The Knowledge Engineering Review, Vol. 21:3, 189–194. © 2006, Cambridge University Press
doi:10.1017/S0269888906000877 Printed in the United Kingdom

Editorial

The infrastructures of Autonomic Computing

TORSTEN EYMANN

[1]*Chair of Information Systems Management, University of Bayreuth, 95440 Bayreuth;*
e-mail: torsten.eymann@uni-bayreuth.de

Abstract

The papers in this special issue represent an early view on different aspects of a new research field in computing science. They were presented as best papers in a series of European workshops called 'Self-adaptive and autonomic systems' (SAACS), which is co-located with the DEXA conference every year.

1 Introduction

> Civilization advances by extending the number of important operations which we can perform without thinking about them
>
> Alfred North Whitehead

The vision of Autonomic Computing has gained significant interest in the last years, and become a popular buzzword. Autonomic computing (Kephart & Chess, 2003) describes a concept of self-organizing information technology, where the functionality of the computing system is an emergent feature of the capabilities and actions of the components. Without any central controller, this system is capable of configuring, healing, organizing and protecting itself (the so-called CHOP properties). In contrast, classical IT control involves a centralized controller instance with global knowledge about the current status of the computing system, and a detailed regulation mechanism to 'heal' deviations from a defined 'normal' status. Centralized computation is said to suffer from poor scalability, high maintenance costs and low adaptability.

Autonomic computing uses a biological paradigm as a design and control metaphor, the autonomic nervous system. The core CHOP properties of the Autonomic Computing concept are intended to be an electronic realization of the respective mechanisms of the human body.

The first article in this issue, 'A health-check model for autonomic systems based on a pulse monitor' by Sterritt & Bustard, brilliantly follows that paradigm to the letter. The approach uses a 'pulse' monitor for each autonomic element, which provides a reflex reaction facility and basic information on the current state (health) of that element.

Self-organization can be found in other parts of our natural environment, e.g. biological evolution, social group behaviour, market dynamics phenomena and other complex adaptive systems. Autonomics refers to our own human nervous system, Catallaxy (Eymann *et al*., 2003) to self-organizing markets in Economics, Stigmergy to coordination without communication in insect colonies. All these ideas have in common that the solution for growing complexity both in scale and semantics is decentralized control, based on local information and executed through local effects which build up to a system-wide desired behaviour.

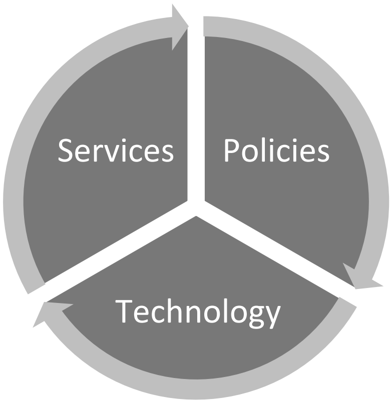

Figure 1 Infrastructure framework for Autonomic Computing

It is not surprising that projects labelled Autonomic Computing are thus manifold, coming from diverse backgrounds and academic habitats, and aiming at a variety of technological and scientific knowledge increase. This is because it is not only hoping to solve the many problems that industries are experiencing, with unreliable or increasingly more complex environments. In addition, it brings a unique challenge to computer scientists and engineers to add the concept of the system's 'self'. Furthermore, it attracts a diverse community from biological systems science to operating systems engineering, who are often able to present solutions that have evolved from earlier work.

Historically, Autonomic Computing clearly builds on the distributed artificial intelligence paradigm. Multiagent systems (Weiss, 1999) have thus gained renewed attention and a rewarding application area. In striking similarity to the evolution of Distributed AI itself, the key motivation aspect again lies in the increasing size and complexity of the information systems to be controlled, and a non-negligible growth in their control costs. The concept of self-organizing computing is not new. Earlier concepts, such as cybernetic (Wiener, 1998) or autopoietic (Maturana, 1999) systems, failed because of technological immaturity, missing business models or a combination thereof. It seems that technological progress now allows another, this time more promising look.

2 Infrastructural spheres of Autonomic Computing

To get a clearer look at the prospects and hurdles, chances and risks of Autonomic Computing, let me divide the complex concept into three spheres: the technological infrastructure, the services infrastructure and the policies infrastructure.

