Karlsruhe Institute of Technology

Software Design and Quality

FZI

CBSE QoSA CBHPC WCOP SI-Day

COMP ARCH 2008

October 14th - 17th, 2008
Karlsruhe (Germany)

Conference Companion

Program Overview

Time	Monday, 13.10.	TUESDAY, 14.10		WEDNESDAY, 15.10.	THURSDAY, 16.10.			FRIDAY, 17.10.		
	WCOP	CBSE & QoSA		SID	CBSE & QoSA		CBHPC	CBSE & QoSA		CBHPC
08:00	Registration	Registration		Registration						
09:00	Welcome			Keynote Michael Stal	Keynote Carlo Ghezzi		Keynote André Ribes	Keynote Florian Matthes		
	Technical Sessions									
10:00		Welcome		Break						
				Industrial Experience Report Track	Formal Methods	Architectural Design Decisions	Component Models and Integrated Environments	Performance 1	Tools	Portability and Parallel Performance
10:30		Break								
11:00		Keynote Thomas Dreier								
12:00	Lunch									
13:30	Breakout Groups	Analysis of Extra-functional Properties 1	New Component Models	Tutorials	Validation & Verification	Methods of Development	Large-scale Simulations	Keynote Philippe Kruchten		Software Engineering Techniques
							Discussion			
								Break		
15:00		Break						Performance 2	Empirical Studies	Break
15:30		Analysis of Extra-functional Properties 2	Middleware for Component-Based Systems	Tutorials cont'd	Social Event *Steam engine trip to the heart of the black forest*					Discussion & Closing
16:00	Presentation and Discussion									
								Break		
17:00	End of WCOP	Break						Closing Assembly		
17:30		*Reception at Schloss Karlsruhe*		Panel						
19:00	Dinner at Vogelbräu			*Software Industrial Day Reception*						

Table of Contents

I Information about CompArch

II Industrial Experience Reports

Part I

Information about CompArch

Welcome Note

Dear participants,

Welcome to CompArch 2008, the federated conference series bringing together researchers and practitioners in the fields component-based software engineering and software architecture, from both academia and industry. CompArch fosters a vibrant community of researchers and practitioners by providing a joint environment for the conferences CBSE and QoSA as well as the workshops CBHPC and WCOP. It is an honour to host this exceptional event for the first time in Germany.

As a tradition for CompArch, in the middle of the week, the industrial day (this time firstly called more pronouncedly the Software Industrial Day, SID) provides an attractor for the mutually fruitful exchange between software practitioners and researchers. This fits well with the uniquely close and fruitful cooperation between the Karlsruhe software research institutions–besides the Informatics faculty at the University, the University of Applied Sciences, the National Research Laboratory Karlsruhe (Forschungszentrum Karlsruhe), the IT Research Centre Karlsruhe (FZI) and the Fraunhofer ITG institute–and an exceptionally strong and innovative IT business with the over 30,000 IT workers in the region.

We are proud that the CompArch 2008 programme forms a distinguished combination of sessions ranging over many component-related areas, touching both research and industry. Our keynote speakers include Dr. Carlo Ghezzi, Professor of Software Engineering at Politecnico di Milano, and holder of the ACM SIGSOFT Distinguished Service Award 2006; Dr. Michael Stal, Head of the Distributed Systems–Middleware, Architecture, Integration group at Siemens AG; Dr. Philippe Kruchten, Professor of Software Engineering at University of British Columbia in Vancouver, and the lead architect of the Rational Unified Process; Professor Dr. Thomas Dreier, Head of the Institute for Information Law at Universität Karlsruhe (TH); Professor Dr. Florian Matthes, Head of the Chair for Software Engineering for Business Information Systems at the Techniche Universität München; and Dr. André Ribes, Researcher at EDF, France.

We hope that this event will advance the debate about important trends and future directions of component-based software engineering and software architectures, and help all participants to get in touch and establish contact with leading experts in the field. Finally, we would like to thank you for participating in CompArch 2008 and making it all possible through your attendance. For the same reason, we thank our supporters and partners for their generous assistance, and all researchers and student volunteers of the Chair of Software Design and Quality at the Faculty for Informatics at Universität Karlsruhe (TH) who contributed to the organisation of CompArch 2008.

Sincerely,

Ralf Reussner
CompArch 2008 General Chair

S. Becker, M. Chaudron, C. Perez, F. Plasil, M. Sosonkina, C. Szyperski, W. Weck, and J.-P. Weiss
For the Chairs of CompArch 2008 Events

Conference Facilities and Related Information

Conference Location

All conferences and events of CompArch 2008 are situated in the university campus in the buildings 11.40 and 10.91. A map of the location together with a detailed room plan can be found at the end of this booklet. CBSE and QoSA have joint technical sessions, two of them being in parallel in the lecture halls “Redtenbacher” and “Grashof” in building 10.91. CBHPC technical sessions are situated in the lecture hall “Mittlerer Hörsaal” in building 10.91. The rooms and lecture halls of particular sessions and events can be found in the Conference Program section of this booklet.

Please wear your badge at all time during the conference, in particular also for lunches, receptions, and the trip to the Black Forest.

Registration & Information Desk

The registration & information desk is located in Tulla Foyer in building 11.40 Tuesday to Thursday and in Redtenbacher Foyer in building 10.91 on Friday. The desk opens at 8:30 on Tuesday and at 8:00 on the following days, and is open throughout the day. Here, you can always find somebody to help you with any kind of questions.

Luggage Room

We provide a luggage room in which you can deposit your luggage. Please note that we accept no responsibility for items left in the luggage room. For the luggage room, please follow the signs starting at the information desk (see above).

Lunch and Coffee Breaks

Lunch is served between 12:00 and 13:30. Tuesday to Thursday, a lunch buffet is served in Tulla Foyer in building 11.40, just where the registration & information desk is located. On Friday, a finger-food buffet is served in the “Gastdozentenhaus” (building 01.52, see the location map at the end of this booklet, the top-right corner). All luncheons are included in the registration fee.

Coffee and snacks are provided during the small breaks in Tulla Foyer in building 11.40 (Tuesday to Thursday) and Redtenbacher Foyer in building 10.91 (Friday only). The foyers are also indicated in the location map.

At the end of the first part of this booklet, an overview of good restaurants close to the conference location is provided by means of a restaurant guide *Towards Food Supply in Karlsruhe*.

Social Events

Three social events are organized within CompArch 2008, two receptions and one trip. Detailed information about the events can be found in the Social Events section of this booklet.

