09201 Abstracts Collection
Self-Healing and Self-Adaptive Systems
— Dagstuhl Seminar —

Artur Andrzejak[1], Kurt Geihs[2], Onn Shehory[3] and John Wilkes[4]

[1] Zuse Institute Berlin (ZIB), D
andrzejak@zib.de
[2] Universität Kassel, D
geihs@uni-kassel.de
[3] IBM - Haifa, IL
onn@il.ibm.com
[4] Google, Mountain View, USA
john.wilkes@google.de

Abstract. From May 10th 2009 to May 15th 2009 the Dagstuhl Seminar 09201 "Self-Healing and Self-Adaptive Systems" was held in Schloss Dagstuhl – Leibniz Center for Informatics. 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 are put together in this paper. Links to extended abstracts or full papers are provided, if available. A description of the seminar topics, goals and results in general can be found in a separate document "Executive Summary".

Keywords. self-healing, self-adaptive systems, dependability, root-cause analysis, system and software modeling, large IT infrastructures

A case for transparent software rejuvenation

Artur Andrzejak (Zuse Institute Berlin (ZIB))

Software rejuvenation - cleansing of application state via a scheduled reboot - is a crude yet very effective technique for resolving problems caused by resource depletion. It is also considered the method of choice for fixing failures due to rare combinations of conditions or due to non-determinism - phenomena of growing importance in complex and distributed systems.

On the negative side, rejuvenation has several drawbacks, including interruption of availability during the reboot and potential loss of working data. Approaches such as microrebooting and recursive restartability developed in the ROC project could mitigate the former problem. However, they require severe changes and even a redesign of a supported application which seriously limits their practicability.

In this position talk we argue that the potential of software rejuvenation for (temporary) healing of a broad spectrum of applications is only partially exploited. Moreover, a prerequisite for wide acceptance of this tool is the assumption that rejuvenation is executed transparently to the application and does not require code changes (and maybe even no knowledge about application internals). We discuss several key elements for implementing this vision, including: transparent process replication via virtualisation and harnessing abundant system resources; techniques for migration of working data; cleaning of state and data structures; integration of rejuvenation support into commodity operating systems.

Keywords: Recovery and repair techniques: adaptive software rejuvenation, Recovery-Oriented-Computing (ROC)

See also: Luis Silva, Javier Alonso, Paulo Silva, Jordi Torres, Artur Andrzejak: Using Virtualization to Improve Software Rejuvenation, IEEE-NCA, Cambridge, MA, USA, July 2007.

A Programming Model and Run-Time Architecture for Adaptive Service Orientation

Umesh Bellur (Indian Institute of Technology Bombay, IN)

Enterprise distributed computing infrastructure today revolving largely around remote access to objects, is mostly static in nature. By this we mean that applications written on them have to have the handle to a remote object and know a priori how to contact it via a language specific stub of some kind usually compiled in with the application.

This leads to hardwired, stovepiped solutions that reuse little and are expensive to maintain. The concept of service oriented computing alters some of that by providing for dynamic binding of requests to services. This coupled with the externalization of control flow sets the stage for adaptive service orientation where elements of service execution can be changed dynamically.

In this paper, we present a programming model and run time architecture for adaptive service orientation. Our approach is based on semantic descriptions of service elements (tasks) augmented with contextually dependent resource-based requirements.

For this purpose, we model the dependencies between service elements as variability points. The runtime dynamically re-configures system behavior by mapping the variability points to services providing the needed functionality. Along with the middleware design needed for such an approach, our key contribution is our programming model - called adaptable programming - and show how it can be used to put together adaptive service-oriented systems.

Keywords: Service Oriented Architecture, adaptable computing

Joint work of: Bellur, Umesh; Nanjungud, Narendra

Quality-aware design of dependable process

Cinzia Cappiello (Politecnico di Milano, IT)

In advanced service oriented system, complex applications are often required to have a high level of autonomy, even in faulty situations. In general, service-oriented compositions can be modeled as abstract business processes that are expected to be completely dependable. Process dependability can be improved by using different mechanisms. Since processes can be executed by invoking a number of available Web services, dependability requirement is often addressed by also considering the adoption of self-healing Web services. Indeed, they are able to monitor themselves, to detect failures and with actions to recover from each failure, where a failure can be either the inability to provide a given service, or a loss in the service quality. In short, self-healability combines methods and tools able to perform monitoring and repair operations. Other approaches focus on improving the dependability of a process by using different types of structural changes. The evaluation of the suitability of all the available mechanisms and thus, their selection is not a trivial task. In fact, each dependability improvement action has different properties such as its complexity, its functional and non functional properties, and the system features that are improved (or worsened) through its adoption. Taking into account all these elements, the presentation is focused on a systematic approach to support the designer in the process analysis and design phase to select the most suitable methods to adopt for the realization of a dependable service composition.

Self-Adaptive Coevolutionary Optimization Using Reorganizational Multi-Agent Systems

Gregoire Danoy (University of Luxembourg, LU)

Since the mid 1970s and the introduction of Genetic Algorithms (GAs) by John H. Holland, the idea of mimicking the capacity of biological systems to adapt to the genetic level in response to environmental challenges has motivated many research studies for applying similar mechanisms to scientific problems. One recent evolution of such algorithms, namely Coevolutionary Genetic Algorithms (CGAs), focuses on the coevolution of populations (competing or cooperating) of individuals representing specific parts of the global solution instead of evolving a population of similar individuals representing a global solution.