1. The computing technology infrastructure describes the technological progress, the software and hardware modules and the engineering processes to build these.
2. Having the technology in place is a prerequisite for creating new products and services that benefit from self-organizing computing. In their entirety, these new products and services build up the services infrastructure of potential Autonomic Computing business.
3. Business, however, needs rules for protecting legitimate rights and properties of the participants. The policies infrastructure describes a joint understanding acceptance of rules, norms and laws as well as agreed-on measures to regulate and enforce compliance.

The infrastructural spheres are not separate from each other. Without technology, there are no services to be invented. Without services in action, no policies are needed to restrict or allow their usage. In a feedback loop, the requirement e.g. to express machine-readable rules and software-based norms enforcement leads again to technological development. New concepts and technologies will inevitably again enable innovative services and at the same time require their regulation. Figure 1 shows the three infrastructure spheres schematically. Taking technological inventions into account as an existing, constant flow of fresh ideas and concepts of computer

https://doi.org/10.1017/S0269888906000877 Published online by Cambridge University Press

scientists, the research challenge will mainly lay in the setup of the services and the policies infrastructure. The following paragraphs describe some research questions of these infrastructures in more detail and set the articles in this special issue in relation.

3 The open service infrastructure of Autonomic Computing

Most Autonomic Computing environments are thought to be closed worlds, in which system elements provide resources benevolently. This is true for the usual scientific environments, e.g. Open Grid networks or closed parallel computing clusters – the benefits for the users are counted in participating in scientific progress, and the promise of eventual reciprocity. In business environments, benevolent cooperation is centrally ordered – autonomous decision-making is only allowed as far as it does not interfere with the business goals of the company.

However, Autonomic Computing is not a concept to be confined within organizational borders. If we take visions of computing infrastructures serious, we will soon witness comparing computing infrastructures to the super-national electricity grid (Grid Computing) or energy, water and other utilities grids (Utility Computing) (Carr, 2003). These ubiquitous infrastructures are open and non-discriminating with regard to users wanting to join – anybody can participate in such an infrastructure. A small glimpse into the future might be the current trend 'Web2.0', which recombines infrastructure software elements to create new services.

Practically, demand and supply of new and existing services will define a services market. Markets have the advantage to collect existing resource supply and thus, usually, to achieve evenly utilization by levelling heterogeneous user behaviour. Like other utilities, the services to be traded on those markets in huge numbers, are of simple nature. They are distinguishable by service quality characteristics, but convertible otherwise. This means that, given equal characteristics, competition will take place by signalling lower prices than the competition. However, the economic models underlying those markets are yet to be found, and similarly, which measure of utility guides the behaviour of participants and how transfer of goods and values (electronic money) will be realized, measured and billed. Without such markets, Autonomic Computing will remain yet another model for running cluster computing environments.

4 About the necessity to create an open Autonomic Computing policy infrastructure

For open 'utility computing' autonomic environments, we will soon encounter the demand for a policy infrastructure, like those in our non-electronic, physical society. Physical communities or networks frequently show very specific, only partly legally binding agreements and rules. Those policies define the resource-wise contributions of the network participants, the share of the utility gain, dealing with conflicts and mechanisms to enforce cooperative behaviour of the participants. Self-organization, as in Autonomic Computing, requires all this, as the participants individually (autonomously) follow their pre-set goals. As computing systems always belong to humans (or businesses run by humans), human goals are those which ultimately define the course of the computational participants.

The paper 'Policy-based techniques for self-managing parallel applications' by Anthony presents an empirical investigation of policy-based self-management techniques for parallel applications, and an adaptive strategy for the run-time deployment of tasks of parallel applications. The strategy is based on embedding numerous policies which are informed by contextual and environmental inputs. The policies govern various aspects of behaviour, enhancing flexibility so that the goals of efficiency and performance are achieved despite high-levels of environmental variability. A prototype self-managing parallel application is used as a vehicle to explore the feasibility and benefits of the strategy.

