Decision-Making Support for Software Adaptation at Runtime

Edith Zavala, Xavier Franch, Jordi Marco

Software Service Engineering research group (GESSI), Universitat Politècnica de Catalunya (UPC), Barcelona, Catalunya, Spain

In the last years, self-adaptive systems have become a crucial topic in the research and industry areas. Many initiatives to explore solutions for this kind of systems have emerged from both communities. One of the most popular approaches is the use of *feedback loops*. The *autonomic element* introduced by Kephart and Chess [1] and popularized with the IBM's architectural blueprint [2] for autonomic computing is one of the most accepted feedback loop solutions (see Figure 1). For instance, in [3] a feedback loop is utilized for detecting and adapting requirements that depend on context, affected by uncertainty. This approach uses machine learning techniques for identifying patterns on top of sensed data and determining the best adaptation of requirements, at runtime. According to the validation performed in [4] this approach is a very promising solution for executing self-adaptation at runtime.

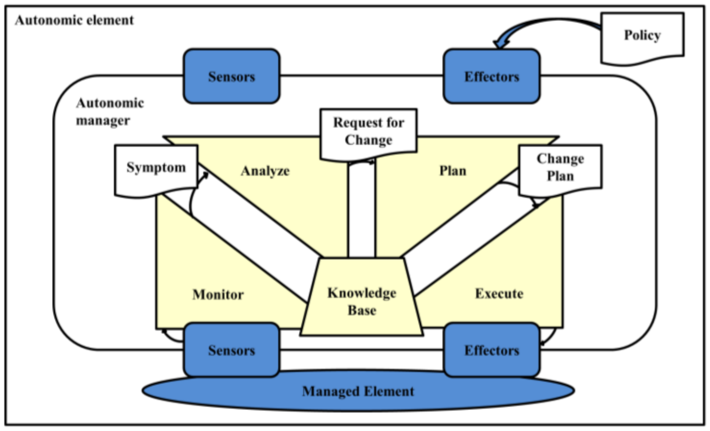

Fig. 1: IBM's autonomic element (from [1], [2], [3]).

Currently, more and more approaches use data analytics (e.g. data mining) for exploiting great amounts of runtime data collected through a monitoring infrastructure (e.g. through sensors and online data sources) in order to support the system runtime adaptation, as in [3]. The quality of the data (i.e. completeness, correctness, timely, etc.) provided by the monitoring infrastructures (e.g. Monitor element in Figure 1) affects directly the performance of the self-adaptive systems. Several approaches have been proposed to support monitoring of software systems; however, according to [5] most of them assume that the monitors are static components. This implies that the providers must know everything to be monitored at design time. This vision is too rigid to be usable in realistic settings. Nowadays, monitoring infrastructures supporting self-adaptive systems need to be reconfigured at runtime as well in order to respond to context changes, e.g. new measures to collect, at a different sampling rate, with a different protocol, etc.

Motivated by the need of new and innovative methods and techniques for effectively and efficiently analyzing, planning and applying monitoring infrastructure's reconfiguration decisions at runtime, we have explored new and existing solutions, and we have proposed an ongoing Monitoring Reconfiguration approach (see Figure 2). Our approach, based on the MAPE-K loop, intends to support Monitor element's reconfiguration for self-adaptive systems (managed by a feedback loop). Through the MAPE-K feedback loop our approach identifies conditions in which the Monitor could require a reconfiguration aligning it with the reconfiguration process of the Managed Elements and coordinately analyses, plans, and executes the reconfiguration and adaptation of both the Monitor (using the Knowledge Base as communication channel) and the Managed Elements.

In order to do so, we introduced a set of new elements to the MAPE-K loop presented in [3] and validated in [4]: a Mine Data element which is in charge of intelligently apply one or more data analytic techniques over the sensed data; a Reconcile Reconfigurations element which coordinates the reconfigurations of the Monitor and the Managed Elements in order to prevent e.g. contradictory reconfigurations, undesired states, etc.; and Policies which in our solution, unlike [3], are utilized by all the MAPE-K elements, in order to abstract specific application (i.e. Managed Elements) details, and distributed by the Knowledge Base. Finally, our approach allows the Domain Experts to modify Policies and supervise and request reconfigurations at runtime.

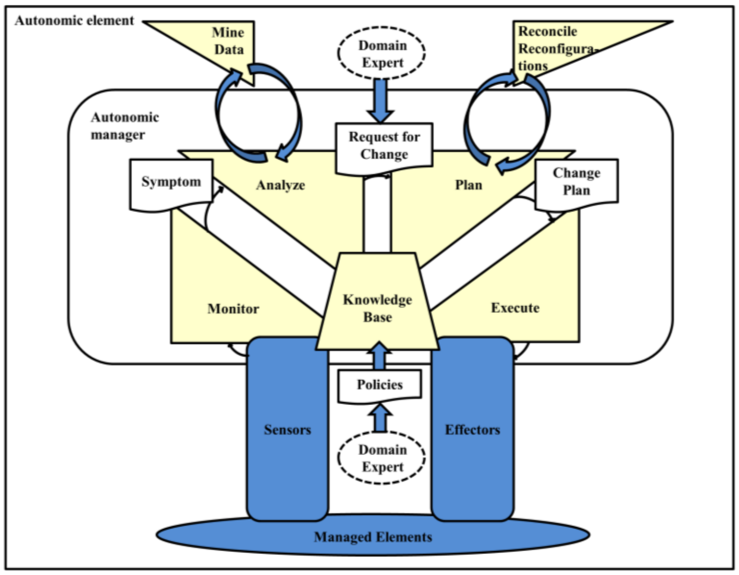

Fig. 2: Monitoring Reconfiguration approach high-level global view

Thanks to the abstraction of the Managed Elements' details (e.g. domain variables, technologies, etc.) our solution can support and be reused for an increasing number of applications at both architectural and implementation level and at design and runtime. Currently, we are defining a set of use cases in order to validate our approach.

References

[1] J.O. Kephart and D.M. Chess. "The Vision of Autonomic Computing," IEEE Computer 36.1, 41–50, 2003.
[2] IBM Corporation. "An Architectural Blueprint for Autonomic Computing," White Paper, 4th Edition, 2006.
[3] A. Knauss, et al. "ACon: A Learning-Based Approach to Deal with Uncertainty in Contextual Requirements at Runtime," Information and Software Technology 70, 85-99, 2016.
[4] E. Zavala, et al. "SACRE: A Tool for Dealing with Uncertainty in Contextual Requirements at Runtime," 23rd IEEE International Requirements Engineering Conference (RE), 278-279, 2015.
[5] R. Contreras and A. Zisman. "A Pattern-Based Approach for Monitor Adaptation," IEEE International Software Science, Technology & Engineering Conference (SWSTE), 30-37, 2010.