In this dissertation we assert that modeling CGAs as organizational multi-agent systems overcomes the lack of explicitness at the level of the algorithms structure, interactions and adaptation to existing models and platforms. We therefore introduce MAS4EVO, Multi-Agent Systems for EVolutionary Optimization, a new agent organizational and re-organizational model dedicated to evolutionary optimization. This model was used to describe existing CGAs as well as to develop new variants, including a self-adaptive competitive CGA.

We will present some results, which were obtained using MAS4EVOSs implementation named DAFO (Distributed Agent Framework for Optimization) on an existing inventory management problem for which we studied multiple static instances. We will demonstrate the improvement brought by the new self-adaptive CGA. Finally we will introduce some ongoing work on a new topology control problem in mobile wireless ad hoc networks.

Why is event correlation, anomaly detection and fault diagnosis so hard?

Gabi Dreo Rodosek (Univ. der Bundeswehr - München, DE)

Since several years, a lot of research work has been focused on the event correlation approaches and fault diagnosis. Approaches spanning from rule-based, model-based, case-based till code-book and so on have not proven to solve the problem. Why is this so hard? May be we need completely new approaches in terms of self-healing. Im my presentation I will try to point out why traditional approaches failed and may be what we can learn from the past when trying to think of self-healing approaches.

Keywords: Event correlation, fault diagnosis

Why do upgrades fail and what can be done about it?

Tudor Dumitras (Carnegie Mellon University - Pittsburgh, US)

Enterprise-system upgrades are unreliable and often result in downtime or data-loss. Errors in the upgrade procedure, such as broken dependencies, constitute the leading cause of upgrade failures. We propose a novel upgrade-centric fault model, based on data from three independent sources, which focuses on the impact of procedural errors rather than software defects. We show that current approaches for upgrading enterprise systems, such as rolling upgrades, are vulnerable to these faults because an upgrade is not an atomic operation and because it can break hidden dependencies. We also present a mechanism for tolerating complex procedural errors during an upgrade. Our system, called Imago, improves availability in the fault-free case, by performing an online upgrade, and in the faulty case, by reducing the risk of failure due to breaking hidden dependencies. Imago performs an end-to-end upgrade atomically and dependably, by dedicating separate resources to the new version and by isolating the old version from the upgrade procedure.

Fault-injection experiments show that Imago reduces the expected unavailability by 70%, compared with two existing mechanisms for online upgrades in enterprise systems, without degrading the system's throughput or response time.

Keywords: Software upgrades, fault model, reliability evaluation

Self-adaptative Systems-of-Systems

Frank Eliassen (University of Oslo, NO)

As software systems become increasingly complex, ubiquitous and distributed, a new class of systems has emerged. These Systems-of-Systems are large-scale heterogeneous systems composed from a set of independent systems to satisfy some global need across multiple domains.

The complex interdependencies and interactions inherent to systems-of-systems pose a major challenge to self-adaptation of software artefacts of such systems. These challenges pertain to heterogeneity, scalability, expressiveness and multi-modality of compositions. For example, due to the large scale adaptation of system of systems there a need for decentralized system of systems adaptation mechanisms that are able to perform adaptations independently based on information communicated by other adaptation mechanisms deployed in inidvidual systems.

Furthermore. since systems-of-systems are made of heterogeneous pieces of software using different underlying technologies, which need to be combined in order to build a coherent platform. In particular there is a need for a common framework that will include a comprehensive support for cross-layer adaptation of systems of systems This will consist of adaptation mechanisms that support the consistent adaptation of the different layers of a system of systemS, from the high level application down to the low level hardware resources.

An example of an approach for handling the heterogeneity dimension, is our development of the QuA technology agnostic adaptation framework. QuA can be used to integrate technology specific composition and adaptation mechanisms into a common adaptation middleware enabling the composition and adaptation of systems built from different underlying software technologies with technology specific adaptation mechanims.

While many solutions (frameworks, methodologies, techniques) have been developed for addressing the challenges of adaptive SoS, it would seem there is a need to address these challenges in a holistic fashion bringing together a range of sub-disciplines in software composition and adaptation to identify the requirements for dynamic adaptation of systems-of-systems, understand the extent to which existing techniques meet these requirements and formulate a roadmap for addressing the shortcomings.

This roadmap could act as a long term benchmark against which self-adaptation research can evaluate its successes when addressing challenges for self-adaptive systems-of-systems.

Keywords: Self-adaptation, systems of systems

Challenges for the Model-Driven Development of Self-Adaptive Applications

Kurt Geihs (Universität Kassel, DE)

Ubiquitous computing environments promise exciting new opportunities for self-adaptive applications that are capable to react to context changes and to exploit dynamically discovered services and information sources. However, the development of such kind of applications is a rather complex undertaking. Model-driven software engineering techniques have been shown to facilitate this task. Nevertheless there are a number of uncharted areas where we need further research and experimentation. In my presentation I will focus particularly on the open questions that need to be solved before Model-Driven Development will be widely adopted for the development of self-adaptive applications.

Keywords: Self-adaptive systems, model-driven development, ubiquitous computing

On the Role of Models for Self-Healing and Self-Adaptive Systems

Holger Giese (Hasso-Plattner-Institut - Potsdam, DE)

Models play an important role in the planning and the construction of software. In this talk we will explore which role models can play at run-time for self-healing and self-adaptive systems. We will discuss which impact models have on self-healing or self-adaptation capabilities of systems, in what manner systems can benefit from run-time models and where limitations exists, and in which respect models employed for construction of the software can also help when it comes to self-healing or self-adaptation. We will outline our findings referring to a number of research results and case studies and also identify several research challenges which have to be overcome to realize the full potential of models for self-healing and self-adaptive systems.