Internet Access

The university campus is covered by a wireless local area networks. During the conference week, you can access the network with the login credentials you find in your conference bag. Further information about Internet settings can be found below. A limited number of desktop PCs with Internet access is also going to be available in Tulla Foyer in building 11.40 from Tuesday to Thursday.

WLAN Access During the CompArch 2008, access to the Internet using WLAN is offered to all CompArch participants. The individual user names and the passwords are distributed together with conference proceedings at the registration desk. Please see http://comparch2008.ipd.uka.de for further information.

If you have any questions, or if your user name/password are missing, please do not hesitate to ask Michael Kuperberg (mkuper@ipd.uka.de) or any other member of the conference staff.

How to use the WLAN from a notebook, in five easy steps:

1. Activate your WLAN antenna and search for available WLAN networks. The following networks are available:
   - `dukath-mb` (Redtenbacher, Mittlerer Hörsaal and Grashof in building 10.91)
   - `dukath-aw` (Tulla and foyer in building 11.40)
2. Go to https://dukath-www.rz.uni-karlsruhe.de/index_en.html
3. Enter the user name and the password distributed to you with this companion, then click "login". Once you are logged in, the browser tab/window with the website that is shown after you logged in must remain open.

You must re-login after restarting your notebook, or after resuming from hibernate/standby modus.

For detailed and OS-specific instructions both for notebooks and mobile devices, please visit the CompArch 2008 website `http://comparch2008.ipd.uka.de`.

Emergency Situations

In case of emergency situation, please contact any of the organizers around (white and blue CompArch T-shirts), or the persons at the registration & information desk. If something happens outside conference location, call 110 for police, and 112 for ambulance and/or the fire fighters. The closest pharmacy can be found near Kronenplatz (Schloss Apotheke, Kronenstraße 24), as well as the closest ATM (Postbank, Kaiserstraße 14).

Public Transport

Trams depart from Kronenplatz and Durlacher Tor (see campus map at the end of this companion). For travelling inside Karlsruhe, you need a 2 Zone ticket (it includes the train station, university, and all hotels proposed by the CompArch organisers). Daily tickets (“City-Karte”) cost 4.50 Euro, one-way tickets 2.00 Euro.

With your registration to CompArch, you received **4 daily tickets** for Karlsruhe. We kindly ask you to return the tickets which you did not need to the organisers at the end of the conference. Thanks a lot!

List of Participants

A list of the CompArch 2008 participants who agreed, with their email addresses, will be available after the conference. Please request the list from Anne Martens, email `<martens@ipd.uka.de>`, until December 31st, 2008.

Conference Organisation

CompArch is a federated conference series, this year consisting of five events: conferences CBSE and QoSA, workshops CBHPC and WCOP, and the Software Industrial Day (SID).

CBSE: 11th International Symposium on Component Based Software Engineering

Component-based Software Engineering (CBSE) has emerged as a technology for rapid assembly of flexible software systems. CBSE combines elements of software architecture, modular software design, software verification, configuration and deployment. To foster exchange and collaboration with the software architecture community, CBSE is co-located with the conference on Quality of Software Architectures (QoSA) as part of the federated CompArch events. The CBSE symposium has a track record of bringing together researchers and practitioners from a variety of disciplines to promote a better understanding of CBSE from a diversity of perspectives, and to engage in active discussion and debate. CBSE 2008 is open to all participants interested in CBSE and related areas. The symposium addresses participants from both universities and industry.

Program Chairs
Michel Chaudron, TU Eindhoven & Leiden University, The Netherlands
Clemens Szyperski, Microsoft, USA

Steering Committee
Ivica Crnkovic, Maelardalen University, Sweden
Ian Gorton, Pacific North West National Laboratory, USA
George Heineman, Worcester Polytechnic Institute, USA
Heinz Schmidt, RMIT University, Australia
Judith Stafford, Tufts University, USA
Clemens Szyperski, Microsoft, USA

Program Committee
Uwe Assmann, Dresden University of Technology, Dresden, Germany
Mike Barnett, Microsoft Research, USA
Antonia Bertolino, CNR Research, Pisa, Italy
Judith Bishop, University of Pretoria, Pretoria, South Africa
Ivica Crnkovic, Malardalen University, Vasteras, Sweden
Dimitra Giannakopoulou, RIACS/NASA Ames, Moffet Field CA, USA
Ian Gorton, Pacific North West National Laboratory, Richland WA, USA
Lars Grunske, University of Queensland, Brisbane, Australia
Richard Hall, LSR-IMAG, Grenoble, France
Dick Hamlet, Portland State University, Portland OR, USA
George Heineman, Worcester Polytechnic Institute, Worcester MA, USA

Jean-Marc Jezequel, IRISA (INRIA & Univ. Rennes 1), Rennes, France
Bengt Jonsson, Uppsala University, Uppsala, Sweden
Joe Kiniry, University College Dublin, Ireland
Gerald Kotonya, Lancaster University, Lancaster, UK
Magnus Larsson, ABB Corporate Research, Vasteras, Sweden
Kung-Kiu Lau, University of Manchester, Manchester, UK
Raphael Marvie, University of Lille, Lille, France
Michael Maximilien, IBM Almaden Research Centre, San Jose CA, USA
Nenad Medvidovic, University of Southern California, Los Angelos CA, USA
Henry Muccini, University of L'Aquila, L'Aquila, Italy
Rob van Ommering, Philips Research Labs, Eindhoven, The Netherlands
Ralf Reussner, University Karlsruhe, Karlsruhe, Germany
Alessandra Russo, Imperial College, London, UK
Christian Salzmann, BMW Car IT, Munich, Germany
Douglas Schmidt, Vanderbilt University, Nashville TN, USA
Heinz Schmidt, RMIT University, Australia
Jean-Guy Schneider, Swinburne University of Technology, Melbourne, Australia
Judith Stafford, Tufts University, USA
Asuman Sünbül, SAP Research, Palo Alto CA, USA
Clemens Szyperski, Microsoft, USA
Massimo Tivoli, University of L'Aquila, L'Aquila, Italy
Wolfgang Weck, Independent Software Architect, Zürich, Switzerland
Dave Wile, Teknowledge Corp., Los Angelos CA, USA

QoSA: 4th International Conference on the Quality of Software Architectures

Today, a system's software architecture cannot be seen simply as a means to an end, the end being the implemented system. Although the ultimate measure of the quality of the software architecture lies in the implemented system, in how well it satisfies the requirements and constraints of the project and whether it can be maintained and evolved successfully, the quality of a system's software architecture is one of the critical factors in its overall system quality - encompassing both functional and extrafunctional properties. In order to treat design as an engineering discipline rather than an art, we need the ability to address the quality of the software architecture directly, not simply as it is reflected in the implemented system.

