08031 Abstracts Collection
Software Engineering for Self-Adaptive Systems
— Dagstuhl Seminar —

Betty H. C. Cheng[1], Holger Giese[2], Paola Inverardi[3], Jeff Magee[4] and Rogerio de Lemos[5]

[1] Michigan State University, USA
`chengb@cse.msu.edu`
[2] Hasso-Plattner-Institut, Potsdam, D
`holger.giese@hpi.uni-potsdam.de`
[3] University of L'Aquila, I
`inverard@di.univaq.it`
[4] Imperial College London, GB
`jnm@doc.ic.ac.uk`
[5] University of Kent, GB
`R.Delemos@kent.ac.uk`

Abstract. From 13.01. to 18.01.2008, the Dagstuhl Seminar 08031 "Software Engineering for Self-Adaptive Systems" 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. Software engineering, requirements engineering, modelling, evolution, assurances, self-adaptability, self-organization, self-management

Software Engineering for Self-Adaptive Systems: A Research Road Map

Software's ability to adapt at run-time to changing user needs, system intrusions or faults, changing operational environment, and resource variability has been proposed as a means to cope with the complexity of today's software-intensive systems. Such self-adaptive systems can configure and reconfigure themselves, augment their functionality, continually optimize themselves, protect themselves, and recover themselves, while keeping most of their complexity hidden from the user and administrator. In this paper, we present research road map for software engineering of self-adaptive systems focusing on four views, which we identify as essential: requirements, modelling, engineering, and assurances.

Keywords: Software engineering, requirements engineering, modelling, evolution, assurances, self-adaptability, self-organization, self-management

Joint work of: Cheng, Betty H.C.; de Lemos, Rogerio; Giese, Holger; Inverardi, Paola; Magee, Jeff

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2008/1500

Learning Libraries

Jesper Andersson (Växjö University, S)

I will present results from a case-study where adaptive libraries or middle-wares are used to realize specific quality attributes such as availability and scalability. The results of this study show that developers and users of these libraries look for techniques that will make the products more flexible and less application specific. To address the flexibility concern, we propose a technique with statically and dynamically configurable libraries which can "learn" which service combination to use in a specific context for a specific application.

Loose Compositions for Autonomic Systems

Luciano Baresi (Politecnico di Milano, I)

Autonomic computing is one of the most promising techniques to manage the complexity of modern software systems. It fosters the idea of systems able to autonomously detect changes and anomalies that might hamper their effectiveness and operation. Supervision and actual business logic are intertwined and work together to supply the autonomic features.

This presentation illustrates our ongoing work on loose compositions to provide an autonomic framework based on Java, aspect oriented programming, and rules. The presentation illustrates how we exploit loose compositions and clustering to make the different application components cooperate in a fully autonomic way. It also presents a prototype implementation, based on the DIET agent framework.

Dynamically Adaptive Systems supported by Reflective Component-based Technologies

Nelly Bencomo (Lancaster University, GB)

I would like to give a talk about our experience developing dynamically adaptive applications supported by reflective middleware platforms at Lancaster. The talk takes into account the crucial role of architecture, (dynamic) variability, models to describe runtime concerns, and reflection.

Some short references of joint research work with other research groups will be shown and, finally a brief agenda of our future research is also included. This talk may take 20 min (if allowed).

The talk looks to me related to topics 2 (High-Level Design), and 5 (Models) and perhaps 7, specifically the sub-topic dynamic variability.

Keywords: Architecture, (dynamic) variability, models@runtime, reflection

Building Biologically-Inspired Self-Adapting Systems

Yuriy Brun (USC - Los Angeles, USA)

Biological systems are far more complex than systems we design and build today. The human body alone has orders of magnitude more complexity than our most-intricate designed systems. Further, biological systems are decentralized in such a way that allows them to benefit from built-in error-correction, fault tolerance, and scalability. It follows that if we can extract certain properties of biological systems and inject them into our software design process, we may be able to build complex self-adaptive software systems.

Biological systems' complexity makes them not only desirable to guide software design, but also difficult to fully understand. Thus one approach to building software similar to biological systems is by first building models of biology that we can understand. Then these models can guide the high-level design, or architecture of the software systems, resulting in systems that retain the model's fault tolerance, scalability, and other properties.

I present a general outline of how one might use biology to create a model to guide the architecture of a software system, and develop one such model and the resulting architectural style, the tile style, for computational systems that can use a large distributed network of computers, such as the internet, to solve computationally-intensive problems in a discreet, fault-tolerant, and scalable manner.