Keywords: Self-healing, self-adaptive, models

On the road toward self-healing datacenters

Moises Goldszmidt (Microsoft Research - Mountain View, US)

In this talk I will first present results from applying statistical modeling and machine learning methods to raw data collected from Microsoft datacenters providing 24x7 enterprise-class user facing applications, in order to:

- summarize the state of the datacenter/application for diagnosis

- classify and recognize performance problems for accurate selection of repair actions
- evaluate the effectiveness of automated closed loop actions (no-op, reboot, reimage, and human intervention) for policy evaluation.

I will then conclude with a set of challenges and opportunities on extending these methods in order to achieve the next steps toward efficient self-healing including:

- automated modeling of performance crises evolution (including human intervention) for policy creation,
- automated causal discovery for root cause analysis,
- prediction and forecasting of problems for crisis mitigation.

Towards Performance and Reliability Prediction for Self-adaptive Systems

Jens Happe (FZI Karlsruhe, DE)

In complex, heterogeneous environments, self-adapting software systems are a necessity to react on changes in the environment to ensure minimal user-perceived impact on performance and reliability. Evaluating possible adaptations for their impact on performance or reliability during runtime is a hard problem, because of the complex interdependencies in system behaviour. State-of-the-art prediction approaches mainly target static systems and are usually used during system design. In the first part of our talk, we will introduce one of these approaches, the Palladio Component Model (PCM). The introduction includes the core features of the PCM and the capabilities of its accompanying simulation tools.

In the second part, we will outline the major challenges of adaptive systems for software performance prediction and present initial ideas to enable analysing self-adaptation. We will sketch an approach to capture the performance properties of complex, heterogeneous, and adaptive systems based on systematic measurements. Furthermore, documented performance patterns can be used to specify adaptation rules, while meta-heuristic search techniques can be used to find adaptations for unplanned environmental changes. During our talk, we will use a running example to illustrate these ideas.

Joint work of: Happe, Jens; Koziolek, Heiko

Adaptive Capacity Management for Resource-efficient, Continuously Operating Software Systems

Wilhelm Hasselbring (Universität Kiel, DE)

This presentation gives an overview about our current work on a framework which aims at operating component-based software systems more efficiently.

Efficiency, in terms of the number of allocated data center resources, is improved by executing architecture-level runtime adaptations based on current workload situations. The proposed framework called SLAstic is described and open questions to be answered in future work are raised.

Keywords: Self-adaptive systems, predictive and proactive methods, large IT infrastructures

Joint work of: van Hoorn, Andre; Rohr, Matthias; Hasselbring, Wilhelm

Automatic Failure Diagnosis Support in Distributed Large-Scale Software Systems Based on Timing Behavior Anomaly Correlation

Wilhelm Hasselbring (Universität Kiel, DE)

Manual failure diagnosis in large-scale software systems is time-consuming and error-prone. Automatic failure diagnosis support mechanisms can potentially narrow down, or even localize faults within a very short time which both helps to preserve system availability. A large class of automatic failure diagnosis approaches consists of two steps: 1) computation of component anomaly scores; 2) global correlation of the anomaly scores for fault localization. In this paper, we present an architecture-centric approach for the second step. In our approach, component anomaly scores are correlated based on architectural dependency graphs of the software system and a rule set to address error propagation. Moreover, the results are graphically visualized in order to support fault localization and to enhance maintainability. The visualization combines architectural diagrams automatically derived from monitoring data with failure diagnosis results. In a case study, the approach is applied to a distributed sample Web application which is subject to fault injection.

Keywords: Predictive and proactive methods, fault detection and management, large IT infrastructures

Joint work of: Marwede, Nina; van Hoorn, Andre; Rohr, Matthias; Hasselbring, Wilhelm

Full Paper:
http://doi.ieeecomputersociety.org/10.1109/CSMR.2009.15

See also: Nina Marwede, Matthias Rohr, André van Hoorn, Wilhelm Hasselbring, "Automatic Failure Diagnosis Support in Distributed Large-Scale Software Systems Based on Timing Behavior Anomaly Correlation," csmr, pp.47-58, 2009 European Conference on Software Maintenance and Reengineering, 2009

Engineering of IT management automation along tasks, loops, function allocation, implementation method catalog

Ralf Koenig (LMU München, DE)

In IT management automation, there has long been a coexistence of ad hoc automation done by system operators and development of management software by software vendors. For upcoming IT systems with self-management capabilities two major problems can be identified: a) to be designed efficiently and effectively by using available knowledge by system operators, theoreticians and system designers and b) to be accepted and trusted by system operators.

Both of these problems can be traced to insufficient communication between operators and designers, as well as lack of structure and content of documentation at both, design time and operation time. The idea of this work is to adopt and translate principles from Systems Engineering. With this knowledge, a method called "Engineered IT Management Automation" (EIMA) is created, which step by step:

- analyzes resource-related IT management tasks,
- associates them with common loops,
- allocates task steps ("functions") along the loop to either humans or machines,
- and finally refers to applicable existing knowledge, methods and algorithms to implement the steps allocated to machines for automated execution.

This idea enables system operators and system designers to map the knowledge that both of these parties have. As the method is rooted in systems engineering, we can build on the experience that has been gathered there in the design of systems with self-management capabilities like autopilots, driver assistance systems, satellites, robots.

This work differs from related work in its cross-domain view and its focus on general patterns instead of point solutions.