This is a specific goal of QoSA - to deal with software architecture in general and simultaneously focus on its quality characteristics by addressing the problems of:

- designing software architectures of good quality,
- defining, measuring, evaluating architecture quality, and
- managing architecture quality, tying it upstream to requirements and downstream to implementation, and preserving architecture quality throughout the lifetime of the system.

Cross-cutting these problems is the question of the nature of software architecture. Software architecture organizes a system, partitioning it into elements and defining relationships among the elements. For this we often use multiple views, each with a different organizing principle.

But software architecture must also support properties that are emergent and cannot be ascribed to particular elements. For this we often use the language of quality attributes. Quality attributes cover both internal properties, exhibited only in the development process (e.g. maintainability, portability, testability, etc.), and external properties, exhibited in the executing system (e.g. performance, resource consumption, availability, etc.). Quality attributes cover properties that are emergent, that have a pervasive impact, that are difficult to reverse, and that interact, thereby precluding or constraining other properties. Thus, QoSA also aims to investigate quality attributes in the context of the problems of the design, evaluation, and management of software architecture.

This year's QoSA main topic is on "Models and Architectures". Modelling software architectures for documentation purposes as well as manual analysis is an established practice. Due to the continuous maturation of model-driven software development methods and tools, software architecture models also become subject to automated model transformations. Their target is either to generate high quality software implementations or to automatically derive analysis models for predicting architectural quality characteristics like performance or reliability.

Program Committee Chairs
Frantisek Plasil, Charles University, CZ
Steffen Becker, University of Karlsruhe / FZI, GER

Steering Committee
Ivica Crnkovic, Mälardalen University, SWE
Ian Gorton, Pacific North West National Laboratory, USA
Sven Overhage, University of Augsburg, GER
Judith Stafford, Tufts University, USA
Clemens Szyperski, Microsoft, USA

Program Committee
Colin Atkinson, University of Mannheim, GER
Achim Baier, itemis AG, GER
Len Bass, Software Engineering Institute, USA
Jan Bosch, Intuit, USA
Jeremy Bradley, Imperial College London, UK
Vincenzo Grassi, Universität Rom "Tor Vergata", IT
Wilhelm Hasselbring, University of Oldenburg / OFFIS, GER
Christine Hofmeister, Lehigh University, USA
Jean-Marc Jezequel, University of Rennes / INRIA, FR
Samuel Kounev, University of Cambridge, UK
Patricia Lago, Vrije Universiteit, NL
Nicole Levy, University of Versailles, FR
Markus Lumpe, Swinburne University, AUS
Eric Madelaine, Inria, FR

Tomi Mannisto, Helsinki University of Technology, FIN
Nenad Medvidovic, University of Southern California, USA
Raffaela Mirandola, Politecnico di Milano, IT
Robert Nord, Software Engineering Institute, USA
Dorina Petriu, Carleton University, CAN
Iman Poernomo, King's College, UK
Sasikumar Punnekkat, Mälardalen University, SWE
Andreas Rausch, Clausthal University of Technology, GER
Matthias Riebisch, Technical University of Ilmenau, GER
Roshanak Roshandel, Seattle University, USA
Bernhard Rumpe, University of Technology Braunschweig, GER
Jean-Guy Schneider, Swinburne University, AUS
Michael Stal, Siemens, GER
Petr Tuma, Charles University, CZ
Axel Uhl, SAP AG, GER
Kurt Wallnau, Software Engineering Institute, USA
Wolfgang Weck, Independent Software Architect, CH
Murray Woodside, Carlton University, CAN
Steffen Zschaler, Technical University of Dresden, GER

CBHPC: International Workshop on Component-Based High Performance Computing

CBHPC 2008 is the third joint event of the HPC-GECO and CompFrame workshop series. This workshop focuses on the role of component and framework technologies in high-performance and scientific computing, and on high-level, component-based and innovative programming tools and environments to efficiently develop high performance applications and exploit them both on individual massively parallel systems and on the Grid.

General Co-Chairs
Christian Perez, INRIA, France
Masha Sosonkina, Ames Laboratory and Iowa State University, USA

Local Chair
Jan-Philipp Weiss, University of Karlsruhe, Germany

Steering Committee
Rob Armstrong, Sandia National Laboratories, USA
David E. Bernholdt, Oak Ridge National Laboratory, USA
Massimo Coppola, Institute of Information Science and Technologies, CNR, Italy
Marco Danelutto, Universita di Pisa, Italy
Vladimir S. Getov, University of Westminster/CoreGRID, UK
Aad van der Steen, Utrecht University, The Netherlands

Program Committee
Rob Armstrong, Sandia National Laboratories, USA

Mikio Aoyama, Nanzan University, Japan
Rosa Badia, Universidad Politéica de Cataluña, Spain
Purushotham Bangalore, University of Alabama at Birmingham, USA
Françoise Baude, University of Nice-Sophia Antipolis, France
David E. Bernholdt, Oak Ridge National Laboratory, USA
Francisco de Carvalho Junior, Universidade Federal do Ceará Brazil
Massimo Coppola, Institute of Information Science and Technologies, CNR, Italy
Marco Danelutto, Universita di Pisa, Italy
Doug Dechow, Tech-X Corporation, USA
Wael Elwasif, Oak Ridge National Laboratory, USA
Vladimir S. Getov, University of Westminster, UK
Madhu Govindaraju, Binghamton University, USA
James Kohl, Oak Ridge National Laboratory, USA
Boyana Norris, Argonne National Laboratory, USA
Christian Perez, INRIA, France
Thierry Priol, INRIA, France
Rainer Schmidt, Austrian Research Centers, Austria
Masha Sosonkina, Ames Laboratory and Iowa State University, USA
Aad van der Steen, HPC Research, The Netherlands
James-Bernard Stefani, INRIA, France
Rainer Stotzka, Forschungszentrum Karlsruhe, Germany
Alan Sussman, University of Maryland, USA
Nanbor Wang, Tech-X Corporation, USA

WCOP: 13th International Workshop on Component-Oriented Programming – Components in a World of Mobile and Distributed computing

WCOP seeks position papers on the important field of component-oriented programming (COP). WCOP 2008 is the thirteenth event in a series of highly successful workshops, which took place in conjunction with every ECOOP since 1996. This year WCOP is part of the CompArch federated event.

