08141 Abstracts Collection
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

Abstract. From March 30th to April 4th 2008, the Dagstuhl Seminar 08141 "Organic Computing - Controlled Self-organization"' was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Organic computing, self-organisation, design, adaptivity

08191 Executive Summary – Organic Computing - Controlled Self-organization

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.

Joint work of: Bellman, Kirstie; Hinchey, Mike; Müller-Schloer, Christian; Schmeck, Hartmut; Würtz, Rolf

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2008/1566

Self-organisation in Ubicomp

Michael Beigl (TU Braunschweig, DE)

Self-organisation and organic principles can be very helpful in Ubiquitous Computing and its applications. This talk presents Ubicomp systems and technology that use such technology to implement real-time behaviour, integration of enterprise software systems and wireless networks, high recognition rates for sensor enabled systems in noisy environments and novel types of communication systems. The talk especially covers aspects of the CoBIs project, Analog network coding and superimposing signals, self organisation for unsupervised map creation, and generation and use of quality of context parameters.

Keywords: Ubiquitous Computing, Ubiquitous Networks, Context detection, Enterprise business systems, organic computing, self organisation

Full Paper:
http://www.teco.edu/∼michael/publications.html

Evaluating the Learning Classifier System XCS on Autonomic System-on-Chip (ASoC)

Andreas Bernauer (Universität Tübingen, DE)

The autonomic System-on-Chip (ASoC) design methodology aims to simplify the chip-designer's life by taking care of the reliability issues of future nano-scale chips, along with the traditional issues of performance, power and area. The design methodology is closely linked to the ASoC hardware architecture, which provides a flexible way to monitor the hardware's state, evaluate it and act according to given goals. This needs a decision system implemented on the chip for which we chose the learning classifier system XCS, a temporal-difference, reinforcement learning method.

In this talk I give a brief overview of the current state of ASoC design methodology, where we can determine system reliability based on activity, power an temperature analysis, together with reliability block diagrams. Then I continue

to present our preliminary results on evaluating the XCS on a QuadCore. The results show that in the evaluated setup, XCS can find optimal operating points, even in changed environments or changed reward functions. This even works, though limited, without the internal genetic algorithm. The results motivate us to continue the evaluation for more complex setups.

Keywords: LCS, XCS, learning classifier system, ASoC, system-on-chip, organic computing, design methodology

Joint work of: Bernauer, Andreas; Fritz, Dirk; Bringmann, Oliver; Rosenstiel, Wolfgang

Current state of ASoC design methodology

Andreas Bernauer (Universität Tübingen, DE)

This paper gives an overview of the current state of ASoC design methodology and presents preliminary results on evaluating the learning classifier system XCS for the control of a QuadCore. The ASoC design methodology can determine system reliability based on activity, power and temperature analysis, together with reliability block diagrams. The evaluation of the XCS shows that in the evaluated setup, XCS can find optimal operating points, even in changed environments or with changed reward functions. This even works, though limited, without the genetic algorithm the XCS uses internally. The results motivate us to continue the evaluation for more complex setups.

Keywords: Dagstuhl Seminar Proceedings, System-on-Chip, design methodology, system reliability, learning classifier system, XCS, ASoC

Joint work of: Bernauer, Andreas; Fritz, Dirk; Sander, Björn; Bringmann, Oliver; Rosenstiel, Wolfgang

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2008/1564

An Artificial Hormone System for Self-Organizing Real-Time Task Allocation in Organic Middleware

Uwe Brinkschulte (Universität Karlsruhe, DE)

Middleware is a good place to realize self-organizing features like self-configuration, self-optimization or self-healing necessary for organic computing. By autonomously choosing an initial task allocation, middleware can configure a distributed system of processing elements. By changing the task allocation, middleware can optimize the system in case of changing environmental conditions and heal it in case of processing element or task failures. Especially for self-healing, it is important that these organic features are decentralized to avoid single points of failure.

This work presents an artificial hormone system for a completely decentralized realization of self-organizing task allocation. We furthermore show, that tight upper bounds for the real-time behavior of self-configuration, self-optimization and self-healing can be given. Finally, the communication load produced by the hormone system is calculated.

