Executive Summary

Organic Computing - Controlled Self-organization

— Dagstuhl Seminar —

Kirstie Bellman[1], Mike Hinchey[2], Christian Müller-Schloer[3], Hartmut Schmeck[4] and Rolf Würtz[5]

[1] Aerospace Corp. - Los Angeles, USA
Kirstie.L.Bellman@aero.org
[2] NASA - Greenbelt, USA
Mike.Hinchey@lero.ie
[3] Universität Hannover, D
cms@sra.uni-hannover.de
[4] Universität Karlsruhe, D
hartmut.schmeck@kit.edu
[5] Ruhr-Universität Bochum, D
rolf.wuertz@neuroinformatik.ruhr-uni-bochum.de

1 The Objective

Organic Computing (OC) has become a challenging vision for the design of future information processing systems: As they become increasingly powerful, cheaper and smaller, our environment will be filled with collections of autonomous systems equipped with sensors and actuators to be aware of their environment, to communicate, and to organize themselves in order to perform the actions and servic-es that seem to be required. However, due to increasing complexity we will not be able to explicitly design and manage all intelligent components of a digitally enhanced environment in every detail and anticipate every possible configuration. Therefore, our technical systems will have to act more independently, flexibly, and autonomously, i.e., they will have to exhibit life-like properties. We call such systems "organic". Hence, an "Organic Computing System" is a technical system, which adapts dynamically to the current conditions of its environment. It will be self-organizing, self-configuring, self-healing, self-protecting, self-explaining, and context-aware. After the successful initial Dagstuhl Seminar on Organic Computing in January 2006 with its emphasis on "Controlled Emergence" this seminar focused on controlled self-organization (SO). The major objective of the seminar was to explore the question "How can we build useful self-organizing systems?" This was expressed by three main topics for the seminar:

1. **Basic understanding of self-organization:** We need an exact definition of the terms commonly used in connection with SO. If possible, a quantitative notion of SO should be reached. This involves compari-sons of SO in technical systems with natural ones such as physical systems, swarms and the brain.

Dagstuhl Seminar Proceedings 08141
Organic Computing - Controlled Self-organization
http://drops.dagstuhl.de/opus/volltexte/2008/1566

2. **Organization of technical SO systems:** It is not clear today how future technical systems will be organized to adapt in re-sponse to changing conditions. It seems that special components will be neces-sary that are able to assess the current state of the system and its environment (observation) and react properly (control) in order to achieve a higher-level objec-tive. It has to be discussed whether these components should be implemented explicitly or implicitly, in a centralized or a distributed way. Since such compo-nents are non-productive, one should discuss the allowable overhead in comparison to the benefit reached by SO.
3. **Design of SO systems:** First of all, in the design of SO systems one has to determine the facets of a sys-tem that should be open for SO, i.e., one has to decide on its degrees of freedom. Beyond that, SO systems should need no design because they are supposed to configure and learn their structure and behavior themselves. More realistically, we have to study carefully how we can guide an SO system into the desired direction, avoiding unwanted or forbidden behavior.

The seminar was attended by 32 participants with the majority coming from Germany and a strong fraction from the United States. Starting with an extensive introductory session, the seminar was organized as a sequence of couples of short presentations followed by intensive discussions, triggered by the presentations and by explicit questions on their overall topic. This more or less created a sequence of panels. The emphasis on discussions inspired a lively exchange of ideas. The first session on "Distributed self-organizing applications" presented generic dis-tributed architectures (reconfigurable hardware, middleware,), and applications like air traffic control (with highly strict security and safety requirements) and smart cam-era systems. The common questions were:

– Why self-organization (roles, problem solved,Ě)?
– What does self-organization mean in your example? How is it demonstrated?
– How was self-organization contained, constrained, controlled?

The second day had sessions on "Bio-inspired concepts and systems", "Learning", and "Dependability", the panelists had to respond to the questions:

– Why self-organization (roles, problem solved,Ě)?
– What does self-organization mean in your example? How is it demonstrated?
– How was self-organization contained, constrained, controlled?
– Is every form of learning of interest to OC?
– What kinds of learning are adequate for dynamic environments?
– What does dependability mean for OC systems?

The centrality of learning became clear on this day. There is growing demand for the integration of symbolic and subsymbolic learning. As system observation becomes simpler and cheaper, methods for mining the observations become more important. The need for user-instructable systems is becoming more pressing, and the technol-ogy is developing rapidly.On Wednesday and Thursday, the sessions addressed concepts for the design, engineering of organic computing systems, their questions were

- How does self-organization influence design?
- How can we determine the balance between explicit design and "freedom for adaptive behavior"
- What kinds of techniques are needed to express constraints, control for self-organization?
- How do we verify or certify the expected behaviour ?
- Is there a specific design methodology for "trustworthy organic/adaptive/SO" systems?

More general talks were presented on Generic Organic Computing architectures and wrappings as a form of test environment for complex systems. It became clear, that the application of self-organizing systems is not confined to toy applications. Rather, they are required to be built around legacy systems to keep these under control.This requirement is particularly strong in hardware design. The need for learning at design time and runtime was emphasized. There are significant commonalities between complex hardware and software systems. Self-organized scheduling for the parallelization of optimization procedures was a new example for this. Thursday afternoon was devoted to working groups, checking the following questions for the applications in that group:

- Who decides on the goals?
- What kinds of interfaces are present to the user and to the environment?
- How are its results observed, controlled and used?
- When do we (or some external entity) actually interfere/control?
- How is interaction/cooperation/collaboration dealt with?

For these applications: What has to be added before they can demonstrate a capability within some larger application system? It would lead too far to list responses to these many questions. A major topic that was raised repeatedly was to identify the essential characteristics of Organic Computing in comparison to other areas like Autonomic Computing or merely self-adaptive or self-organizing systems. Some commonly agreed answers to this were that OC is a combination of self-adaptive and self-organizing systems. Whereas self-adaptive systems allow for some type of top-down design and the global behavior of self-organizing systems is typically designed in a bottom-up way, the design of OC sys-tems seems to need a combination of both, in particular, if we want to provide the possibility of controlled self-organization, which sometimes is also called controlled emergence. In the workgroup reports, striking commonalities in OC methodology be-tween subject areas as dieverse as robotics, hardware design, parallelization, net-working, and software quality control became apparent. A central theme was depen-dability. The requirement of different components for prediction and decision about actions was emphasized. As a direction for future applications multi-application test-beds were envisioned that would make rapidly changing objectives tractable. This will probably be robot playgrounds and surveillance scenarios. The talks of the seminar clearly demonstrated a range of applications where principles of OC have been used successfully. But, definitely, there is an urgent need for more investigations on how we can find adequate methods for managing the complexity of self-adaptive and self-organizing systems. The demand is obvious, and good partial solutions are already there.