Keywords: Biologically-inspired, software architecture, fault tolerance, scalability

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2008/1499

Multiple Concern Adaptation for Runtime Composition in Context-Aware Systems

Carlos Canal (Univ. de Malaga, E)

Context-Aware computing studies the development of systems which exploit context information (e.g., user location, network resources, time, etc.), which is specially relevant in mobile systems and pervasive computing.

When these systems are built assembling pre-existing software components (COTS), the composition process must be able to solve potential interoperability problems, adapting component interfaces. In addition, the composition must be adapted to the execution conditions of such systems, which are likely to change at run-time, affecting component behaviour. This work presents an approach to the flexible composition of possibly mismatching behavioural interfaces in systems where context information can vary at run-time. Our approach enables composition at runtime, enabling dynamic changes in composition according to context changes. Furthermore, our approach simplifies the specification of composition/adaptation by keeping Separation of Concerns, and is able to handle context-triggered adaptation policies.

Keywords: Component-based Software Development, Run-time Composition, Model-based Adaptation

Joint work of: Canal, Carlos; Cámara, Javier, Salaün, Gwen

See also: Proc. of the 4th International Workshop on Formal Aspects of Component Software (FACS'07), ENTCS, Elsevier, 2008, (to appear)

Model Driven Engineering of High-Assurance Adaptive Systems ... Harnessing Digital Evolution to Generate the Software Models

Betty H. C. Cheng (Michigan State University, USA)

In order to support the model-driven engineering of high-assurance adaptive systems, we need automated techniques to generate innovative software models that satisfy safety properties.

We describe an automated method to generate state diagrams for a set of interacting objects, including the extension of an existing model to support new behavior. The approach is based on *digital evolution*, a form of evolutionary computation that enables a designer to explore an enormous solution space for complex problems. In our application of this technology, an evolving population of *digital organisms* is subjected to natural selection, where organisms are rewarded for generating state diagrams that support key scenarios and satisfy critical properties as specified by the developer.

To achieve this capability, we extended the Avida digital evolution platform to enable state diagram generation, and integrated Avida with third-party software engineering tools, e.g., the Spin model checker, to assess the generated state diagrams. We are also exploring how Avida can be used to generate formal properties from the software models in order to discover latent requirements.

See IEEE Computer, Jan 2008 for an overview of our objectives with Harnessing Digital Evolution for Software Design.

Keywords: Model-driven engineering, high-assurance systems, automatic UML model generation, digital evolution

Validating an Autonomous Adaptive System: Why's and How's

Bojan Cukic (West Virginia Univ. - Morgantown, USA)

The goal of our research has been the development of practical methods for validation of adaptive safety critical flight control systems. This presentation overviews the developed techniques, summarizes results and discusses the open research questions.

The presentation is related to topic areas 4 and 8. Time permitting, this can be a 30 minutes talk.

Keywords: Software validation, adaptive systems, convergence and stability, novelty detection, neural network adaptation, intelligent flight control

Monitoring techniques for an online neuro-adaptive controller

Bojan Cukic (West Virginia Univ. - Morgantown, USA)

The appeal of biologically inspired soft computing systems such as neural networks in complex systems comes from their ability to cope with a changing environment. Unfortunately, adaptability induces uncertainty that limits the applicability of static analysis to such systems. This is particularly true for systems with multiple adaptive components or systems with multiple types of learning operation.

This work builds a paradigm of dynamic analysis for a neuro-adaptive controller where different types of learning are to be employed for its online neural networks. We use support vector data description as the novelty detector to detect unforeseen patterns that may cause abrupt system functionality changes. It differentiates transients from failures based on the duration and degree of novelties. Further, for incremental learning, we utilize Lyapunov functions to assess real-time performance of the online neural networks. For quasi-online learning, we define a confidence measure, the validity index, to be associated with each network output. Our study on the NASA F-15 Intelligent Flight Control System demonstrates that our novelty detection tool effectively filters out transients and detects failures; and our light-weight monitoring techniques supply sufficient evidence for an insightful validation.

