A survey of Autonomic Computing — degrees, models and applications

MARKUS C. HUEBSCHER,
JULIE A. MCCANN

Summarized by David Gonzalez

Overview

- Autonomic Computing “definition” and properties.
- Chronology
- Autonomic element breakdown
- MAPE-K Loop Model
- Degrees of Autonomicity.
- Emerging fields
- Conclusions

What does Autonomic Computing (AC) means?

- Autonomic comes from Biology (e.g. Nervous system).
- AC seeks improving systems by decreasing human involvement.
- There is an active debate on this.
- Key: focus on the dynamism of the system.

Chronology

SAS Situational Awareness System	1997	DARPA	Decentralised self-adaptive (ad-hoc) wireless network of mobile nodes that adapt routing to the changing topology of nodes and adapt communication frequency and bandwidth to environmental and node topology conditions.
DASADA Dynamic Assembly for Systems Adaptability, Dependability, and Assurance	2000	DARPA	Introduction of gauges and probes in the architecture of software systems for monitoring the system. An adaptation engine then uses this monitored data to plan and trigger changes in the system, e.g. in order to optimise performance or counteract failure of a component.
AC Autonomic Computing	2001	IBM	Compares self-management to the human autonomic system, which autonomously performs unconscious biological tasks. Introduction of the four central self-management properties (self-configuring, self-optimising, self-healing and self-protecting).
SPS Self-Regenerative Systems	2003	DARPA	Self-healing (military) computing systems, that react to unintentional errors or attacks.
ANTS Autonomous NanoTechnology Swarm	2005	NASA	Architecture consisting of miniaturised, autonomous, reconfigurable components that form structures for deep-space and planetary exploration. Inspired by insect colonies.

Table 1 from Huebscher Paper .

Self-Management properties

- Self-Configuration.
- Self-Optimization.
- Self-Healing.
- Self-Protection.

Inspired by Self-X properties: Autonomy, social ability, reactivity and Pro-activeness.

MAPE-K Autonomic Loop(I)

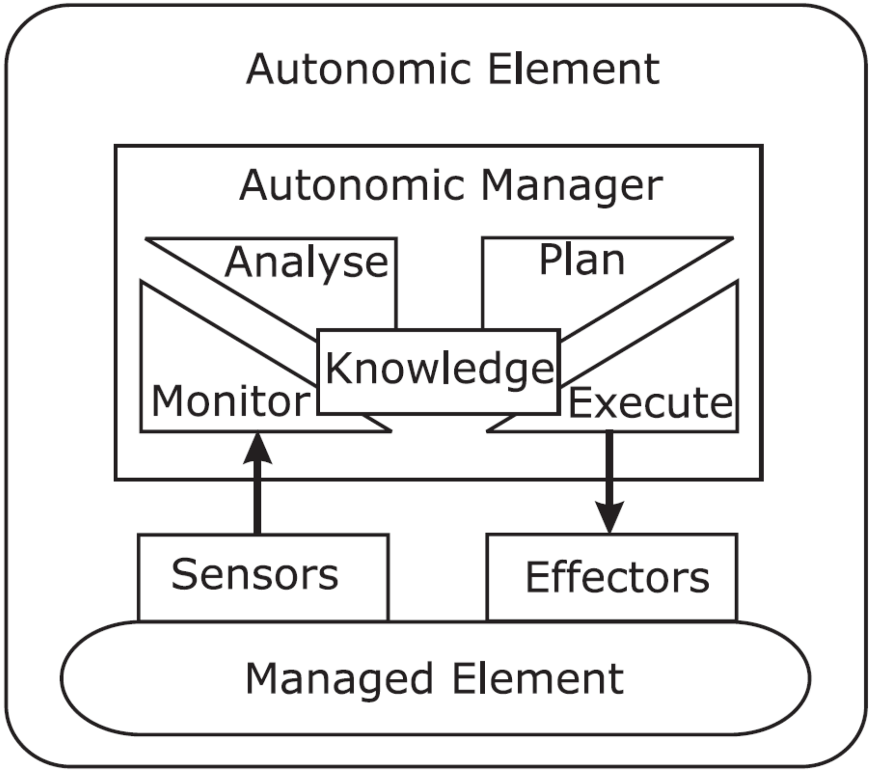

Figure 1 from Huebscher Paper .

MAPE-K Autonomic Loop(II)

- Model suggested by IBM.
- Monitor, Analyze data from sensors.
- Plan, Execute changes trough effectors.
- Manage knowledge (tricky).

Autonomic Manager (I)

- Monitors data from sensors and execute changes through effectors.
- Changes are based on goals fulfillment.
- There high-level goals achieved by completing low-level tasks.
- The goals are expressed trough Event-Condition-Action (ECA) policies, e.g.

"when 95% of web servers' response time exceeds 2s and there are available resources, then increase number of active web servers"

Autonomic Manager (II)

- An Utility function provide a quantitative measurement to the desirability of a system's state. But, is hard to define.
- Relates to Multi-Agents System, which cannot assure that an agent's state is desirable to the system's state.
- "As Autonomic Management solutions become more decentralised and less deterministic, we may begin to observe emergent features." .

Monitoring component

- Captures relevant data through sensors.
- Could be passive or Active.

Active differs from passive by adding custom functionalities to the software.

Adding a more dynamic approach when monitoring facilitates autonomicity.

Planning component

- Makes changes trough effectors based on what is being monitored.
- An stateless approach is very limited.
- The use of Architecture model provides a verification mechanism to evaluate if the integrity of the system kept after changes are made. Although, it does not eliminate ECA rules.

Knowledge Component

- Hard to distinguish with planning.
- Provides an effect adaption to the system.

Definitions:

- Utility. Measures “usefulness”.
- Reinforced learning. Policies from managed actions.
- Bayesian Techniques. Classify policies.

Multi-tier systems

- Aims to manage complex distributed systems.
- Must be aware of system responsiveness.
- Still there is a need for intervention in critical conditions.

Degrees of Autonomicity

1. Support(performance)
2. Core(prime objective)
3. Autonomous(fault tolerance)
4. Autonomic(Achieve SLAs or business goals)

Emerging Areas

- Power management.
- Data centers, Clusters
- Ubiquitous Computing

Conclusion

- AC is maturing and there are clear differentiation from other fields.
- Is a challenge to achieve SLAs.
- In the future, AC is expected to be part of general computing

Conclusion*

- Metasytems?
- Standardization?