COP has been described as the natural extension of object-oriented programming to the realm of independently extensible systems. Several important approaches have emerged over the recent years, including component technology standards, such as CORBA/CCM, COM/COM+, J2EE/EJB, .NET, and most recently software services, but also the increasing appreciation of software architecture for component-based systems, as in SOA, and the consequent effects on organizational processes and structures as well as the software development business as a whole.

COP aims at producing software components for a component market and for late composition. Composers are third parties, possibly the end users, who are not able or not willing to change components. This requires standards to allow independently created components to interoperate, and specifications that put the composer into the position to decide what can be composed under which conditions. On these grounds, WCOP'96 led to the following definition:

A component is a unit of composition with contractually specified interfaces and explicit context dependencies only. Components can be deployed independently and are subject to composition by third parties.

Where WCOP'96 focused on the fundamental terminology of COP, the subsequent workshops expanded attention to the many related facets of component software.

WCOP 2008 will discuss components in the context of mobile and distributed computing. How can components be deployed effectively in distributed environments? For instance, distribution and mobile computing always imply that individual operations may fail and that services may be temporarily unavailable in unpredictable ways due to communication problems, such as network failures. How does it affect the idea of contract-based trust, if contract signers may not be reachable?

In particular on mobile platforms resources are limited and balancing resource requirements and allocations among a potentially open set of (installable) add-on components becomes a particular challenge.

Finally, in addition to submissions addressing the theme, we explicitly solicit papers reporting on experience with component-oriented software systems in practice, where the emphasis is on interesting lessons learned, whether the actual project was a success or a failure.

Workshop Co-Organizers
Ralf Reussner, Universität Karlsruhe, Germany
Clemens Szyperski, Microsoft, USA
Wolfgang Weck, Independent Software Architect, Switzerland

SID: Software Industrial Day

The Software Industrial Day / Software-Industrie-Tag aims at a tight integration of practitioners at CompArch 2008. The Software Industrial Day includes an invited talk of Dr. Michael Stal on Dealing with Design Erosion–Architecture Refactoring; several experience reports from different companies; as well as three tutorials on Systematically Designing Component Frameworks, Model-based Software Performance Prediction, and Principles of Service-Oriented Architectures; and a panel discussion on Extra-functional Contracts Versus Service Level Agreements.

Event Chair
Ralf Reussner, Universität Karlsruhe & FZI Forschungszentrum Informatik, GER

Program Committee
Ivica Crnkovic, Real Time Research Centre, Maelardalen, SWE
Ian Gorton, Pacific North West National Laboratory, USA
George Heineman, Worcester Polytechnic Institute, USA
Sven Overhage, Oversoft, GER
Heinz Schmidt, RMIT University, AUS
Judith Stafford, Tufts University, USA
Clemens Szyperski, Microsoft, USA

Program

Time	Monday, 13.10.2008 WCOP	TUESDAY, 14.10.2008 CBSE & QoSA		WEDNESDAY, 15.10.2008 CBSE & QoSA, Software Industrial Day		
08:00						
08:30	Registration [50.34; 333]	Registration *[Tulla Foyer]*		Registration *[Tulla Foyer]*		
09:00	Welcome Technical Sessions (with breaks) *[50.34; 348]*			Keynote Michael Stal Siemens AG Deadling with Design Erosion - Architecture Refactoring *[Tulla]*		
10:00		Welcome *[Redtenbacher]*		Break		
10:30		Break		Industrial Experience Report Track *[Tulla]*		
11:00		Keynote Thomas Dreier Universität Karlsruhe *Components, Architecture and the Law* *[Redtenbacher]*				
12:00	Lunch	Lunch		Lunch		
13:30	Breakout Groups (with breaks) *[50.34; 348]*	Technical Sessions 1a Analysis of Extra-functional Properties 1 *[Redtenbacher]*	1b New Component Models *[Grashof]*	Tutorial 1 Wolfgang Weck *Systematically Designing Component Frameworks* *[TBA]*	Tutorial 2 Steffen Becker *Model-based Software Performance Prediction* *[TBA]*	Tutorial 3 Gregor Engels *Principles of Service-Oriented Architectures* *[TBA]*
15:00		Break		Break		
15:30		Technical Sessions 2a Analysis of Extra-functional Properties 2 *[Redtenbacher]*	2b Middleware for Component-Based Systems *[Grashof]*	Tutorial 1 cont'd *[TBA]*	Tutorial 2 cont'd *[TBA]*	Tutorial 3 cont'd *[TBA]*
16:00	Presentation and Discussion *[50.34; 348]*					
17:00	End of WCOP	Break		Break		
17:30		*Reception at "Schloss Karlsruhe"*		Panel: Components and Services - Extra-functional Contracts Versus Service Level Agreements organised by Dr. Wolfgang Theilmann (SAP AG), Mircea Trifu (FZI) *[Tulla]*		
18:30	Workshop Dinner at Vogelbräu *[Kapellenstr. 50, see map]*			*Software Industrial Day Reception* *[Tulla Foyer]*		
19:00						

Overview

THURSDAY, 16.10.2008		FRIDAY, 17.10.2008		Time
CBSE & QoSA	CBHPC	CBSE & QoSA	CBHPC	
Registration *[Tulla Foyer]*		Registration *[Redtenbacher Foyer]*		08:00 08:45
Keynote Carlo Ghezzi Politecnico Milano *Rethinking the Use of Models in SA* *[Redtenbacher]*	Keynote André Ribes EDF France *Salome platform: focus on the component model* *[Mittlerer Hörsaal]*	Keynote Florian Matthes TU Munich *Visualizing and Managing the Evolution of Socio-Technical Systems of Systems* *[Redtenbacher]*	Technical Session P2 Parallel Applications *[Mittlerer Hörsaal]*	09:00
Break		Break		10:00
Technical Sessions 3a Formal Methods *[Redtenbacher]* 3b Architectural Design Decisions *[Grashof]*	Technical Session P1 Component Models and Integrated Environments *[Mittlerer Hörsaal]*	Technical Sessions 5a Performance 1 *[Redtenbacher]* 5b Tools *[Grashof]*	Technical Session P3 Portability and Parallel Performance *[Mittlerer Hörsaal]*	10:30 11:00
Lunch		Lunch		12:00
Technical Sessions 4a Validation & Verification *[Redtenbacher]* 4b Methods of Development *[Grashof]*	Technical Session A1 Large-scale Simulations *[Mittlerer Hörsaal]*	Keynote Philippe Kruchten University of British Columbia *Architecture and Agility - An Oxymoron?* *[Redtenbacher]*	Technical Session A2 Software Engineering Techniques *[Mittlerer Hörsaal]*	13:30
	Discussion of Sessions P1 and A1 *[Mittlerer Hörsaal]*			
		Break		14:30
Break		Technical Sessions 6a Performance 2 *[Redtenbacher]* 6b Empirical Studies *[Grashof]*	Break Discussion of Session P2, P3 and A2. Closing Disc. *[Mittlerer Hörsaal]*	15:00
Social Event *Steam engine trip to the heart of the black forest* *[for bus departure location see campus map]*		Break		
		Closing Assembly *[Redtenbacher]*		17:00 17:30
				19:00