Keywords: Neural network, Adaptive system, Run-time monitoring, Dynamic cell structure, Support vector data description, Validity index

Joint work of: Liu, Yan; Cukic, Bojan; Fuller, Edgar; Yerramalla, Sampath; Gururajan, Srikanth

See also: Y. Liu, B. Cukic, S. Gururajan, Validating Neural Network-based Online Adaptive Systems: A Case Study, Software Quality Journal, Vol. 15, No. 3, (2007), pp. 309-326., 3. Y. Liu, B. Cukic, E. Fuller, S. Yerramalla, S. Gururajan, "Monitoring Techniques for an On-Line Neuro-Adaptive Controller," Journal of Systems and Software, Vol. 79 (2006), pp. 1527-1540.

Mapping Control Theory Concepts to Software Engineering

Bojan Cukic (West Virginia Univ. - Morgantown, USA)

Control theoretic approaches to system convergence and stability analysis have been studied for almost a 100 years. While not applicable in their original form (because software systems cannot be concisely described as dynamical systems) the goals of stability analysis for adaptive software systems are a close match. The presentation describes basic principles of stability analysis and shows examples of its application in the validation of an intelligent flight control system.

Keywords: Reactive systems, software validation

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

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

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

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.

Towards Context-based Autonomic services

Schahram Dustdar (TU Wien, A)

We live in a world of multi-dimensional globalization in the domains of business, technology, and organizational (team) forms. The interdependence of computing devices, their underlying information systems, as well as processes involving humans become more complex, thus requiring business processes and software services to attain higher degrees of autonomic and self-adaptive behavior.

In this talk I discuss some of the main challenges of building the required novel conceptual abstractions as well as needed technological implementations we have been investigating so far.

Keywords: Autonomic Service composition, Context-based service composition

Requirements Reflection

Anthony Finkelstein (University College London, GB)

I will present a short research agenda in the area of providing requirements information at run-time. I will try and link work on requirements specification, monitoring and context sensitive adaptation. This falls within Topic 1

Generic Mechanism for Enabling Dynamic Reconfiguration

Cristina Gacek (Newcastle University, GB)

In this talk I propose the use models related to describing product line architectures in order to generalize reconfiguration policies to be application independent. This combination allows the application specific concerns to be restricted to the architectural models, such that reconfiguration policies can be described and manipulated in an application independent fashion.

Topics it relates to: (2) High-level design; (3) Design & implementation; (5) Models; (7) Relation to others (Product line architectures)

Keywords: Reconfiguration, domain models

Architectural Patterns for Self-Optimizing Mechatronic Systems

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

Future generations of advanced mechatronic systems are expected to behave smarter than today's systems by exploiting local processing power and local networking capabilities to self-optimize their behavior and realize otherwise not possible functionality by means of self-adaptation and selfcoordination.

We present in this talk a set of architectural patterns developed in the context of the Mechatronic UML approach which can be employed to support self-adaptation and selfcoordination system development. We compare the architectural patterns with the reference architecture for self-managed and adaptive systems introduced by Jeff Kramer and Jeff Magee in their Future of Software Engineering paper in 2007.

Verification of Self-Adaptive and Self-Coordinating Systems

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

Future generations of advanced mechatronic systems are expected to behave smarter than today's systems by exploiting local processing power and local networking capabilities to self-optimize their behavior and realize otherwise not possible functionality by means of self-adaptation and selfcoordination.

However, today no solution for the systematic development and verification of the outlined future generation of intelligent, distributed, embedded systems exists which can also guarantee their safety. We present in this talk how the architectural patterns available in the Mechatronic UML approach enable several advanced verification techniques and their composition. The talk will focus on specific verification techniques which have been developed to address self-adaptive and self-coordinating behavior and their proper combination to verify crucial safety properties for rather complex mechatronic systems with self-adaptive and self-coordinating behavior.

An architectural approach to QoS-aware adaptation for the SOA domain: some modeling issues

Vincenzo Grassi (Università di Roma, I)

Our starting point is the consideration that one of the main motivations for self-adaptation is to deal effectively with the task of fulfilling the functional requirements of a complex system while at the same time meeting some extra-functional (QoS) requirements, despite changes in the system environment and operating conditions. Then, in this short talk (15 min.) we intend to discuss some modeling issues that should be tackled to build a QoS-aware self-adaptive system. In particular, we present our ideas in the framework of the architectural model proposed by Kramer and Magee (FOSE 2007) for self-adaptive systems, focusing our analysis/discussion on the domain of service-oriented architectures.