Ultimately, as potential follow-up research, the method can itself be translated to development tools that support the design of systems with self-management capabilities in IT management as well as other domains with models, simulations and system assessment.

This paper was supported in part by the EC IST-EMANICS Network of Excellence (#26854).

Keywords: IT management automation, systems engineering, automation engineering, task analysis, feedback loop, function allocation, implementation methods

Towards CloudComputing@home

Derrick Kondo (INRIA Rhône-Alpes, FR)

Cloud computing platforms provide massive scalability, near-100% reliability, and speedy performance at relatively low costs for demanding applications and services. We raise the possibility of cloud computing for large-scale scientific applications and services over unreliable resources, in particular, those volunteered over the residential Internet. The motivation is the immense collective power of volunteer resources, and the near-zero amortized costs of using them.

We discuss one major challenge in supporting volunteer cloud computing, namely the assurance of collective resource availability. We propose that failure modelling and prediction techniques should partition hosts into subsets according to their statistical properties of availability.

Exmples of criteria for partitioning include randomness, the probability distribution, and predictability of resource availability. We argue that isolating subsets of resources that exhibit similar patterns of availability is critical for stochastic scheduling.

Self-Healing for Concurrent Software

Bohuslav Krena (Brno University of Technology, CZ)

The talk sums up a research done at Brno University of Technology within the EU FP6 project SHADOWS in the field of detection and self-healing of data races and atomicity violations. The talk is conluded by the challenges discovered during the research (e.g., overhead reduction or healing assurance).

Keywords: concurrent software, self-healing, bug detection, multi-core, data races, atomicity violation

Joint work of: Hruba, Vendula; Krena, Bohuslav; Letko, Zdenek; Nir-Buchbinder, Yarden; Tzoref-Brill, Rachel; Ur, Shmuel; Vojnar, Tomas

Full Paper:
http://www.fit.vutbr.cz/∼vojnar/Publications/kltuv-padtad-07.pdf

Full Paper:
http://www.fit.vutbr.cz/∼vojnar/Publications/lvk-padtad-08.pdf

See also: B. Krena, Z. Letko, R. Tzoref, S. Ur, and T. Vojnar. Healing Data Races On-The-Fly. In Proc. of 5th International Workshop on Parallel and Distributed Systems: Testing and Debugging—PADTAD'07, London, UK, 2007. ACM Press. + B. Krena, Z. Letko, and T. Vojnar. AtomRace: Data Race and Atomicity Violation Detector and Healer. In Proc. of 6th International Workshop on Parallel and Distributed Systems: Testing and Debugging—PADTAD'08, Seattle, WA, USA, pages 1–10, 2008. ACM Press.

Self Healing and Self Adaptive Software

Robert Laddaga (BBN Technologies - Cambridge MA, US)

Self adaptive software (SAS) has a stong and growing research and development community. That community has applied SAS to a wide range of applications including vision and robotics. SAS is also intrinsically a Self Healing technology, which makes extensive use of models and model based diagnosis and recovery to accomplish its self healing goals. The talk first presents the concepts, history and applications of SAS. Next it discusses the importance of contexts and models in SAS, and uses GRAVA (Grounded Reflective Adaptive Vision Architecture) as an example of SAS system. The talk ends with a description SAS current research challenges.

Keywords: Self adaptive software, self healing, model based diagnosis

Runtime Monitoring for Proactive Fault Management

Miroslaw Malek (HU Berlin, DE)

Runtime monitoring for fault diagnosis, recovery and failure prediction has become indispensable in systems with enhanced availability. Classical approaches to increase availability and dependability are too rigorous to scale up to enterprise systems and are bound to fail given the complexity and degrees of freedom of current commercial systems. We believe that a comprehensive approach to runtime monitoring, field data analysis, modeling and failure prediction, based on real-world industrial data can bridge the gap between academic approaches and industry needs by effectively avoiding failures and optimizing repair times. We propose a variable selection technique, called Probabilistic Wrapper Approach (PWA) which simplifies modeling and reduces complexity of the failure prediction algorithms. From our perspective we could gain substantial benefits by integrating selective monitoring and failure prediction approaches with optimized recovery schemes.

Keywords: Failure prediction, fault management, runtime monitoring, variable selection

Joint work of: Malek, Miroslaw; Hoffmann, Günther

Component-Oriented Behavior Extraction for Autonomic System Design

Tiziana Margaria (Universität Potsdam, DE)

Rich and multifaceted domain specific specification languages like the Autonomic System Specification Language (ASSL) help to design reliable systems with self-healing capabilities.

The GEAR game-based Model Checker has been used successfully to investigate properties of the ESA Exo- Mars Rover in depth. We show here how to enable GEAR's game-based verification techniques for ASSL via systematic model extraction from a behavioral subset of the language, and illustrate it on a description of the Voyager II space mission.

Keywords: Self-healing, model driven design, game based model checking, model extraction

Joint work of: Margaria, Tiziana; Bakera, Marco; Wagner, Christian

In-the-field Self-Healing

Leonardo Mariani (University of Milano-Bicocca, IT)

Testing and analysis techniques cannot exahustively validate software systems. Thus, faults are often lately discovered in the field, when causing failures. Self-Healing solutions can mitigate the impact of in-the-field failures by providing effective solutions to heal executions on-the-fly and in-the-field, and produce support information that facilitate debugging and bug fixing.

In this presentation, we will show two solutions for healing executions in-the-field. The first solution is a technique for healing the known integration problems that usually threat enterprise systems. The second solution is a technique for healing concurrency problems without information about the specific nature of the fault.