Autonomic computing, by definition, is not a topic for synchronized, targeting and on-the-spot network management. It needs mechanisms to make risks calculable and process efficiently – without, the system as a whole will become incapable of action. For managing the individual,

https://doi.org/10.1017/S0269888906000877 Published online by Cambridge University Press

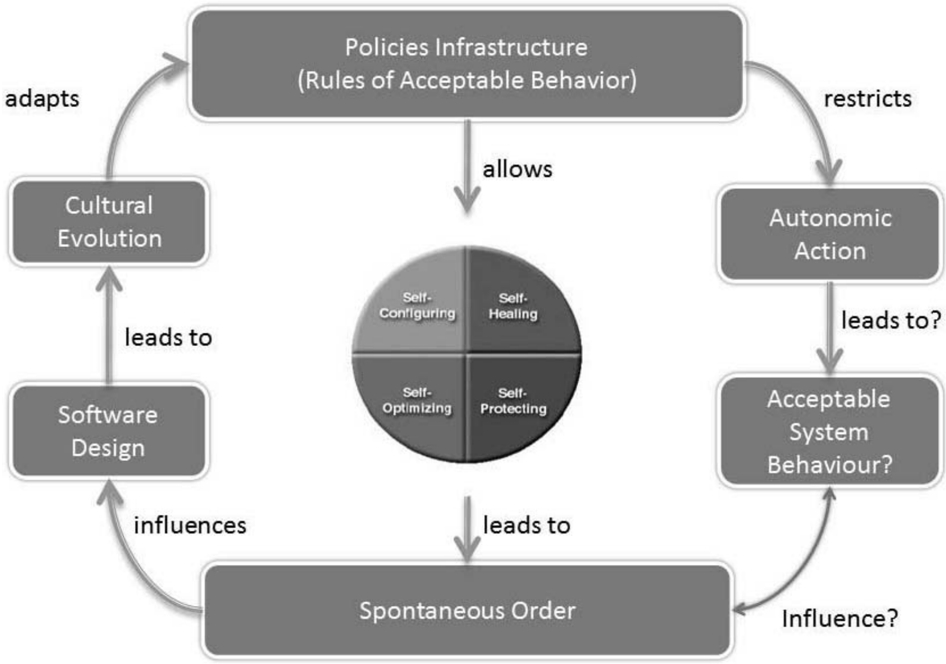

Figure 2 Regulatory framework for self-organizing computing

autonomous processes in the system, one needs to find, enforce and monitor common rules, norms and institutions. Without those, we will face unintended consequences, non-innovation and resistance to application of that technology. However, the guarding policies are not fixed. With changing environment and changing system goals, even the policies need to adapt. This creates a metaproblem: the definition of policies for adaption of policies.

Figure 2 shows a framework of elements to achieve spontaneous order. The circle begins with the software designers of the original system creating and defining an element's behaviour. By way of a cultural evolution, rules of acceptable behaviour become refined and give way to the next version of system inhabitants, who will be released in the information system and shape it to their needs. The final open question is, whether the spontaneous order provides 'acceptable behaviour' of the system – in principle, spontaneous order has no conscience and knows no guilt.

A related framework is presented in the paper 'Enabling dynamic composition and coordination for autonomic Grid applications using the Rudder agent framework' by Li & Parashar. Rudder characterizes how agents discover, select and compose elements, and defines agent interaction and negotiation protocols to enable appropriate application behaviours to be dynamically negotiated and enacted.

Dowling *et al.* provide a positive prototypical case, in their paper 'Building Autonomic Systems using Collaborative Reinforcement Learning'. The paper presents Collaborative Reinforcement Learning (CRL), a coordination model for online system optimisation in decentralised multi-agent systems. In CRL system optimisation problems are represented as a set of discrete optimisation problems, each of whose solution cost is minimized by model-based reinforcement learning agents collaborating on their solution. CRL systems can be built to provide autonomic behaviours such as optimising system performance in an unpredictable environment and adaptation to partial failures. In this case, the system behaviour will always be acceptable.

It is, however, as possible to disrupt the self-coordination through targeted violation of the 'regulatory framework'. The automated pursuit alone of the individual goals does not produce an acceptable behaviour of the entire system, e.g. in terms of robustness, controllability and the adherence of security criteria for the individual participant. Whether these perceptions can be generalized and used for the design of decentralized information systems (or information systems

https://doi.org/10.1017/S0269888906000877 Published online by Cambridge University Press

in a decentralized environment) and lead to more efficient paradigms for the implementation of computers, needs future research efforts.