Joint work of: Grassi, Vincenzo; Mirandola, Raffaela

From Hell to Heaven - Design Patterns for Self-Adaptive Systems

Ethan Hadar (CA Inc. - Yoneam, Israel, IL)

This position paper raises questions and encourages research that will define an architecture-centric-evolution cookbook for self-adaptive systems. It should focus on a set of design patterns, ontology, and a methodology that encompasses it all. These patterns should relay on key performance indicators (KPI) that will indicate the need for adaptation by sensing and measuring the environment. Moreover, remedy techniques, which are transient in nature, should be aggregated and analyzed using dashboards. The adaptation activation approach should be local and system based, thus, leading for complex event systems adaptation management.

Keywords: Design patterns, self adaptation, architecture centric evolution, CEP, Quality of Service

Self-Adapting Applications for Mobile Users in Ubiquitous and service-oriented Computing Environments

Svein Hallsteinsen (SINTEF ICT - Trondheim, N)

The MUSIC project is concerned with building applications capable of performing well in the highly dynamic computing environments resulting when users carrying handheld networked devices move around in ubiquitous and service-oriented computing environments. We build on results from the MADAM project which has developed tools and middleware for the development of context aware and self-adapting distributed applications based on component oriented development, context monitoring, adaptation decision making using property predictor and utility functions, and adaptation by dynamic (re)composition.

This talk will present research challenges involved in extending the MADAM model to tackle ubiquitous and service-oriented computing environments, and some preliminary results. In such computing environments applications typically depend on external services available either through the internet or more directly from specialised devices in the environment. There will typically be several alternative providers of a needed service differing in the offered quality of service. Also available providers come and go, mostly caused by movement, but also by hardware and software failures and maintenance activities in the environment.

The following issues will be discussed:

- Introducing services in the MADAM model
- Service description and discovery
- Planning with service offers
- Monitoring SLA conformance
- Property prediction
- Scalability

Context-oriented Programming for Dynamic Software Adaptation

Robert Hirschfeld (Hasso-Plattner-Institut - Potsdam, D)

Most if not all software systems need to be changed after their initial deployment, possibly on demand, at runtime, without disruption of service. Adjustments or extensions need to be made to reflect new perspectives on domain concepts, to deal with changed requirements, or to just offer a pleasant user experience.

Context-oriented Programming, or COP, provides programmers with dedicated abstractions and mechanisms to concisely represent behavioral variations that depend on execution context. By treating context explicitly, and by directly supporting dynamic composition, COP allows developers to better express software entities that adapt their behavior late-bound at runtime.

Keywords: Context-oriented Programming, COP, Dynamic Software Adaptation, DSA, Dynamic Aspect-oriented Programming, DAOP, AOP

Full Paper:
http://www.swa.hpi.uni-potsdam.de/cop/

Software of the Future Is the Future of Software?

Paola Inverardi (Universitá di L'Aquila, I)

Software in the near ubiquitous future (Softure) will need to cope with variability, as software systems get deployed on an increasingly large diversity of computing platforms and operates in different execution environments. Heterogeneity of the underlying communication and computing infrastructure, mobility inducing changes to the execution environments and therefore changes to the availability of resources and continuously evolving requirements require software systems to be adaptable according to the context changes. Softure should also be reliable

and meet the user's performance requirements and needs. Moreover, due to its pervasiveness, Softure must be dependable, which is made more complex given the highly dynamic nature of service provision. Supporting the development and execution of Softure systems raises numerous challenges that involve languages, methods and tools for the systems thorough design and validation in order to ensure dependability of the self-adaptive systems that are targeted. However these challenges, taken in isolation are not new in the software domain. In this paper I will discuss some of these challenges, what is new and possible solutions making reference to the approach undertaken in the IST PLASTIC project for a specific instance of Softure focused on software for Beyond 3G (B3G) networks.

Keywords: Adaptable systems, ubiquitous computing, Beyond 3G (B3G) networks

Full Paper:
http://www.springerlink.com/content/8213438275082440/

See also: Ugo Montanari, Donald Sannella, Roberto Bruni (Eds.): Trustworthy Global Computing, Second Symposium, TGC 2006, Lucca, Italy, November 7-9, 2006, Revised Selected Papers. Lecture Notes in Computer Science 4661 Springer 2007, ISBN 978-3-540-75333-9

Model-based dynamic architectures in self-adaptive systems

Gabor Karsai (Vanderbilt University, USA)

Self-adaptive systems necessitate reflection and reasoning to achieve self-adaptivity and there are many different ways to implement these. In this talk we focus on a specific class of self-adaptive, component-based systems, where the adaptation is limited to activating/deactivating components and changing their interconnection patterns. We propose to express such architectural changes as structural alternatives with constraints influencing their applicability, and then the self-adaptation could be viewed as a constraint-guided, design-space exploration problem. The resulting meta-architecture implicitly represents all architectural variants, and the one fitting to the current situation could be selected via a search process. The talk will present some initial ideas and techniques for implementing these concepts.