Keywords: Self-healing, healing integration problems, healing concurrency faults

Robustness in Query Optimization

Volker Markl (TU Berlin, DE)

We present an overview of adaptive query optimization research that we have conducted during the last 5 years and offer a critical reflection as well as areas that still need scientific exploration. In the talk, we will focus on the area of improving cardinality estimation in a query optimizer through a autonomic feedback loop as well as how to incorporate ovserved statistics into a cardinality estimation framework using entropy maximization. We will also critically discuss the risk of autonomic computing adversely affecting robustness and predictability of a computer system.

Keywords: Query processing, cardinality estimation, database systems, machine learning, feedback loop

See also: This is a summary of several papers during the last years, particularly: Volker Markl, Peter J. Haas, Marcel Kutsch, Nimrod Megiddo, Utkarsh Srivastava, Tam Minh Tran: Consistent selectivity estimation via maximum entropy. VLDB J. 16(1): 55-76 (2007) and Michael Stillger, Guy M. Lohman, Volker Markl, Mokhtar Kandil: LEO - DB2's LEarning Optimizer. VLDB 2001: 19-28

Functionality Recomposition for Self-Healing

Josu Martinez (University College - Dublin, IE)

The exposure of the Internet in modern societies and the exponentially increasing technological innovation have leaded to the creation of complex pervasive environments -trillions of inter-connected heterogeneous computing devices which form large scale systems that ceaselessly consume distributed services. These highly evolving environments introduce new levels of complexity, which require novel self-healing and self-adaptive strategies to handle consequences such as functionality unavailability at execution time while ensuring the 24/7 operation of the system.

We suggest a Functionality Recomposition for Self-Healing (FReSH) strategy to cope with functional disruptions previously mentioned. This technique consists on providing complex systems with some self-assembly capabilities to enable autonomous recomposition of unavailable functionality out of some of the components existing in the system. Each of these components is considered a reusable and shareable building block that provides a single functionality, and may be dynamically combined with other components to compose higher-level software entities equivalent to the failing or unavailable binary structures. Thus, FReSH enables the creation of robust pervasive systems through the support of flexible and dynamic architectural modifications during execution without any human intervention.

Keywords: Self-healing, self-adaptive, component-based, formal specification matching

Verification and Validation of Continuously Evolving Systems

Henry Muccini (Univ. degli Studi di L'Aquila, IT)

Modern software systems have to satisfy new and strong adaptability and extensibility requirements in order to be competitive on the market: adaptable to the users needs, and usable in diverse and heterogeneous environments (thus, extensible).

In order to cope with such a new requirements, we can observe today the born of new types of software systems (hereafter referred as continuously evolving systems), able to auto-adapt, reconfigure and change during the execution

and depending on the context. Continuously evolving systems change at run-time, frequently in an unexpected way, and depending on external agents. Such a systems are characterized by the need of evolution during execution, and without the possibility to stop the run (for example, let us think to a telecommunication system). Changes can be specified at design-time (i.e., planned from the beginning) or may happen in unexpected or unplanned ways.

The possibility of continuously integrating new components and services developed by third parties and not specifically built for running in a specific software system, introduces new and challenging issues to the verification and validation of such systems.

In particular, while the classic issue consists in validating a pre-defined configuration, in continuously evolving systems the new objective is to validate an architecture continuously evolving during execution in predictable or unpredictable ways, so to avoid that an erroneous evolution can make the system instable or faulty and for discovering faults introduced during the evolution phase.

The verification and validation of continuously evolving systems poses new challenges: i) the focus moves from validating the designed system to validating the changing overtime one. While in static systems the verification can be done once and for all before deployment, for systems changing at run-time validation becomes a perpetual activity to be performed during system execution; ii) traditional model-based analysis techniques might not be practical since it is not clear how the models would evolve at runtime and how closely the evolving models relate to and reflect information from the executing code; iii) traditional regression analysis techniques can become inefficient, due to the rapid changing of the system configuration; iv) frequent and unpredictable changes can make typical verification and validation techniques unaffordable.

Therefore in such a context, traditional verification techniques (i.e., one-time analysis of static systems) need to be complemented with run-time verification techniques that permit to control the behavior of the system during regular usage (i.e., perpetual analysis of evolving systems). In general such techniques will have to cope with many constraints deriving from run-time execution such as fast responses - in order to take countermeasures, and reduced complexity - in order to be affordable.

Goal of the ViVaCE (Verification and Validation of Continuously Evolving Systems) project is to propose new verification, validation and healing techniques for functional aspects of continuously evolving systems. Such a techniques shall complement existing ones, by keeping into primary consideration the dynamic evolution characteristic of such systems, and being applied in parallel to the development phases.

The main aim of this project is thus to study and propose new techniques for:

- "fault prevention" (i.e., avoiding failures to happen),
- "fault removal" (i.e., to identify faults when introduced in the system),
- "monitoring" (i.e., for a continuous analysis of the evolving system during its run) and

– "healing" techniques (i.e., to put the system into a correct state through system reconfiguration).

Keywords: Fault prevention and removal, monitoring, healing

Sandbox Learning: Try without error?

Christian Mueller-Schloer (Leibniz-Universität Hannover, DE)

Adaptivity is enabled by learning. Natural systems learn differently from technical systems. In particular, technical systems must not make errors. On the other hand, learning seems to be impossible without occasional errors.

We propose a 3-level architecture for learning in adaptive technical systems and show its applicability in the domains of traffic control and communication network control.

