Self-organizing Service Supervision*

Concept Demonstration

Peter H. Deussen, Edzard Höfig
Fraunhofer Institute for Open Communication Systems
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
peter.deussen|edzard.hoefig@fokus.fraunhofer.de

ABSTRACT

Self-organisation and self-management are assumed to be complementary principles in the design of autonomic systems. We demonstrate by means of a simple example that self-organisation mechanisms are suitable for the autonomic formation of supervision infrastructures that supports the self-management of service providing configurations.

Keywords

Self-management, self-organisation, supervision, autonomic systems

1. BACKGROUND

One of the main anticipated capabilities of next generation service provisioning platforms is that of supporting self-organization, i.e. the autonomous formation of configurations to provide a certain service [7]. With the event of agent-like software platforms, self-organization can be expressed as emergent property of the complex interactions of ensembles of basic entities according to (more or less) simple behavioural rules (see [4] for an overview).

A principle which is orthogonal is that of *self-management*, i. e. the enactment of a system by some structure which automates management functions like fault management, global optimization and performance improvement, configuration, etc. The nowadays most prominent example of a self-management approach is that of Autonomic Computing [5], although similar approaches have been elaborated [3].

2. SUPERVISION IN THE ACE FRAMEWORK

A prototypical example of a software platform with self-organization capabilities is currently under development in the CASCADAS project [1]. It bases on so-called *Autonomic Communication Elements*—ACEs, for short—that provide a framework for autonomic services (Fig. 1). ACEs feature a discovery and binding mechanism that allows the formation of complex services on demand. Each ACE consists of a repository of basic functions that are available for service composition. A so-called *self-model* described the set of possible compositions of basic functions and external sub-services provided by other ACEs. To perform a particular composite service, a *plan* is derived form the self-model and executed by means of a specific interpreter. The derivation of plans is performed by an expert system (based on RuleML [8]).

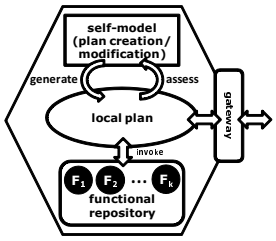

Figure 1: ACE architecture (simplified)

Self-organisation in the ACE framework is provided by a specific protocol of ACEs which bases on match-making [6]. A match-making ACE is one which connects other ACEs according to a certain service logic (which defines the conditions for a successful match), to available capacities (in this case work items are exchanged between matching ACEs), etc. For service discovery and binding, a simple protocol is provided as basic mechanism for self-organisation.

A particular instance of a service that bases on ACEs is a that of supervision, i. e. the ongoing observation of a configuration of ACEs and issuing of corrective measures if the supervised configuration enters an problem state such as an failure state, a performance problem, a configuration error, etc. The fact that supervision is defined as a supplementary service that supports the provisioning of another one implies that the same mechanisms for self-organized service composition are available both for the formation of the supervised and the supervision service. Since moreover the organisation of the supervision service (in terms of component distribution) follows essentially the organisation structure of the supervised service, we end up with a supervision infrastructure that "pervades" the supervised service configuration. We therefore refer to supervision configurations as *pervasions*.

Fig. 2 shows the basic pattern of a supervision pervasion comprising the following components (all components are implemented itself as ACEs):

*This work has been supported by the project "Component-ware for Autonomic, Situation-aware Communications, And Dynamically Adaptable Services (CASCADAS)" (IST-027807) funded by the FET Program of the European Commission.

Bionetics '07, December 10-13, 2007, Budapest, Hungary

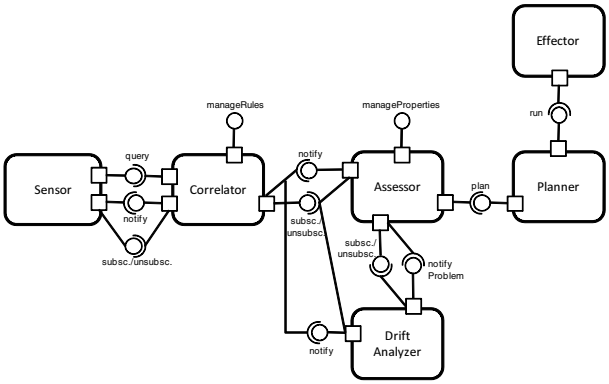

Figure 2: Supervision pervasion architecture (basic case)