Keywords: Model-based systems, design-space exploration, dynamic architectures

Joint work of: Karsai, Gabor; Neema, Sandeep; Sztipanovits, Janos

Legal Issues of Self-Adaptive Systems

Holger M. Kienle (University of Victoria, CA)

Legal issues are a concern that crosscuts all kinds of software systems. Example are copyright issues for the World Wide Web and privacy and security laws that govern heath care systems. Consequently, early identification of legal issues can prevent costly changes later on. This work focuses on one particular legal issue, data protection, in the context of self-adaptive systems.

Self-adaptive features can be found in ambient intelligence, autonomous agents, and e-commerce systems. There is a concern for data protection in such systems if data or information about users is collected, processed, or transmitted. This is typically the case if the self-adaptive functionality is related to user profiling and personalization.

Self-adaptive systems have to address data protection laws in order to avoid potential legal implications as well as losing trust with users. Examples of laws that may affect self-adaptive features are the EU's data protection directive (95/46/EC), telecommunications directive (97/66/EC), and data retention directive (2006/24/EC). This work identifies legal issues related to EU directives in the areas of user consent, automated decision-making, and data transfers that cross national boundaries.

Keywords: Data protection, privacy, EU directive, self-adaptive systems, profiling, personalization

Model Driven Self-Optimization

Marin Litoiu (IBM Toronto Lab. - Markham, CA)

The talk will discuss the role of performance models in self-optimization. A performance model must be accurate and adapt when the underlying system changes. We show how to combine design and run time knowledge to build and automatically tune and maintain the performance model.

The performance model can then be used in feed-back and feed-forward self-optimzation schemes. The talk will also show research results in model based self-optimization for vertically and horizontally scalable applications.

The talk fits into topic 8, and the subtopic is autonomic computing. The talk is 20 min long.

Keywords: Autonomic computing, performance models, self-optimization

Self-Managed Systems: an Architectural Challenge

Jeff Magee (Imperial College London, GB)

A self-managed software architecture is one in which components automatically configure their interaction in a way that is compatible with an overall architectural specification and achieves the goals of the system.

The objective is to minimise the degree of explicit management necessary for construction and subsequent evolution whilst preserving the architectural properties implied by its specification. This paper discusses some of the current promising work and presents an outline three-layer reference model as a context in which to articulate some of the main outstanding research challenges.

An Extensible Framework for Autonomic Analysis and Adaptation of Software Deployment Architectures

Sam Malek (George Mason Univ. - Fairfax, USA)

I will be able to give a 30 min talk on topics 2 and 3:

A distributed software system's deployment architecture can have a significant impact on the system's properties, which depend on various system parameters, such as network bandwidth, frequencies of software component interactions, and so on. Recent studies have shown that the quality of deployment architectures can be improved significantly via active system monitoring, efficient estimation of the improved deployment architecture, and system redeployment. However, the lack of a common framework for improving a system's deployment architecture has resulted in ad hoc solutions. In this paper, we present an extensible framework that guides the design and development of solutions to this problem, enables the extension and reuse of the solutions, and facilitates autonomic analysis and adaptation of a system's deployment architecture.

Model-Based Design-Unpolished

Pieter J. Mosterman (The MathWorks Inc. - Natick, USA)

This presentation provides a number of widely varying views into the overall use of computational models for control system design, concentrating on the functional part. Specific aspects that are discussed include requirements engineering and control system architectures with particular attention dedicated to adaptive, switched, and hierarchical control. Different approaches to the design of a control law are demonstrated. Stages in the functional control design are briefly sketched and related to requirements engineering. The gathering of technologies and methodologies from control system design are intended to allow drawing parallels with the field of self-adaptive systems.