Joint work of: Brinkschulte, Uwe; von Renteln, Alexander; Pacher, Mathias

Self-aware Memory: Managing Memory Access in Heterogeneous and Dynamic Parallel Systems

Rainer Buchty (Universität Karlsruhe, DE)

Following current forecasts, future computing systems will not only be massively parallel but also heterogeneous and reconfigurable during run-time. The use of established methods to perform memory management, securing memory consistency and coherency, and providing memory-based synchronization methods is therefore not only hampered by the sheer amount of nodes, requiring other but centralized management methods to not become a bottleneck, but also by the heterogeneous and dynamically changing nature of such systems.

We therefore propose an alternative method based on self-awareness where management functions are moved into the memory components of a system instead of relying on centralized directory services or controlling software layers: memory components offer their resources to computing nodes based on that nodes' requirements, such as size, speed, and latency. Subsequent memory accesses are monitored and evaluated with respect to access rights and potential optimization scenarios such as autonomous data locality optimization or memory configuration directly performed among memory nodes. No involvement of processing nodes or higher system layers is required.

Using this approach not only enables scalability as required with respect to future massively parallel systems, but directly addresses the issue of dynamics and heterogeneity as the memory system itself is able to adopt to computing nodes' requirements. The approach therefore, as a side effect, provides an abstract, uniform memory view where a memory is solely defined by size and access parameters such as latency and throughput, but not its actual implementation.

Keywords: Memory Management, Memory Access, Synchronization

Joint work of: Buchty, Rainer; Karl, Wolfgang

See also: Rainer Buchty, Oliver Mattes, Wolfgang Karl: Self-aware Memory: Managing Distributed Memory in an Autonomous Multi-master Environment. ARCS 2008: 98-113

Engineering self-adaptive and self-organising systems with policies and metadata

Giovanna Di Marzo Serugendo (University of London, GB)

This talk provides a unifying view for the engineering of self-adaptive (SA) and self-organising (SO) systems. We first identify requirements for designing and building trustworthy self-adaptive and self-organising systems. Second, we propose a framework combining design-time and run-time features, which permit the definition and analysis at design-time of mechanisms that both ensure and constrain and control the run-time behaviour of an SA or SO system, thereby providing some assurance of its self-* capabilities. We show how this framework applies to both an SA and an SO system, and discuss several current proof-of-concept studies.

Keywords: Self-adaptive, self-organising, design-time, run-time, control, policies, metadata

Full Paper:
http://www.dcs.bbk.ac.uk/ dimarzo/papers/CS-TR-1018.pdf

See also: G. Di Marzo Serugendo, J. Fitzgerald, A. Romanovsky, N. Guelfi, "A Generic Framework for the Engineering of Self-Adaptive and Self-Organising Systems", CS-TR-1018, Technical Report, School of Computing Science, University of Newcastle, Newcastle, UK, April 2007

A Generic Framework for the Engineering of Self-Adaptive and Self-Organising Systems

Giovanna Di Marzo Serugendo (University of London, GB)

This paper provides a unifying view for the engineering of self-adaptive (SA) and self-organising (SO) systems. We first identify requirements for designing and building trustworthy self-adaptive and self-organising systems. Second, we propose a generic framework combining design-time and run-time features, which permit the definition and analysis at design-time of mechanisms that both ensure and constrain the run-time behaviour of an SA or SO system, thereby providing some assurance of its self-* capabilities. We show how this framework applies to both an SA and an SO system, and discuss several current proof-of-concept studies on the enabling technologies.

Keywords: Metadata, policies, self-adaptive, self-organising, software architecture

Joint work of: Di Marzo Serugendo, Giovanna; Fitzgerald, John; Romanovsky, Alexander; Guelfi, Nicolas

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2008/1563

Full Paper:
http://www.dcs.bbk.ac.uk/~dimarzo/papers/CS-TR-1018.pdf

See also: G. Di Marzo Serugendo, J. Fitzgerald, A. Romanovsky, N. Guelfi, "A Generic Framework for the Engineering of Self-Adaptive and Self-Organising Systems", CS-TR-1018, Technical Report, School of Computing Science, University of Newcastle, Newcastle, UK, April 2007.

Organic control by emergent behavior?