- **Sensor** components links the supervision system with the ACE configuration under supervision. Each ACE provides an extensive management interface that allows accessing its internal state and session object. It furthermore provides notifications on all types of internal events including deployment of new local plans, state changes, sending and receiving of messages from other ACEs, etc.
- **Correlators** are responsible to aggregate monitored data from distributed sources and to correlate them, in order to extract meaningful indicators of the current health condition of the system under supervision
- **DriftAnalysers** try to anticipate future problem situations in the system under supervision. Additionally, information from the environment may be used to supplement the analytical process. Thus if Correlators are concerned with the current state of the system, DriftAnalysers are concerned with the possible evolution of the system.
- **Assessors** make assumptions on the current (or future) system health on the basis of raw data or the output of Correlators and DriftAnalysers, and invoke a Planner if necessary.

Sensors, Correlators, DriftAnalysers, and Assessor thus form an analysis system. The reactive part is provided by the following components:

- On the basis of the assessments generated by the Assessor, the **Planner** tries to compute a course of actions that is intended to resolve the detected problem. Planning is based on the actions described in the self-model of the ACE (or ACEs) under supervision.
- **Effectors** are responsible to execute plans. This is done by intercepting and modifying internal processes of the ACE under supervision in a comprehensive way utilizing its management interface.

3. DEMONSTRATION

The demonstration that we provide shows that self-management and self-organisation are not contradictionary but complementary organisation principles for autonomic systems. In particular, we demonstrate that a self-organisation mechanism such as the simple discovery and binding protocol provided by ACEs can be used for automatic (topological) configuration of supervision pervasions.

In the demonstration, we concentrate on a simple but essential supervision task. Given a complex service configuration, some components may crash, deadlock, etc. The detection of those "dead" components and their replacement or restart as well as their re-initialisation and re-alignment to the state of the complete service configuration, is referred to as *lifelines validation*. We demonstrate that:

1. A supervision pervasion suitable for lifelines validation can be set-up in a purely automated way that requires no human intervention.
2. Non-functional components can be detected, and replaced, without again without human intervention.

The concept demonstration provides an illustration of the idea that self-organisation and self-management (exemplified by service supervision) are orthogonal but combinable design principles for autonomic systems.

4. OUTLOOK

In this demonstration, we have concentrated on the structural formation of supervision pervasions. Job assignment, component configuration, and the associated computation of a suitable distribution of supervision components have not been addressed. The question how this can be done in a generic way with minimum or none human intervention is one of the ongoing research topic in the CASCADAS project. Recent advances [2] in the area of model based validation are considered as a fruitful direction to solve these problems.

5. REFERENCES

[1] Cascadas project home page: www.cascadas-project.org.

[2] P. H. Deussen. Model based reactive planning and prediction for autonomic systems. In *Proc. INSERTech (INnovative SERvice Technologies), workshop co-located with: First International Conference on Autonomic Computing and Communication Systems (Autonomics 2007)*, Rome, Italy, October 2007. ICST. Published on CD.

[3] P. H. Deussen, G. Valetto, G. Din, T. Kivimaki, S. Heikkinen, and A. Rocha. Continuous on-line validation for optimized service management. In *EURESCOM Summit*, 2002.

[4] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli. A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, 1(2):223–259, 2006.

[5] J. Kephart and D. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–52, 2003.

[6] P. Michiardi, P. Marrow, R. Tateson, and F. Saffre. Aggregation dynamics in service overlay networks. *Proc. First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, pages 129–140, 2007.

[7] C. Moiso, R. Alfano, and A. Manzalini. Distributed service framework: an innovative open eco-system for ict/telecommunications. In *Proc. INSERTech (INnovative SERvice Technologies), workshop co-located with: First International Conference on Autonomic Computing and Communication Systems (Autonomics 2007)*, Rome, Italy, October 2007. ICST. Published on CD.

[8] The rule markup initiative. http://www.ruleml.org/.