Keywords: Learning, real time, natural systems, technical systems

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2009/2123

Recovery, built-in adaptation, and the update problem: The good, the bad, and the ugly

Simin Nadjm-Tehrani (Linköping University, SE)

Architectures, methods and tools for building dependable systems have been the focus of research in distributed systems for several decades. However, big headlines due to significant failures witness that our methods perpetually face new challenges and unresolved issues. In this talk I present a few interpretations of self-healing in different application areas, suggesting that distributed algorithms with performance evaluations are part of the solution towards systems acting in dynamic environments. Building fault management in middleware, and adaptations to resource constraints in protocols is shown to be a successful recipe.

I then go on to describe a specific application area in which I find it difficult to provide an out-of-the-box solution to the fault diagnosis and recovery problem implemented in software.

It appears that results in software engineering, AI, real-time and embedded systems, and dependability research need to be combined to address self healing requirements in this automotive related problem.

Keywords: Dependability, system and software modelling, fault detection and management

Automated Online Fingerpointing in Large Distributed Systems

Priya Narasimhan (Carnegie Mellon University - Pittsburgh, US)

Problem diagnosis (or fingerpointing) involves instrumenting systems to yield meaningful data, automatically detecting errors, failures and anomalous behavior within these systems, and ascertaining their root-cause, i.e., the underlying fault. Fingerpointing is difficult because the distributed interactions, protocols and inter-component dependencies in computer systems can cause a problem to change "shape" or manifestation, leading to potential red herrings in problem determination.

There can be many root causes of an outward manifestation of a problem and there might be insufficient information to distinguish between the various root causes. On the other hand, too much monitoring and too many error messages might overwhelm the system, obscure the root cause, and lead to increased latencies and additional resource costs.

This talk describes the data and visual analysis techniques that we have developed for automated fingerpointing in distributed systems such as MapReduce (Hadoop), PVFS, Lustre, amongst others – the aim in all these cases was to perform online and offline root-cause analyses in order to identify a problem node/process, to diagnose the source of the problem, and report it to the user or administrator in a meaningful/useful manner.

This talk will cover both the black-box and white-box problem-diagnosis techniques that we have developed so far, and will highlight our lessons learned and experiences with these systems.

Keywords: Failure diagnosis, MapReduce, storage systems, root-cause analysis

Joint work of: Narasimhan, Priya; Tan, Jiaqi; Kavulya, Soila; Pan, Xinghao; Gandhi, Rajeev

Full Paper:
http://www.ece.cmu.edu/~fingerpointing

Monitoring Architectural Properties in Dynamic Component-Based Systems

Andrea Polini (Università di Camerino, IT)

Modern systems are increasingly required to be capable to evolve at run-time, in particular allowing for the dynamic plugging of new features. It is important that this evolution happens preserving some established properties (which can concern the structure, the interaction patterns, or crucial extra-functional properties, such as reliability or security), and due to dynamicity this needs to be checked at run-time, as the changes occur.

In this work we consider evolving component-based systems formed by a kernel architecture to which new components can be plugged in at run-time, and introduce the MOSAICO approach for the run-time monitoring of architectural properties. MOSAICO uses Aspect-oriented technologies for instrumenting and monitoring the system according to selected architectural properties. MOSAICO can handle evolving black-box component systems since it continuously watches the events occurring at the extension points of the kernel architecture. The application of a prototype implementation of MOSAICO, capable to handle interaction pattern properties, is illustrated on the NewsFeeder case study.

Keywords: fault detection, evolving systems

Joint work of: Polini, Andrea; Muccini, Henry

Full Paper:
http://www.springerlink.com/content/l382755n01543124/?p=f5aee9201f3c4d9582ec9dcfbcaf7f8c&pi=8

Context modelling and reasoning for adaptive applications in ubiquitous computing environments

Roland Reichle (Universität Kassel, DE)

Ubiquitous computing environments provide new opportunities, but also imply a number of challenges for the development of context-aware adaptive applications. For example, services available in the environment can be exploited to enhance application functionality, but at the same time, applications are required to adapt to the dynamically changing execution context. This requires context modelling and reasoning approaches which have to be tailored to the characteristics of ubiquitous computing environments as well. Context providers (or reasoners) may appear and disappear in the environment and may provide imprecise or even unreliable information. In addition, heterogeneity adds to the complexity due to independent development of applications and context elements. In this talk, we present a new context modelling and reasoning approach that faces the challenges of ubiquitous computing environments and focuses on the exchange of and reasoning about heterogeneous, imprecise and uncertain information.

Autonomous storage management for personal devices with PodBase (talk by Ansley Post, MPI-SWS)

Rodrigo Rodrigues (MPI für Software Systeme - Saarbrücken, DE)

People use an increasing number of personal electronic devices such as notebook computers, PDAs, MP3 player, and mobile phones in their daily lives.

Making sure that the data stored on these devices is available where needed and backed up regularly represents a time-consuming and error-prone burden on users. In this talk, I present PodBase, a system that automates storage management on personal devices. PodBase ensures the durability of data despite the loss or failure of a subset of devices; at the same time, PodBase aims to make sure that data is available on devices where it is useful. The system takes advantage of unused storage resources and pairwise connectivity between devices to propagate the system state and replicate files.

PodBase must be able to operate under a wide range of conditions and without attention from expert users. Towards this goal, PodBase uses a linear programming solver to compute an optimal replication plan based on the current distribution of files, availability of storage, and the likelihood of future device connections. Whenever two devices connect, PodBase computes a new plan and executes its first step (which can be viewed as a move in a game between the system and its environment). Results from a small prototype deployment with real users show that, under a wide range of conditions, this approach allows PodBase to maximize the durability and availability of data stored on personal devices with minimal user attention.