Peter Dittrich (Universität Jena, DE)

In this talk I will discuss the following questions: Can we use "emergence" to control the increasing complexity of our computational systems? And if so, where is the controller? And how to tell the controller our demands? These questions are especially interesting for those organic computing architectures that do not posses any explicit sub-system that can be identified as a controller.

Bio-inspired data management in sensor and actor networks

Falko Dressler (Universität Erlangen, DE)

We address the domain of data management in wireless sensor and actor networks by adapting concepts known from molecular biology. In particular, we present a rule-based control system for network-centric communication and data processing. Starting with an outline of biological communication techniques, which are known as cellular signaling pathways, we show the adaptation of these principles to the world of sensor networks. With our rule system, we are able to pre-process data, e.g. for data aggregation, and to provide data-centric communication based only on local state in the contributing sensor nodes. The programming of all nodes is performed using an extensible rule system that supports heterogeneous programming and self-adaptation of both, single sensor nodes and groups of nodes. First simulation results demonstrate that this system is able to compete with classical approaches for routing and data management in sensor and actor networks.

Keywords: Bio-inspired networking, sensor and actor networks, rule-based programming

Organic Computing for Recognizing and Dissolving Traffic Jams

Sándor Fekete (TU Braunschweig, DE)

Traffic is a complex systenm with many organic properties. New capabilities of ad-hoc networking combined with distributed algorithms and protocols allow changing the way how drivers in traffic interact. This talk presents a number of objectives, concepts and results, in particular for recognizing traffic jams and improving traffic flow. Important concepts are Hovering Data Clouds (HDCs), Organic Information COmplexes (OICs), and Adaptable Distributed Strategies (ADSs). Results include methods for recognizing and dissolving traffic jams.

This is joint work with Christopher Tessars, Christiane Schmidt (Braunschweig), Stefan Fischer, Horst Hellbrueck, Axel Wegener (Luebeck).

How self-adaptive systems can deal with software faults

Michael Grottke (Universität Erlangen - Nürnberg, DE)

Self-adaptive systems should automatically repair software (and hardware) problems. Wherever possible, they should anticipate such problems and prevent them from occurring in the first place.

This talk presents various types of software faults and discusses approaches for dealing with them.

Software faults are classified into Bohrbugs and Mandelbugs. While Bohrbugs manifest consistently under well-defined sets of conditions, Mandelbugs have complex causes, making their behavior appear chaotic or even non-deterministic. It is therefore usually difficult to identify Mandelbugs via root cause analysis; however, reactive recovery (e.g., application restart, failover to a replica) can be used to treat failures caused by Mandelbugs even before the problem has been fully understood. We discuss an elevated levels of failure recovery approach, in which a sequence of recovery techniques is applied.

Aging-related bugs, a subtype of Mandelbugs, have the property that their failure manifestation rate can increase with the time of execution. For such faults, proactive techniques reducing the fault activation rate and/or removing internal errors can prevent future failure occurrences. Both time-based and event-based approaches to scheduling "software rejuvenation" are addressed.

Keywords: Fault classification, failure recovery, failure prevention

Joint work of: Grottke, Michael; Trivedi, Kishor

Full Paper:
http://ieeexplore.ieee.org/iel5/2/4085604/04085640.pdf

See also: M. Grottke and K. S. Trivedi. Fighting bugs: Remove, retry, replicate, and rejuvenate. IEEE Computer 40(2):107-109, 2007.

Why Predicting Outliers in Software is a Good Thing to Do!

Michael G. Hinchey (Lero, IE)

A software reliability model is used to predict outliers, that is, values that significantly depart from the expected or mean values. In contrast with most projects that cleanse outliers from their databases because their presence distorts results, we tend to like outliers because by predicting them, we can help ensure safety in critical systems and perhaps use them to predict issues in evolving systems. We use the Space Shuttle failure data to make predictions of whether reliability goals are met if outliers should occur during test or the operational mission. In addition, prospective Shuttle software releases are analyzed to detect possible anomalous behavior that would call for re-inspection of the software to bring it into conformance with reliability specifications.