Keywords: Adaptive control, model-based design, control systems

Intensive Instrumentation and Monitoring for Adaptive Ecosystems

Hausi Müller (University of Victoria, CA)

In this talk, we argue that the next generation of software systems must support continuous evolution effectively (i.e., software systems are under constant development, can never be fully specified and are subject to constant adjustments and adaptation). To accommodate continuous evolution, we need a fundamental change of perspective: from satisfaction of requirements through traditional, top-down engineering to satisfaction of requirements by regulation of complex, decentralized systems.

One way to accomplish "flexible" satisfaction of requirements is to introduce feedback loops for each trade-off. To be able to observe and possibly orchestrate the continuous evolution of software systems in a complex changing environment, we need to push the monitoring of properties of evolving systems to unprecedented levels. In particular, to instrument software systems with autonomic elements using software reverse engineering and migration technology to enhance their monitoring and assessment capabilities. Finally, to get proficient in unprecedented monitoring and feedback loops in software, we need to teach the notion of a control loop in 1st Year Programming Courses (i.e., not only pure objects, but also objects with embedded control loops or autonomic elements).

Keywords: Adaptive systems, software ecosystems, autonomic systems, monitoring, diagnostics

Full Paper:
http://csdl.computer.org/dl/proceedings/wcre/2006/2719/00/27190009.pdf

Supporting change in long-lived software systems

Oscar Nierstrasz (Universität Bern, CH)

Real software must be capable of adapting itself to a changing context, both in the long term and in the short term. Unfortunately current programming languages, runtime systems and development environments are too static — they attempt to control change and limit its impact rather than enable it. We argue instead that, in order to enable change and adaptation, software systems should (i) support a first-class notion of change, (ii) be fully reflective, i.e., rather than being model-driven, they should have access to a causally-connected model of they own structure and behaviour, (iii) monitor and analyze their own behaviour at run-time, and (iv) feed this information back to the developer in the IDE. We will illustrate these ideas with live demos of research prototypes under development.

Professor

Sooyong Park (Sogang University - Seoul, ROK)

On-line evaluation and planning is one important issues in self-managed software research. In my talk, I would like to discuss on goal oriented architecture evaluation and learning based planning approach. Some self-managed software applications in Robot and mobile phone are also presented. Additionally, I would like to comments on some issues in self-managed software and software product line approach.

Keywords: On-line evaluation, planning, SPL

Healing Functional Problems: From Failure Detection to Fault Healing

Mauro Pezzé (University of Lugano, CH)

Dynamic integration and evolution of software systems can result in integration failures that depend on functional mismatches difficult to detect and remove at design time. We argue that many integration failures that depend on functional problems can be automatically detected and eliminated at run time. We discuss a set of techniques for detecting functional failures, and for diagnosing, locating and healing the corresponding faults. We will focus on the different steps with particular emphasis on fault healing.

Keywords: Functional integration faults, failure detection, fault diagnosis, fault healing

Development and Execution of Adaptive Component-based Applications

Andreas Rasche (Hasso-Plattner-Institut - Potsdam, D)

The talk introduces the Adapt.Net project, which provides an execution environment for adaptive applications on top of component-platforms such as the Java- and .NET-platform. The talk concentrates on a new algorithm for dynamic reconfiguration supporting multi-threaded applications with cyclic call dependencies and the dynamic update of component-based applications. In addition the talk focuses on tool-support for adaptive applications including the usage of aspect-oriented programming (AOP) for generating (re-)configuration-specific logic. Finally a case study on adaptive control applications is presented, where we use the execution platform is used to protect experiment-hardware in a virtual/remote laboratory environment from malicious code submitted via the Internet.

Keywords: Dynamic reconfiguration components dynamic updates AOP

A Brief Introduction to Organic Computing

Hartmut Schmeck (Institute of Technology - KIT, D)

The mission of Organic Computing is to tame complexity in technical systems by providing appropriate degrees of freedom for self-organized behavior adapting to changing requirements of the execution environment, in particular with respect to human needs. According to this vision an organic computer system should be aware of its own capabilities, the requirements of the environment, and it should be equipped with a number of "self-x" properties allowing for the anticipated adaptiveness and for a reduction in the complexity of system management. These self-x properties are e.g. self-organisation, self-configuration, self-optimization, self-healing, self-protection and self-explanation.

To achieve these ambitious goals the German Research Foundation is funding a priority research program on organic computing for a six year period (2005 - 2011) to develop new methods, techniques, and system architectures. The talk gives a brief overview of this priority program (more details at www.organic-computing.de/SPP).