Keywords: storage

Joint work of: Ansley Post et al.

Failure Prediction for Proactive Fault Management

Felix Salfner (HU Berlin, DE)

The ever increasing complexity of contemporary computer systems puts traditional fault tolerance techniques to their limits. Major reasons for this are increased configurability, dynamicity by ongoing updates/upgrades and the use of off-the-shelf components, which leads to the situation that runtime behavior can be significantly different from expected behavior at design time. One approach to address this problem is to continuously monitor the system and to assess its runtime state in order to predict whether any fault might evolve into a failure. In case of an upcoming failure a remedy for the problem is applied in order to either prevent the upcoming failure, to limit its effects, or to prepare repair mechanisms such that time-to-repair can be reduced. We call this approach *Proactive Fault Management*.

One of the key techniques for proactive fault management is the prediction of upcoming failures from monitoring data. In order to address the lack of detailed knowledge about the internal behavior of a specific computer system statistical learning theory is applied, which provides means to infer system behavior from measurement data with little a-priori knowledge. It also enables to adapt the failure prediction algorithm to changing system behavior over time. This talk presents a failure prediction algorithm based on a hidden semi-Markov model

(HSMM). Unlike the majority of failure prediction algorithms, the HSMM predictor operates on patterns of error events that occur in the system. Its key notion is to identify event patterns that are similar to patterns that have previously led to a failure. A case study analyzing data of a commercial telecommunication system is presented. Results show that the HSMM failure predictor achieves significantly improved prediction accuracy in comparison to best-known event-based failure prediction approaches.

Keywords: Failure prediction, proactive fault management, Hidden Semi-Markov Model

Generic techniques for self-healing distributed systems

Benjamin Satzger (Universität Augsburg, DE)

The growing complexity of distributed systems demands for new ways of control. This presentation proposes generic concepts and algorithms to build self-healing systems:

- The detection of node failures in distributed environments is a non-trivial problem. A new flexible failure detection algorithm is proposed.
- Grouping algorithms can be used to allow for an autonomous installation of scalable monitoring relations in complex large scale distributed systems.
- A failure recovery engine based on automated planning, which manages a distributed system according to user-defined objectives, is proposed. It is able to generate and execute plans to autonomously recover a system from unwanted states.

Organic Computing - a Generic Approach to Controlled Self-organization in Adaptive Systems

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

Organic Computing is a visionary concept and paradigm for the design and management of complex technical systems, addressing the need for adaptive, self-organising systems which are capable of dealing with changing requirements and unanticipated situations in a robust and trustworthy way and which allow an external "observer" or "user" to interfere and influence the system whenever it does not show acceptable behaviour by itself. Research in this area is supported by a priority research program of the German Research Foundation (www.organic-computing.de/SPP), running from 2005 to 2011. The talk will give a brief introduction into some major concepts of Organic Computing.

Organic Energy Management - controlled and self-organized adaptive demand side management in the energy system

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

The growing demand for an increased efficiency of the energy system has led to several research projects on the intelligent use of information and communication technology in the energy system. A major concern is the reduction of the need for (expensive) balancing power which has to be used by the balancing group manager whenever there are significant imbalances of energy supply and usage in the balancing groups and balancing zones. We present an approach for a self-organised adaptive energy management which is using the degrees of freedom that are present in a smart home environment: energy consumers like fridges, freezers, air-conditioners, or washing machines and energy generators like combined heat and power generators allow for a dynamic adjustment of energy consumption in response to fluctuations in energy generation which are typical for photo voltaic cells or wind-based power generators. Our approach is using a pool of smart devices which are aware of their energy needs and react to requests from the Balancing Group Manager by locally optimizing their plans for energy usage.

Joint work of: Schmeck, Hartmut; Kamper, Andreas

Self-organizing QoS-Management in Service Oriented Architectures

Markus Schmid (FH Wiesbaden, DE)

Traditional hierarchical Service Level Management (SLM) frameworks fail to cope with the challenges imposed by the runtime dynamics and loose coupling of components in Service Oriented Architectures (SOA).

To support QoS management aspects throughout the SOA lifecycle (design, implementation, test, production), we developed methods and tools to support model-based monitoring of system performance and response times consistently across different SOA services. We see monitoring and manageability as vital aspects of a SOA service that need to be considered already from service design time. A MDSD-based approach is used to add performance-monitoring to newly designed services, for existing services, middleware instrumentation and semi-automatic source-code instrumentation are supported.

Based on these monitoring capabilities, we introduce a decentralized QoS management approach for SOA services. The approach uses self-management techniques to realize a flexible and modular SLM system. Coordination between different management components is based on principles adopted from the domain of organizational theory: the system uses auctions as a way to stimulate

collaboration and to optimize overall quality of service in large-scale environments.

Local enforcement of QoS constraints is supported by our modular "self-manager" framework: implementation-specific QoS manager components are assigned to individual SOA services and control the service implementation in order to enforce agreed-on Service Level Agreements (SLAs). In addition resource managers control the underlying virtual machine infrastructure. QoS manager collaboration can result in a request for resource managers to manipulate virtual-machine runtime parameters in order to prioritize selected services.

Software Self-Healing: Towards Generic, Industry-Grade Solutions?

Onn Shehory (IBM - Haifa, IL)

Large and complex software systems are inherently error-prone. Software self-healing is an emerging approach to addressing this problem. Software self-healing solutions pre-sented to date commonly address a single class of problems. Generic self-healing solu-tions are usually not applicable in industrial systems. Recently, some attempts were made to address the need for industry-grade, generic software self-healing. We will present a few cases of such solutions and discuss their properties. We will try to understand whether those indeed advance towards generic solutions applicable in the field, and what is still required to meet that goal.