Joint work of: Hinchey, Michael G.; Schneidewind, Norm

Distributed Smart Cameras: An example of Self-organising systems

Jörg Hähner (Universität Hannover, DE)

We propose a decentralised, self-organising system architecture for wireless networked Smart Cameras in which PTZ cameras are used to spatially partition the area they observe. Each camera communicates with its neighbours and independently calculates the optimal position for its field of view. This is achieved by a decentralised algorithm that makes way for self-organisation in Smart Camera Systems. The system quickly adapts to new situations caused by joining and failing nodes. Simulations under real world constraints with hundreds of cameras show that scalability and reliability is achieved.

Additionally, I will outline further issues that need to be addressed in Smart Camera Systems.

Self Adaptive Software, Learning and Organic Computing

Robert Laddaga (BBN Technologies -Cambridge, US)

The talk is about the foundations of organic computing, with special attention to the relationships among self adaptive software, learning, and self organizing systems. It begins with a discussion of fundamental properties and foundational issues of Self-* systems. It briefly comments on the spectrum of biologically motivated approaches, and on the relationship between Self Adaptation and Self Organization. The talk then suggests that there can be a useful relationship

between symbolic learning and OC, and this could be further exploited by the OC research community, particularly in terms of communication between OC Systems and human users of those systems.

It describes the Bootstrap Learning Cup as an opportunity to take up that research direction. Finally, there is a description of Self Adaptive Software, and its current research challenges.

Keywords: Self adaptation, self organization, computer vision, symbolic learning

Organic Programming

Falk Langhammer (Living Pages Research GmbH - München, DE)

In this talk, I criticise the traditional approcah to object-oriented programming (OOP) as being too formal and failing to replicate the properties of real-world objects. Rather, I propose to introduce "TrueObjects" as being a close as possible reproduction of real-world objects within the software world.

Such TrueObjects can be manipulated, altered, cloned, used, persisted, searched for, protected, damaged, repaired, viewed, etc., just as we are used to – within a live production system without limitations. A formal programming language is only required in order to represent software TrueObjects within algorithms. Systems composed from TrueObjects are called "organic" because they can be grown organically step by step by users when and while using – just like humans grow cities – thereby allowing for software systems with unprecedented complexity.

I also present one particular such system which has been implemented and is in industrial use. It uses XML and Java: the "ercato engine". The TrueObjects in this engine are called "ercatons".

Keywords: Ercatons Objects Java XML Organic Software Engineering SOA Webservices

Self organisation applications: distributed control in network-based systems

Peter A. Lindsay (The University of Queensland, AU)

The talk begins with a brief introduction to the ARC Centre for Complex Systems (ACCS) and its research programs in distributed control in network-based systems. We are using applications from 3 different areas to study 3 very different paradigms of distributed control, and what methods and tools are common to them:

1. biology - how Genetic Regulatory Networks control cell-level growth and development of biological organisms;

2. socio-technical systems - how to let airlines choose aircraft 4D trajectories in Air Traffic Control;
3. economics - how agents acting in markets control prices, with particular application to electricity networks.

See www.accs.edu.au for more details.

The rest of the talk outlined 3 example applications of self-organisation being studied in the ACCS:

The first concerned GRNs and nodule formation on roots of legume plants. Botanists believe that the activation pattern for certain genes in the cells at root tips regulate nodule formation. In turn, they have observed that what is happening in cells at the shoot tips affect the gene activation patterns at the root cells, after a time lag. The challenge is to develop models that help biologists understand how the two systems (GRNs at root and shoot cells) interact to control nodule formation.

The second example was from ATC and investigated whether the "rules" for flocking behaviour in birds might be adapted to yield a simple set of rules for aircraft flying together in a moving ATC sector. The aim is to allow individual aircraft to self-optimise their 4D trajectories, while the whole group remains safely separated and stay within the moving sector.

The third example concerned understanding how individual sector controller behaviour affects traffic congestion and delays across multiple sectors. We have developed a modelling and simulation framework that allows us to modify controller agent behaviour and investigate emergent system behaviour, such as accumulated delays.