The question remains open as to which is the most effective framework to achieve spontaneous order. The necessary regulation framework required can *ex ante* only be specified as trial and error. In this context, trust in, and generating trust through, Autonomic Computing is the main requirement for acceptability. In their article 'Can self-managed systems be trusted? Some views and trends', McCann *et al.* discuss four aspects of trust in self-adaptive computing systems with the aim to identify trends and pinpoint areas that require more investigation.

4 Testing autonomic systems – the two faces of tomorrow

Two articles in this issue finally deal with the question of setting up a test environment for an Autonomic System – which is not only a research question, but has already been the central topic of a SF novel. With regard to the framework in Figure 2, the testing process tries to answer the question, if autonomic action leads to acceptable system behaviour.

In his 1979 novel *The Two Faces of Tomorrow*, science fiction author James P. Hogan staged his plot against a background of a worldwide Autonomic Computing system in the 21st century. This computing infrastructure is responsible for managing many of the mundane tasks of running everyday life: switching city lights on and off, regulating traffic, transporting goods and materials between locations. The system is self-organizing, self-healing and learns from small mistakes. New algorithms are attached to the system frequently, adding complexity. Over time, the need for a fundamental software update arises.

However, as the system operates so much of Earth's infrastructure by now, the consequences of running unknown software can be grave, but testing the software is impossible without having a test planet environment at hand. Therefore, for updating the autonomic system, two contrary positions develop: one is, to just reboot with new, untested algorithms, and take the ensuing errors into account until the adaptation algorithms have succeeded – errors that can include human deaths, e.g. when traffic regulations fail. The second position is to find a testing environment, which in the novel is a space station, currently under construction.

After the autonomic system on the space station is freshly installed, the CHOP circle of constructing elements, healing, organizing and protecting begins. The interesting twist of the story is that the human construction workers become themselves enemies of the Autonomic Computing infrastructure – by attaching and re-attaching space station elements and switching electricity on and off during the construction process, which the Autonomic Computing system then consequently tries to heal and to protect itself from. Bottom line: if you ever receive an electric shock when trying to unplug your computer, you may have experienced your computer becoming autonomic – maybe it just protected itself.

The paper 'Composing simulation architectures for autonomic systems' by Butler *et al.* introduces the concept of Dreaming. In their approach, the authors remark that a critical differentiator between a simulation and a deployed application is that simulations are allowed to fail. As truly autonomous applications evolve, this capacity for simulation must be built in from the ground up or the benefits of experience – including the ability to tolerate failure – will be lost. This must be achieved without undermining the global correctness of visible application behaviour. The authors suggest an engineering approach to enable the introduction of such simulation with minimal or no recoding and propose a composition architecture to allow for safe dynamic deployment in substantial autonomic systems.

Thereska *et al.* add consciousness in their paper 'Towards self-predicting systems: What if you could ask ''what-if''?' They propose a what-if approach that allows interactive exploration of the effects of system changes, thus converting complex tuning problems into simpler search problems. Through two concrete management problems, automating system upgrades and deciding on service migrations, the authors identify system design changes that enable a system to answer What-If questions about itself.

References

Carr, N. G. 2004. *Does IT Matter?* Boston, MA: Harvard Business School Press.

Eymann, T., Reinicke, M., Freitag, F., Navarro, L., Ardaiz, O. & Artigas, P. 2005. A Hayekian self-organization approach to service allocation in computing systems. *Advanced Engineering Informatics* **19**(3), 223–233.

Eymann, T., Sackmann, S., Müller, G. & Pippow, I. 2003. Hayek's Catallaxy: a forward-looking concept for information systems. In *Proceedings of the American Conference on Information Systems (AMCIS'03), Tampa*.

Kephart, J. O. & Chess, D. M. 2003. The vision of autonomic computing. *IEEE Computer* **36**(1), 41–50.

Maturana, H. R. 1999. The organization of the living; A theory of the living organization. *International Journal of Human-Computer Studies* **51**(2), 149–168.

Weiss, G. 1999. *Multiagent Systems*. Cambridge, MA: MIT Press.

Wiener, N. 1998. The history and prehistory of cybernetics. *Kybernetes* **27**(1), 29–37.

https://doi.org/10.1017/S0269888906000877 Published online by Cambridge University Press