Keywords: Software self-healing

Self-healing and Self-adaption in the Internet - Does "Economic Traffic Management" Belong to this Self-* Class of Mechanisms?

Burkhard Stiller (Universität Zürich, CH)

Self-healing and self-adaption determine important aspects of systems, especially of the Internet, considered as a distributed system. Thus, there comes up a structural question to be posed, which investigates about the correlation between self-adaption and Economic Traffic Management (ETM) mechanisms in a distributed system: Does "Economic Traffic Management" belong to this class of self-healing and self-adaptive mechanisms?

This talk outlines very briefly those relationships between self-organization (the umbrella term for self-* mechanisms) and ETM, discusses the SmoothIT Information Service (SIS) proposal, and its evaluation of its first implementation results on EMANICSLab.

While the talk concludes that based on the context of input variables available and defined, the set of ETM mechanisms are belonging to the class of self-* or not, the question can be answered as "it depends"!

Therefore, this talk opens the discussion to a general suitability of such mechanisms in tomorrows networks, again, especially for the Internet of a residential customer and the Internet of commercial customers.

Keywords: Self-adaption, Economic Traffic Management (ETM), Internet, distributed systems

A Virtualisation Architecture for Many-cores with Real-time Constraints and Self-X Support

Theo Ungerer (Universität Augsburg, DE)

The short talk shows the reliability problems that are envisioned for future many-core chips and proposes a virtualisation architecture that includes self-x support to solve the envisioned problems. Moreover, the potential application of the virtualisation architecture in hard real-time domains as e.g. automotive, aerospace and space applications is discussed.

Keywords: Virtualisation, many-cores, self-x

Cost-Aware and Adaptive Modeling and Monitoring for Problem Determination

Paul A. S. Ward (University of Waterloo, CA)

While there has been significant work on modeling and monitoring systems for problem determination, much of it is cost-oblivious; it either presumes complete availability of monitoring data at zero cost, or that whatever is available is sufficient both for detection and diagnosis. In this talk I will describe our work at Waterloo on adaptive modeling and monitoring that takes costs into account; I will describe the challenges our approach must overcome, including, when modeling metric data, heteroscedasticity, evolving models, model reliability, etc. I will show how we apply information theory and statistical techniques to address these challenges. Error-detection and fault-diagnosis approaches will be described, as well as the benefits of our approach in web-commerce systems. Time-permitting, I will discuss the implications of system-structure and fault knowledge, model adaptation, levels of abstraction (component, machine, data center), and front-end visualization for administrators.

How much is self healing worth?

John Wilkes (Google Inc. - Mountain View, US)

Self-healing is not cost-free: it typically comes at some cost, either in resources, control complexity, or responsiveness. Blindly making all systems self-healing is thus likely to be counter-productive, causing unwanted wastage and delays, and occasional customer dissatisfaction. This can be ameliorated if a service's clients are able to specify how much self-healing the service does on their behalf, allowing their intents to drive the service's behavior. Result: happier customers; better services. I'll talk about the use of Service Level Agreements to express these ideas.

Keywords: SLA

The day when decision support systems adapt to the failings of humans and start making decisions

Elaine Wong (EADS - Singapore, SG)

Decision support systems have been around for half a century and they have become increasingly sophisticated in their core functions, namely managing access to information, processing large amounts of data, and manipulating models of a particular domain. Because of these advancements, decision support systems have been implemented for numerous (and possibly all) industries - from banking, agricultural, healthcare, logistics to politics.

While access patterns, data structures and models get more complicated, another factor (which is less talked about but equally important) is also evolving - that is the failings of humans working with such systems. The recent economic crises reemphasize the fact that even the best leaders make the occasional bad call with potentially fatal repercussions. There is therefore a need for processes that can balance or perhaps overcome the innate limitation of the biased human mind.

In this discussion, I would like to begin on the premise that decision support systems can learn the failings of the human decision making process, and in so doing, start making intelligent decisions that can help humans make better decisions. The question however is: what framework should govern the creation of such systems so as to make their proposals and actions dependable?

Programs that fix themselves

Andreas Zeller (Universität des Saarlandes, DE)

Many software problems have embarrassingly habitual causes - a misnamed variable, an omitted function call, a loop executed one time too many or too few, the mishandling of border cases.

In this work, we conjecture that one can automatically synthesize corrections to these problems: If a program fails, we search for simple corrections that make the failure go away. The AutoFix approach we propose is robust, relying on runtime checks and regression tests to locate the error and to avoid introducing new ones. It is also fully automatic, which allows us to start it as soon as a program fails, effectively repairing the program while it is still executing.

Joint work of: Dallmeier, Valentin; Zeller, Andreas

Can we fix it?

Andreas Zeller (Universität des Saarlandes, DE)

Past years have seen several advances in defect localization - the problem of suggesting a good fix location for a given failure. We leverage *differences in program behavior* to generate the fix directly. Our PACHIKA tool automatically summarizes executions to object behavior models, determines differences between passing and failing runs, generates possible fixes, and assesses them via the regression test suite. Evaluated on the ASPECTJ defect history, PACHIKA generates a valid fix for 3 out of 18 crashing bugs; each fix pinpoints the defect location and passes the ASPECTJ test suite.

Keywords: Program analysis, automated debugging, automatic fixes

Joint work of: Dallmeier, Valentin; Zeller, Andreas