Keywords: Self organisation, air traffic control, genetic regulatory networks

The Organic Robot Control Architecture ORCA and Its Application to the Fault-Tolerant Walking Machine OSCAR

Erik Maehle (Universität Lübeck, DE)

During their evolution living organisms have developed very effective and efficient mechanisms like the autonomic nervous system or the immune system to make them adaptive and self-organizing also in case of new unforeseen situations, in particular injuries and diseases. These systems operate unconsciously and in an emergent way to make the body self-protecting, self-healing, self-optimizing and self-configuring. Inspired by these organic principles the control architecture ORCA (Organic Robot Control Architecture) for fault-tolerant robots is currently developed at the Universities of Lübeck and Osnabrück in cooperation with Fraunhofer AIS, St. Augustin. In contrast to the traditional fault tolerance approach of defining fault models explicitly, the "health status" of the system is continuously monitored by so called OCUs (Organic Control Units).

OCUs are closely coupled with BCUs (Basic Control Units) which implement the regular behaviours. In case an anomaly is detected OCUs modify BCUs such that the fault is tolerated in the best possible way. Also online learning shall be provided to improve the fault tolerance capabilities over time. To evaluate this approach, a six-legged walking robot OSCAR (Organic Self-Configuring and Adapting Robot) has been constructed. OSCAR uses a self-organizing gait pattern generation inspired by insects. First experiments on monitoring the health status of OSCAR's legs and to adapt to injured or lost legs by self-organization are reported.

Keywords: Self-Organization, Robot Control Architectures, Organic Robots

Optimization using the Grid

Sanaz Mostaghim (Universität Karlsruhe, DE)

Stochastic iterative search methods such as Evolutionary Algorithms (EAs) and Particle Swarm Optimization (PSO) have been shown to solve many real-world applications for decades.

These algorithms are naturally parallel; they contain several individuals which are being improved through generations. This parallel nature is particularly useful when implementing the algorithm on parallel computers. The motivation for parallelization is that for many practical optimization problems, evaluating a single solution already requires a significant computational effort. On the other hand most of the nature-inspired optimization techniques like EAs or PSO usually require the evaluation of a large number of solutions before producing some good results. One way to resolve this predicament is to employ parallel processing. While only few people have access to a dedicated parallel computer, recently, it also became possible to distribute an algorithm over any bunch of networked computers, using a paradigm called "grid computing". Grid Computing enables the sharing, selection, and aggregation of a wide variety of geographically distributed computational resources (such as supercomputers, computer clusters, storage systems, etc.) and presents them as a single, unified resource for solving large-scale and data intensive computing applications.

In this talk, two different parallelization algorithms for multi-objective optimization techniques using PSO are introduced.

Keywords: Multi-objective Optimization, Grid Computing, Particle Swarm Optimization

Organic Computing: Exemplary Scenarios

Gero Mühl (TU Berlin, DE)

After a short motivation for self-managing distributed systems, the talk focuses on three exemplary application scenarios in which self-organization is applied.

The first scenario deals with covering a location such as a shopping mall with wireless services using an ad-hoc network provided by nodes distributed among the location. The talk concentrates on how to adapt the number and position of service replicas to minimize the communication costs.

The second scenario discusses how to adapt the broker overlay topology of a publish/subscribe systems with respect to traffic patterns that vary over time. Finally, the third scenario gives an outlook of how to apply a model-driven software development approach such that non-experts are enabled to implement self-organizing applications.

The Importance of Human-System Negotiations in OC Systems

Phyllis R. Nelson (Cal Poly - Pomona, US)

The potential of OC systems to behave in unexpected ways creates new challenges for human-system communication in all phases of the lifecycle. In the design phase, the complexity and mutability of OC systems means that we will need to partner with the system (or a collection of models and simu- lations of the system) to discover and evaluate strategies for accomplishing overall purposes and goals. We will also need to partner with the system in devising, implementing, and interpreting tests in order to provide acceptable assurance of the reliability and trustworthiness of the realized and deployed system. During operation, the system and its owners, operators and users will need to negotiate in order to ensure both appropriate internal states and the ability of the system to address changing environments, purposes, and goals. Ultimately, these considerations suggest that the system will need new capabilities to model itself, abstract symbols of its internal state, and communicate both the symbols and their meanings to humans.