Conference Program WCOP

Monday, 13.10.2008

08:30 **Registration** *[Building 50.34, Level 3, Room 333]*

09:00 **Welcome** *[Building 50.34, Room 348]*

09:15 **Technical Session 1** *[Building 50.34, Room 348]*

Plug-ins: An Architectural Style for Component Software
Jörg Rathlev

Software Extension Mechanisms
Benjamin Klatt, Klaus Krogmann

Closing the Gap between Business Functions and Software Components in Distributed Enterprise Systems
David Helton

10:15 **Break**

10:40 **Technical Session 2** *[Building 50.34, Room 348]*

Performance-oriented Design Space Exploration
Anne Martens, Heiko Koziolek

Treaty - A Modular Component Contract Language
Jens Dietrich, Graham Jenson

A Component-Based Methodology to Increase the Adoption of Standards
Enric Jaen Villoldo, Emilio Luque-Fadon, Joan Serrat-Fernandez

A Component Design Process based on Feature Model Transformations
Matthias Riebisch, Periklis Sochos

12:00 **Lunch**

13:30 **Breakout Groups** *[Building 50.34, Room 348]*

14:30 **Break**

15:00 **Breakout Groups (cont'd)** *[Building 50.34, Room 348]*

16:00 **Presentation and Discussion** *[Building 50.34, Room 348]*

17:00 **End of WCOP**

18:30 **Workshop Dinner at Vogelbräu** *[Kapellenstr. 50, see map]*

Conference Program CBSE and QoSA

Tuesday, 14.10.2008

08:30 **Registration** *[Tulla Foyer, Building 11.40]*

10:00 **Welcome** *[Redtenbacher, Building 10.91]*
Greeting of Prof. Dr. Norbert Henze, the Vice President of the University Karlsruhe (TH)

10:30 **Break** *[Tulla Foyer, Building 11.40]*

11:00 **Keynote** *[Redtenbacher, Building 10.91]*
Components, Architecture and the Law - A Tricky Interplay
Thomas Dreier
Universität Karlsruhe (TH), Germany
Chair: Judith Stafford

12:00 **Lunch** *[Tulla Foyer, Building 11.40]*

13:30 **Technical Session 1a** *[Redtenbacher, Build. 10.91]*
Analysis of Extra-functional Properties 1
Chair: Heinz Schmidt

Validating Access Control Configurations in J2EE Applications
Lianshan Sun, Gang Huang, Hong Mei

Classification of Component Vulnerabilities in Java Service Oriented Programming (SOP) Platforms
Pierre Parrend, Stephane Frenot

Middleware Architecture Evaluation for Dependable Self-managing Systems
Yan Liu, Muhammad Ali Babar, Ian Gorton

Technical Session 1b *[Grashof, Build. 10.91]*
New Component Models
Chair: Heiko Koziolek

A Component Model for Control-Intensive Distributed Embedded Systems
Séverine Sentilles, Aneta Vulgarakis, Tomas Bures, Jan Carlson, Ivica Crnkovic

The CoSi Component Model: Reviving the Black-Box Nature of Components
Premysl Brada

Ada-CCM: Component-based Technology for Distributed Real-Time Systems
Patricia López Martínez, José M. Drake, Pablo Pacheco,

15:00 **Break** *[Tulla Foyer, Building 11.40]*

15:30 **Technical Session 2a** *[Redtenbacher, Build. 10.91]*
Analysis of Extra-functional Properties 2
Chair: Christian Bunse

Architectural Specification and Static Analyses of Contractual Application Properties
Guillaume Waignier, Anne-Francoise Le Meur, Laurence Duchien

Style-based Model Transformation for Early Extrafunctional Analysis of Distributed Systems
Julien Mallet, Siegfried Rouvrais

Component-Level Energy Consumption Estimation for Distributed Java-Based Software Systems
Chiyoung Seo, Sam Malek, Nenad Medvidovic

Technical Session 2a *[Grashof, Build. 10.91]*
Middleware for Component-Based Systems
Chair: Ian Gorton

ESCAPE: A Component-based Policy Framework for Sense and React Applications
Giovanni Russello, Leonardo Mostarda, Naranker Dualy

Experiences from Developing a Component Technology Agnostic Adaptation Framework
Eli Gjørven, Frank Eliassen, Romain Rouvoy

A Practical Approach for Finding Stale References in a Dynamic Service Platform
Kiev Gama, Didier Donsez

17:00 **Break** *[Tulla Foyer, Building 11.40]*

17:30 **Reception at "Schloss Karlsruhe"**

Conference Program SID

Wednesday, 15.10.2008

08:00 **Registration** *[Tulla Foyer, Building 11.40]*

09:00 **Keynote** *[Tulla, Building 11.40]*
Dealing with Design Erosion - Architecture Refactoring
Dr. Michael Stal
Siemens AG, Germany
Chair: Ralf Reussner

10:00 **Break** *[Tulla Foyer, Building 11.40]*

10:30 **Industrial Experience Report Track** *[Tulla, Building 11.40]*
Chair: Wolfgang Weck

Index-based Process and Software Quality Control in Agile Development Projects
Nicole Rauch, Eberhard Kuhn, Holger Friedrich
andrena objects ag

Evaluating Failure Propagation in the Application Landscape of a Large Bank
Josef Lankes, Florian Matthes, Tarmo Ploom
TU Munich

Quality Considerations in SAP Architectures
Wolfgang Theilmann, Roger Kilian-Kehr
SAP Research, CEC Karlsruhe

Analyzing the Extensibility Options of Business Software Solutions
Marcus Echter
SAP AG

12:00 **Lunch** *[Tulla Foyer, Building 11.40]*

13:30 **Tutorials** *[To be announced: Mittlerer Hörsaal, Build. 10.91; Seminar Room 202 and Seminar Room 214, Build. 11.40]*