Keywords: Organic computing, robustness, adaptivity, self-organisation, trustworthiness

An Approach to Characterizing Self-organisation, Adaptivity, Robustness,...

Hartmut Schmeck (Institute of Technology - KIT, D)

This talk presents a characterization of (controlled) self-organisation and adaptivity that is motivated by the main objectives of the Organic Computing Initiative. We propose a systematic classification of robust, flexible, adaptable, and adaptive systems and define a degree of autonomy to be able to quantify how autonomously a system is working. The degree of autonomy distinguishes and measures external control which is exhibited directly by the user (no autonomy) from internal control of a system which might be fully controlled by an observer/controller architecture that is part of the system (full autonomy). The main purpose of this classification is to provide a common basis for a comparative evaluation of adaptive systems.

Keywords: Adaptivity, robustness, degree of autonomy, controlled self-organisation

Joint work of: Müller-Schloer, Christian; Schmeck, Hartmut

The Generic Observer/Controller Architecture of Organic Computing as a Template for Software Architectures for (Self-)Adaptive Systems

Hartmut Schmeck (Institute of Technology - KIT, D)

The Observer/Controller architecture has evolved as a generic template for the design of organic computing systems. Its components and structure may serve in an obvious way as a template for software architectures for self-adaptable systems. Particular features of this architecture are an observer structure (containing components for monitoring, analysis, prediction and aggregation) that can be modified by selecting an appropriate model of observation and the presence of on-line and off-line learning mechanisms in the controller to adapt the information on the feasibility of available actions and to generate new actions for previously unanticipated situations in the system under observation and control. The O/C architecture may be mapped in a straightforward way to the reference architecture presented by Jeff Magee.

Keywords: Observer/controller architecture, monitoring, adaptive control

Trustworthiness of (Self-)Adaptive Systems

Hartmut Schmeck (Institute of Technology - KIT, D)

The usability of adaptive systems or services significantly depends on the trust relationship between the user and the service provider. This brief statement addresses the need for trust engineering to provide initial trust and to support a reasonable trust management (e.g. build up higher levels of trust and regain trust after failures). Software Engineering is challenged to go far beyond current credential- and reputation-based approaches and to address the certification of adaptive software components as a basis for trust engineering.

Keywords: Trust, safety, security, trust engineering

Reactive, Anticipatory, and Implicit Adaptation

Mary Shaw (CMU - Pittsburgh, USA)

This talk comments on three general styles of adaptation: reactive (the most common), anticipatory (incorporating prediction of future state), and implicit (requiring no explicit representation of current state)

Keywords: Adaptive software, reactive adaptation, anticipatory adaptation, implicit adaptation, control loops

Architectures for Adaptation Should have Control Loops

Mary Shaw (CMU - Pittsburgh, USA)

An adaptive system will usually have a feedback loop. That should be clearly visible in the architecture

Keywords: Adaptive architecture, feedback loop

Hazard Analysis of Self-Adaptive Systems

Matthias Tichy (Universität Paderborn, D)

I will give a short (15 min) talk about our latest research on hazard analysis of self-adaptive systems. This talk relates to topic 4 - safety evaluation.

Full Paper:
http://www.springerlink.com/content/j10903k426238622/

Multi-agent systems and software engineering for self-adaptive systems

Danny Weyns (Katholieke Universiteit Leuven, B)

I would like to give a talk on the relation of multi-agent systems and software engineering for self-adaptive systems. This would fit in the slot of cross-pollination (topic 8) or high-level design (topic 2).

Estimated time: 30 minutes

Self-Adaptability in the Presence of Failures

Rogerio de Lemos (University of Kent, GB)

The level of self-adaptability, which one extreme is full autonomy, in any system is essentially constrained by the intent of the system and the resources available to achieve that intent, thus the reasoning about failures in system resources should be an integral part in system design. In other words, the provision of autonomy should rely on how the system is built in terms of its hardware and software components, and how these can be exploited for the continue delivery of system services despite the presence of failures. For that, system architecture is a key factor that influences the flexibility of a system adapting to change and handling faults. For instance, how the system is structured is fundamental for establishing the loci of change, in providing the appropriate modularity for preventing the introduction of faults and their subsequent removal, and in restricting the propagation of errors, which facilitates the processes of fault diagnosis and treatment.

Moreover, it is at the architectural level that tradeoffs are solved between what is required from the system, and what the system resources are actually able to offer.

Keywords: Autonomy, fault tolerance, architectures, simulation