: Client server communication
 - When: Response time is more than max allowed
 - What: Decrease load or increase bandwidth
 - Why: Keep the response time within the desired SLA
 - How: Add server to server group or change server group
 - Who: Manually or client code or client adaptation engine

Adaptation Processes

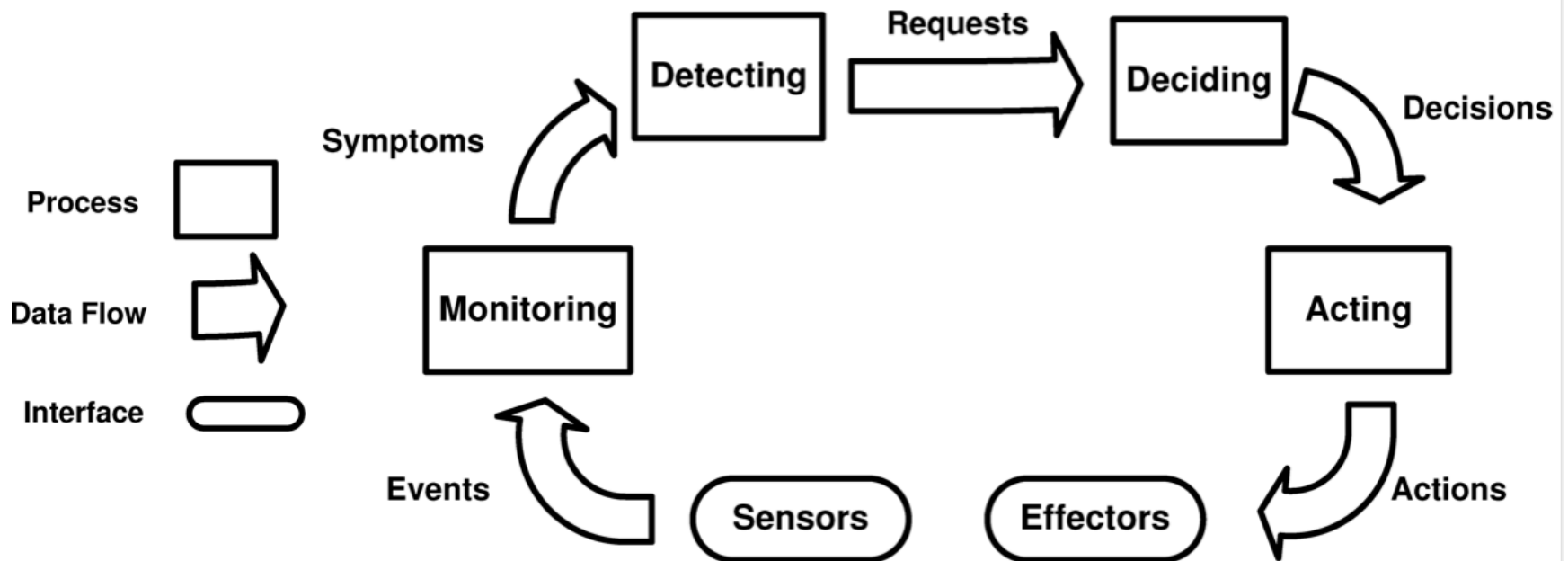


Figure 2 of Salehie et al. Paper

Adaptation Processes For Our Example

- Sensors: Logging or utility function
- Monitoring: Examining logs or utility to check response time against the SLAs
- Detecting: Response time crosses the threshold
- Deciding: If server is available, add server to the server group. Otherwise find a new server group
- Acting: Find an appropriate effector to apply actions and act
- Effectors: Function pointers or middleware based

Self-Adaptation Taxonomy I

Object to adapt (What/Where needs to change)

- Layer: Application or middleware?
- Artifacts: Objects or modules or subsystems
- Granularity: Fine level or coarse level
- Impact: Low or high aftereffects
- Cost: Low or high execution time, resource possession

Self-Adaptation Taxonomy II

Realization Issues (How to change): Approach

- Static DM: Hardcoded DM
- Dynamic DM: Policies or QoS definitions
- Internal Adaptation: Conditional expressions, parametrization, exceptions
- External Adaptation: External adaptation engine
- Making: Design/develop self-adaptivity
- Achieving: Learning self-adaptivity

Self-Adaptation Taxonomy III

Realization Issues (How to change): Types

- Close Adaptation: Static set of actions applied
- Open Adaptation: Actions added at runtime
- Free Adaptation: Requirements, goals used
- Model Based Adaptation: System model used
- Specific Adaptation: For a single domain
- Generic Adaptation: For multiple domains

Self-Adaptation Taxonomy IV

Temporal Characteristics (When to change)

- Reactive Adaptation: Respond to change
- Proactive Adaptation: Predict change
- Continuous Monitoring: Collects all data
- Adaptive Monitoring: Collects specific data in case of an anomaly

Self-Adaptation Taxonomy V

Interaction Concerns (Who will change)

- Human Involvement
 - Level of autonomicity
 - Interface with users and administrators
- Trust
 - Security
 - Assurance and Dependability
- Interoperability Support

Research Landscape

- Software Engineering
 - Architectures, Models, SOA
- Artificial Intelligence
 - Planning, Multi-Agent Systems, ML, Utility
- Network/Distributed computing
 - Policy and resource management, P2P
- Control Theory

Research Challenges I

- Translate software expectations to the adaptation actions requirements
- Design adaptable software from scratch or for legacy systems or for large scale systems
- Implementing these frameworks
- Testing these frameworks
- Maintaining the quality of adaptation

Research Challenges II

- Feature to monitor or the amount of data to collect
- Detecting which system are healthy and which are not
- Deciding on a less than optimal solution or deciding based on less than perfect measurement
- Coordination and organization of systems of systems

Research Challenges III

- What about?
 - Privacy issues when dealing with the monitored data
 - Policy of how to deal with sensitive data
 - Not all systems are transparent
 - Not all systems are open to applied change
 - Space complexity and time complexity of processing monitored data for large scale systems

Research Challenges IV

- What about?
 - Detecting false positives
 - Detecting an intruder in the system
 - Deciding the course of action if a change does not result in positive measurement
 - Standardizing the language of communication among the self-adaptive systems