Tutorial 1	Tutorial 2	Tutorial 3
Systematically Designing Component Frameworks	*Model-based Software Performance Prediction*	*Principles of Service-Oriented Architectures*
Dr. Wolfgang Weck	Dr. Steffen Becker	Prof. Dr. Gregor Engels
independent Software-Architect, Zürich, Switzerland	FZI Forschungszentrum Informatik, Germany	University of Paderborn, Germany

17:00 **Break** *[Tulla Foyer, Building 11.40]*

17:30 **Panel** *[Tulla, Building 11.40]*
Components and Services - Extra-functional Contracts Versus Service Level Agreements
Dr. Wolfgang Theilmann (SAP AG, Germany)
Mircea Trifu (FZI Forschungszentrum Informatik, Germany)

19:00 **Software Industrial Day Reception** *[Tulla Foyer, Building 11.40]*

Conference Program CBSE and QoSA

Thursday, 16.10.2008

08:00 **Registration** *[Tulla Foyer, Building 11.40]*

09:00 **Keynote** *[Redtenbacher, Building 10.91]*
Rethinking the Use of Models in Software Architecture
Prof. Dr. Carlo Ghezzi
Politecnico di Milano, Italy
Chair: Ivica Crnkovic

10:00 **Break** *[Tulla Foyer, Building 11.40]*

10:30 **Technical Session 3a** *[Redtenbacher, Building 10.91]*
Formal Methods
Chair: Barbora Zimmerova

Synthesis of Connectors from Scenario-based Interaction Specifications
Meng Sun, Farhad Arbab

State Space Reduction Techniques for Component Interfaces
Markus Lumpe, Lars Grunske, Jean-Guy Schneider

Model Checking of Control-User Component-Based Parametrised Systems
Pavlina Varekova, Ivana Cerna

Technical Session 3b *[Grashof, Building 10.91]*
Architectural Design Decisions
Chair: Jens Happe

Sharing the Architectural Knowledge of Quantitative Analysis
Anton Jansen, Tjaard de Vries, Paris Avgeriou, Martijn van Veelen

Comprehensive Architecture Evaluation and Management in Large Software-Systems
Frank Salger, Marcel Bennicke, Gregor Engels,
Claus Lewerentz

Designing the Enterprise Architecture Function
Max Stahlecker, Bas van der Raadt,
Hans van Vliet

12:00 **Lunch** *[Tulla Foyer, Building 11.40]*

13:30 **Technical Session 4a** *[Redtenbacher, Building 10.91]*
Validation & Verification
Chair: Lars Grunske

Automatic Protocol Conformance Checking of Recursive and Parallel Component-Based Systems
Andreas Both, Wolf Zimmermann

Structural Testing of Component-Based Systems
Daniel Sundmark, Jan Carlson, Sasikumar Punnekkat, Andreas Ermedahl

Towards Component-based Design and Verification of a micro-Controller
Yunja Choi, Christian Bunse

Technical Session 4a *[Grashof, Building 10.91]*
Session 4b: **Methods of Development**
Chair: Carlo Ghezzi

Towards a Systematic Method for Identifying Business Components
Antonia Albani, Sven Overhage, Dominik Birkmeier

Life-Cycle Aware Modelling of Software Components
Heiko Koziolek, Steffen Becker, Jens Happe, Ralf Reussner

A Component Selection Framework for COTS Libraries
Bart George, Régis Fleurquin, Salah Sadou

15:00 **Break** *[Tulla Foyer, Building 11.40]*

15:30 **Social Event** *[for bus departure location please see the campus map at the end of this companion]*

Conference Program CBHPC

Thursday, 16.10.2008

08:00 **Registration** *[Tulla Foyer, Building 11.40]*

09:00 **Keynote** *[Mittlerer Hörsaal, Building 10.91]*
Salome platform: focus on the component model
André Ribes
EDF France

09:45 **Discussion** *[Mittlerer Hörsaal, Building 10.91]*

10:00 **Break** *[Tulla Foyer, Building 11.40]*

10:30 **Technical Session P1: Component Models and Integrated Environments** *[Mittlerer Hörsaal, Building 10.91]*

Methodology for Component-based Development of Grid Applications
Artie Basukoski, Peter Buhler, Vladimir Getov, Stavros Isaiadis, Thomas Weigold

A GCM-Based Runtime Support for Parallel Grid Applications
Elton Mathias, Francoise Baude, Vincent Cave

Towards Software Component Assembly Language Enhanced with Workflows and Skeletons
Marco Aldinucci, Hinde Lilia Bouziane, Marco Danelutto, Christian Perez

12:00 **Lunch** *[Tulla Foyer, Building 11.40]*

13:30 **Technical Session A1: Large-scale simulations** *[Mittlerer Hörsaal, Building 10.91]*

Incorporating Concurrent Component Execution in Loosely Coupled Integrated Fusion Plasma Simulation
Samantha Foley, Wael Elwasif, Aniruddha G. Shet, David E. Bernholdt, Randall Bramley

FACETS - a physics driven parallel component framework
Svetlana Shasharina, John Cary, Ammar Hakim, Gregory Werner, Scott Kruger, Alex Pletzer

Integrating Large-scale Distributed and Parallel HPC Applications using a Component-Based Architecture
Nanbor Wang, Paul Hamill, Fang Liu, Stephen Tramer, Rooparani Pundaleeka, Randall Bramley

14:30 **Discussion of sessions P1 and A1** *[Mittlerer Hörsaal, Building 10.91]*

15:00 **Break** *[Tulla Foyer, Building 11.40]*

15:30 **Social Event** *[for bus departure location please see the campus map at the end of this companion]*

Conference Program CBSE and QoSA

Friday, 17.10.2008

08:00 **Registration** *[Redtenbacher Foyer, Building 10.91]*

09:00 **Keynote** *[Redtenbacher, Building 10.91]*
Visualizing and Managing the Evolution of Socio-Technical Systems of Systems
Prof. Dr. Florian Matthes
TU Munich, Germany
Chair: Frantisek Plasil

10:00 **Break** *[Redtenbacher Foyer, Building 10.91]*

10:30 **Technical Session 5a** *[Redtenbacher, Building 10.91]*
Performance 1
Chair: Sven Overhage

Model-Driven Performance Analysis
Gabriel A. Moreno, Paulo Merson

Performance Prediction for Black-Box Components using Reengineered Parametric Behaviour Models
Michael Kuperberg, Klaus Krogmann, Ralf Reussner