Keywords: Human-system interactions

Biologically Inspired Methods for Organizing Distributed Services on Sensor Networks

Franz J. Rammig (C-LAB - Paderborn, DE)

Future Wireless Sensor Networks (WSN) will require increasingly comfortable operating system services. This demand may go beyond the level as supported by state of the art approaches like TinyOS, Mantis, or ContikiOS. We propose to make use of a completely distributed way of implementing such OS services. I.e. instead of having an instance of the OS on each node of a WSN the services of the OS are distributed over the WSN. Of course this approach implies new challenges. Two of them are discussed in the presentation: Migration of services

to nodes such that the overall communication costs are minimized and forming clusters with the tendency to concentrate service requests inside the clusters and at the same time minimizing intra-cluster communication. For both problems biologically inspired solutions are discussed.

Migration of services is mapped on an Ant Colony Optimization (ACO) technique. Whenever an application makes use of a service, it leaves virtual pheromone on its path to the service. As usual this pheromone has the tendency to evaporate. Service migration is carried out in a two phase approach. Explorer packets travel from applications towards needed services comparing at each visited node the pheromone level of the neighbored nodes. The best candidate node for a service is such one where the "forces" due to the observed pheromone level are balanced. In a second phase it is checked whether the selected node has sufficient resources. If not a close-by node with sufficient resources is selected. This approach can be enhanced to take care also of request paths which are correlated.

As a clustering heuristics Division of Labor in swarms of social insects is used. By properly setting stimuli and thresholds, appropriate nodes obtain the role of cluster heads. Membership then is assigned in a similar manner. All remaining nodes assign themselves a suitability level. Their reaction time to requests from cluster heads to join a cluster then is reverse proportional to this suitability level. I.e. the first k nodes and by this the k nodes with highest suitability level receiving the request to join will be accepted to join. This heuristics works fine in quasi static environments. For highly dynamic ones another inspiration from nature is used. Making populated clusters more attractive but over-populated ones less attractive a positive and negative feedback are introduced. By combining these feedbacks each cluster gets assigned a certain value. Differences in these values of neighbored clusters result a force causing nodes from an over-populated cluster to migrate to the less populated neighbor.

Keywords: Sensor networks, distributed OS, bio-inspired heuristics

Joint work of: Rammig, Franz J.; Heimfarth, Tales; Janacik, Peter

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2008/1565

The Emergence of Intelligent Bahavior Through Play

Paul Robertson (BBN Technologies -Cambridge, US)

Humans seem to have an enormous capacity to learn and recall raw memories. While we are able to learn and improve skills by repetitive practice of those skills we are also able to draw upon past experiences in the formation of new skills and reuse aspects of old skills in the formation of new ones. Much of machine learning attempts to compress a large amount of training data into a very compact representation of a learned entity while discarding the training data itself. Much of this appears to be have been driven by cost of memory in

computers and the computational cost of manipulating large amounts of data. What is lost from discarding the training data is (1) incrementality, and (2) flexibility of knowledge structure. With memory prices plummeting and with modern computing architectures, we are in a position to move away from the highly compressed forms of learning and benefit from a more organic computing approach to learning intelligent systems. We describe some experiments with memory-based learning and discuss the practicality of the approach.

Keywords: Memory-Based Learning, High Dimensionality, Play, Exploration

Organic Computing: Controlled Emergence - Generic Architectures

Hartmut Schmeck (KIT - Karlsruhe Institute of Technology, DE)

During this seminar we have seen several approaches to the design of architectures supporting Organic Computing Systems. While it is valid to use different approaches there seem to be some common ingredients that are captured by the generic observer/controller architecture suggested by Branke et al. (2006). Important components of the observer are the data analysis unit, a predictor and an aggregator; the controller is responsible for mapping situations to actions, evaluating the fitness of executed actions with respect to their impact on the system (which allows for on-line learning) and generating new actions whenever necessary (by off-line learning, based on simulation). An evaluation of this architecture with respect to an application in traffic control reveals that it resembles a generic reference architecture for self-adaptive systems, i.e. it should be extended to include also the aspect of cooperation between collections of agents in the system under observation and control. Three different scenarios resembling self-adaptive, self-organising and controlled self -organising systems are outlined, followed by definitions of the degree of autonomy and of controlled self-organisation.

Keywords: Observer-controller architecture, self-adaptive systems, self-organising systems

Invasion - A New Programming and Architecture Paradigm

Jürgen Teich (Universität Erlangen-Nürnberg, DE)

We introduce a novel paradigm of parallel computing called invasive programming as well as a new class of computer architectures called invasive.