MOSES: MOdeling Software and platform architEcture in UML 2 for Simulation-based performance analysis
Vittorio Cortellessa, Pierluigi Pierini,
Romina Spalazzese, Alessio Vianale

Technical Session 5b *[Grashof, Building 10.91]*
Tools
Chair: Steffen Becker

Carmen : Software Component Model Checker
Ales Plsek, Jiri Adamek

A Tool to Visualize Architectural Design Decisions
Larix Lee, Philippe Kruchten

Integrating Quality-attribute Reasoning Frameworks in the ArchE Design Assistant
Andres Diaz-Pace, Hyunwoo Kim, Len Bass,
Phil Bianco, Felix Bachmann

12:00 **Lunch** *[Gastdozentenhaus, Building 01.52]*

13:30 **Keynote** *[Redtenbacher, Building 10.91]*
Architecture and Agility -- an oxymoron?
Prof. Philippe Kruchten
University of British Columbia, Canada
Chair: Clemens Szyperski

14:30 **Break** *[Redtenbacher Foyer, Building 10.91]*

15:00 **Technical Session 6a** *[Redtenbacher, Building 10.91]*
Performance 2
Chair: Samuel Kounev

Quality Prediction of Service Compositions through Probabilistic Model Checking
Stefano Gallotti, Carlo Ghezzi, Raffaela Mirandola,
Giordano Tamburrelli

Automating Performance Analysis from Taverna Workflows
Rafael Tolosana, Omer Rana, Jose Bañares

Deploying Software Components for Performance
Vibhu Sharma, Pankaj Jalote

Technical Session 6b *[Grashof, Building 10.91]*
Empirical Studies
Chair: Raffaela Mirandola

An Empirical Investigation of the Effort of Creating Reusable Component-Based Models for Performance Prediction
Anne Martens, Steffen Becker, Heiko Koziolek, Ralf Reussner

Scrapheap Challenge: A study of developing systems from scrap components
Gerald Kotonya, Simon Lock, John Mariani

Design Reasoning Improves Software Design Quality
Antony Tang, Minh H. Tran, Jun Han, and Hans van Vliet

16:30 **Break** *[Redtenbacher Foyer, Building 10.91]*

17:00 **Closing Assembly** *[Redtenbacher, Building 10.91]*

Conference Program CBHPC

Friday, 17.10.2008

08:00 **Registration** *[Redtenbacher Foyer, Building 10.91]*

09:00 **Technical Session P2: Parallel Applications** *[Mittlerer Hörsaal, Building 10.91]*

Multi-Physics Coupling of Einstein and Hydrodynamics Evolution: A Case Study of the Einstein Toolkit
Erik Schnetter

Parallelizing Scientific Code with Invasive Interactive Parallelization - a Case Study with Reuseware
Andreas Leha, Mikhail Chalabine, Christoph Kessler

10:00 **Break** *[Redtenbacher Foyer, Building 10.91]*

10:30 **Technical Session P3: Portability and parallel performance** *[Mittlerer Hörsaal, Building 10.91]*

A Component Infrastructure for Performance and Power Modeling of Parallel Scientific Applications
Van Bui, Boyana Norris, Lois Curfman McInnes, Oscar Hernandez, Barbara Chapman, Kevin Huck, Li Li

Hardware-accelerated Components for Hybrid Computing Systems
Daniel Chavarria-Miranda, Jarek Nieplocha, Ian Gorton

Stopping Safely Hierarchical Distributed Components
Marcela Rivera, Henrio Ludovic

12:00 **Lunch** *[Gastdozentenhaus, Building 01.52]*

13:30 **Technical Session A2: Software engineering techniques** *[Mittlerer Hörsaal, Building 10.91]*

Automating SIDL-Based Development for New and Legacy Software
Benjamin Allan, Boyana Norris

Scientific Workflows and Components: Together at Last!
Kostadin Damevski, Ayla Khan, Steven Parker

Composition and Optimization
Morgan Ericsson, Welf Löwe, Christoph Kessler, Jesper Andersson

Hierarchical Usability Levels for Sparse Linear System Solver Components
Masha Sosonkina, Dane Coffey, Fang Liu, Randall Bramley

14:50 **Break** *[Redtenbacher Foyer, Building 10.91]*

15:30 **Discussion of sessions P2, P3 and A2, as well as closing discussion** *[Mittlerer Hörsaal, Building 10.91]*

Invited Sessions

Keynote Talks

Components, Architecture and the Law - A Tricky Interplay

Speaker: Prof. Dr. Thomas Dreier, Universität Karlsruhe (TH), Germany
CBSE/QoSA, Tuesday, 14.10.2008, 11:00 *[Redtenbacher, Building 10.91]* W

The law sets the framework for any social, and hence also any technical activity. It prescribes what to do and what not to do, and it also contains rules on legal consequences in case the legal rules are not followed. From a software architect's perspective, legal rules may at times appear to work as a "brake" on technological innovation. However, it should be kept in mind that with regard to software, the primordial role of the law is to enable innovation, while at the same time protecting the integrity and property of software users from potential software related risks. Moreover, in contractual settings, the law provides mechanisms for solving future conflicts which might arise between the parties concerned.

The relationship between law and technology is not a one-way-street. Rather, by way of influencing upon each other, the law puts certain limits on technological activities while at the same time being formed by technology. At the center of this development are issues of what constitutes justice; how to balance conflicting freedoms of software producers and software users; and how to avoid and allocate the risks which come with the use of defective software, both from an economic and a social perspective.

The keynote speech will present and discuss the legal tools and methodologies available to answer these questions and it will give an overview of the areas of law that are relevant to software architecture, such as intellectual property rights, contract rules and rules of liability. In particular, issues of copyright and open source (OS), contractual questions and liability issues will be discussed with regard to component-based software engineering. Moreover, special attention will be given to the legal issues raised by predicting and maintaining quality of service (QoS). This gives rise to the central question which is not always easy to answer: when will software components be free of legally relevant defects? To what extent is or should the imperfection of complex systems and its components be subject to legal control?

In sum, this keynote speech attempts to raise the awareness of both software engineers and lawyers as regards the necessity of a common dialogue which not only looks at technology and the law, but which likewise takes into consideration economic and social concerns.