Invasive Programming denotes the capability of a program to run on a parallel computer to request and temprarily claim processing, memory, and communication resources in its neighborhood, to then execute in parallel on these claimed

resources, and finally to free these resources again. Invasion therefore includes self-organizing behaviour and requires the scalability of a program to run on a at compile-time unknown number of processors.

We will present examples of the benefit of invasiveness with respect to rapidly growing numbers of available resources on systems-on-a-chip.

Keywords: Organic Computing, MPSoCs

An Excuse for not being that Self-organized

Wolfgang Trumler (Universität Augsburg, DE)

In 1962 William Ross Ashby published an article on the "Principles of self-organizing systems" in which he gives a clear and formal description of an organization in terms of technical systems. He also proposes a definition of a good organization. Furthermore he shows, that a system in the usual sense can not be self-organizing. In this paper the principles introduced by Ashby are presented and explained in the context of the seminars scope to foster a better understanding of what self-organization can mean for the OC community.

A Self Distributing Virtual Machine for multicore FPGA

Klaus Waldschmidt (Universität Frankfurt, DE)

For complex embedded system on chips (MPSoC) a self distributing virtual machine on FPGA basis is discussed. The self distributing machine (SDVM) represents the virtualisation of a parallel, adaptive and heterogeneous FPGA (multicore) chip. It is a middleware between application and number and type of cores where the application run transparently distributed on several sites. Regular code and configware for dynamic reconfigurable FPGA migrates in the same way so that optimal computing resources can be provided at runtime.

The SDVM exploits the parallelism and dynamic features of todays FPGA. Some important features are:

- Task mobility between all PEs even if they are heterogeneous
- Virtualization of the I/O system
- Combination of the distributed memory of each PE to a virtual shared memory
- Reconfiguration of number and type of active FPGA resources at runtime
- Dynamic scheduling as well as code and data distribution

All features allow easier programming of the heterogeneous system

Keywords: FPGA, Multicores, Middleware

See also: ARCS 08, Proceedings, Dresden, Germany

A Virtual Layer for FPGA Based Parallel Systems (MP-SoCs)

Klaus Waldschmidt (Universität Frankfurt, DE)

Besides performance and time to market, robustness and reliability are important design targets for modern Systemson- Chip (SoCs). Despite these features the power consumption must be as low as possible. To meet these design goals parallel, flexible, and adaptive architectures are required [1].

Today, dynamically reconfigurable FPGAs are well suited to form a parallel architecture because they incorporate serveral hard- and softcores. To efficiently use such multicore systems a hardware independent system must be created which handles all cores. Further, optimizing the power management the number of active cores must be adapted dynamically to the current workload. To make these features manageable and augment the system with adaptivity a virtual layer is required which hides the - due to runtime reconfiguration - changing hardware system from the application software. The Scalable Dataflow-driven Virtual Machine [2] is such a virtualization of a parallel, adaptive and heterogeneous cluster of processing elements (PE). Thus, it is well suited to serve as a managing firmware for multicore FPGAs.

Joint work of: Hofmann, Andreas; Waldschmidt, Klaus

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2008/1561

Learning to look at humans by organic computer vision techniques

Rolf Würtz (Ruhr-Universität Bochum, DE)

In order to reach its goals of organic interaction with humans Organic Computing technology needs some understanding of the actions of humans, which can be analyzed based on camera data. Computer vision has developed thousands of algorithms to solve specialized tasks under special constraints, but is still falling short of human visual capabilities. We are trying to push this border by organic combination and extension of computer vision procedures.

I will present three steps on a way to interpret human behavior. First, I discuss a multiagent architecture and the technique of democratic integration for the combination of many cues to arrive at a robust decision. This implements a combination of hand and head localization modules leading to trajectories, which need to be interpreted. To that end I present an self-organizing extension of Hidden Markov models with an application to recognizing signs from British sign language.

Finally, I present results from the attempt to automatically learn tracking constraints for human bodies based only on video data. Here, the trajectories

of points on the body are clustered to reveal coherently moving parts, which correspond to limbs. This is done by a self-organized parameter tuning of spectral clustering.

Joint work of: Walther, Thomas; Krüger, Maximilian; Schäfer, Achim; Tewes, Andreas; Würtz, Rolf