Dealing with Design Erosion – Architecture Refactoring

Speaker: Dr. Michael Stal, Siemens AG, Germany
SID, Wednesday, 15.10.2008, 09:00 *[Tulla, Building 11.40]* p

When developing software systems, engineers should follow an approach of piecemeal growth where they refine the software architecture step by step until the system finally is complete. Unfortunately, “nothing is permanent except change” (Heraclitus). Hence, a pure top-down process could only work in practice if all requirements are complete and consistent from day 1, and architects and developers never encounter any bugs. In the real world, change is the rule, not the exception. Hence, not systematically coping with change leads to software systems that are hard to maintain due to increased entropy. It is not sufficient to just add parts in each step of architecture design. Instead, software architects should review and clean up their intermediate solutions before addressing further refinement activities. Unfortunately, existing literature broadly covers code refactoring, although refactoring principles are likewise applicable to other artifacts, especially to software architecture.

The keynote introduces the foundation of software architecture refactoring. It illustrates how to leverage architecture refactoring in a development process. But it also presents its limitations. For example, it explains when reengineering might be a better option. The keynote focuses on the principles of software architecture refactoring, and gives a brief overview of a catalog of architecture refactorings that should be in the toolbox of every software architect.

Rethinking the Use of Models in Software Architecture

Speaker: Prof. Dr. Carlo Ghezzi, Politecnico di Milano, Italy
QoSA/CBSE, Thursday, 16.10.2008, 09:00 *[Redtenbacher, Building 10.91]* W

Models play a central role in software engineering. Traditionally, they are used to reason about requirements, to identify possible missing parts or conflicts. They are also used at design time to analyze the effects and trade-offs of different architectural choices before starting an implementation, anticipating the discovery of possible defects that might be uncovered at later stages, when they might be difficult or very expensive to remove. Finally, they may be used at run time to support continuous monitoring of compliance of the running system with respect to the desired model.

The talk emphasizes the requirements for model usage in the context of evolving applications that are situated in an open-world. In this setting, models should be kept alive at run time to support software adaptation due to changes that affect running systems. The talk also illustrates possible scenarios that show how this can occur, focusing in particular on models that support analysis of non-functional system properties — namely, performance and reliability.

Focus on the Component Model

Speaker: Dr. André Ribes, EDF, France: Salome Platform
CBHPC, Thurdsay, 16.10.2008, 09:00 *[Mittlerer Hörsaal, Building 10.91]* S

Numerical simulation is more and more widely used for the design of new products and the improvement of technologies. Within Electricité de France, numerical simulation is also used to determine the evolution of equipments like nuclear reactor or hydraulic dam during their lifespan. This kind of technology is one of the major tool that allows to produce and transport energy in good conditions.

Nevertheless, using numerical simulation codes is a complex task. Indeed, a numerical study is not only a run of numerical applications but it involves many pre-processing steps such as CAD, meshing or data settings, code deployment, and post-processing steps like visualization. Therefore, a scientist who needs to carry out a study has to use many different pieces of software, each piece may use its own data format and its specific framework.

Salomé is a platform for numerical simulation whose aims is to address these issues. It provides on the one hand a unified framework offering all the basic functionalities like mesh and visualization modules, and on the other hand a programming model to embed numerical codes. To be able to achieve realistic results and to be used on a wide range of applications, Salomé provides a module named YACS (for dYnamic pArallel Coupling System) that allows to design and control execution of calculation schemas on computer networks and clusters. A calculation schema is mainly a graph of nodes that refer to computational tasks or control structures. Interconnected scientific applications like code coupling applications can be seen as a collection of computational tasks that are executed in a specific order.

To be able to provide these features, Salomé provides a distributed component oriented programming model. This model is divided in three main parts. First, Salomé provides an extension of CORBA object model that adds dynamic interfaces to the distributed object model (named DSC for Dynamic Software Component). Second, a static description of the components is used by YACS to create calculation schemas. Finally a deployment service is provided by the platform to launch the components into different distributed resources. This keynote describes how the Salomé platform uses the component technology to create numerical simulations.

Visualizing and Managing the Evolution of Socio-Technical Systems of Systems

Speaker: Prof. Dr. Florian Matthes, TU Munich, Germany
CBSE/QoSA, Friday, 17.10.2008, 09:00 *[Redtenbacher, Building 10.91]* W

Application landscapes in large enterprises consist of hundreds or thousands of highly connected semi-autonomous application systems which are designed, created, evolved, maintained, used and financed by people with diverse interests and sometimes incompatible educational background. We report on recent efforts in academia and industry to improve the long-term and strategic management of this core enterprise asset by improving the communication between these stakeholders. A key challenge is to develop models, visualizations, tools and management practices which simultaneously address social, technical and economic aspects in a balanced and pragmatic manner.

Architecture and Agility – an oxymoron?

Speaker: Prof. Philippe Kruchten, University of British Columbia, Canada
CBSE/QoSA, Friday, 17.10.2008, 13:30 *[Redtenbacher, Building 10.91]* W

Software architecture is taking a bad rap with many agile process proponents; BUFD = big up-front design, massive documentation, smell of waterfall, ... it is pictured as a non-agile practice, something we do not want to even consider (though everybody want to be called an architect). However, certain classes of system, ignoring architectural issues too long “hit a wall” and collapse by lack of an architectural focus. Agile architecture: a paradox, an oxymoron, two totally incompatible approaches? I’ll review the real issues at stake, try to go past the rhetoric and posturing, and suggest that the two cultures can coexist and support each other, where appropriate.

Tutorials

The CompArch tutorials, all presented within the Software Industrial Day, are going to take place in three lecture rooms: Mittlerer Hörsaal, Building 10.91; Seminar Room 202 and Seminar Room 214, Building 11.40. Specific assignment of the rooms to the tutorials will be specified on site.

Systematically Designing Component Frameworks

Speaker: Wolfgang Weck, independent Software-Architect, Zürich, Switzerland
SID, Wednesday, 15.10.2008, 13:30

A component framework has been defined as a set of interfaces and rules of interaction that govern how components ’plugged into’ the framework may interact. Typical component frameworks also provide an implementation that partially enforces these rules of interaction. The implementation of the component framework and those of the participating components remain separate.

In this tutorial we will learn a few lessons from looking at a small but easy to understand (toy) example. (i) Plug-and-play composition relays on standardized contracts. (ii) These contracts often are symmetric and will often consist of several interfaces (in the sense of programming languages) defining various roles. (iii) Defects in plug-and-play components may cause hard to resolve problems to third-party composers, who then experience component hell. (iv) Contract design can help to avoid these problems, and (v) this may require some small implementation to become part of the contract itself.

Reflecting these lessons and the illustrating example, we will derive a systematic way to design component frameworks.

Model-based Software Performance Prediction

Speaker: Dr. Steffen Becker, FZI Forschungszentrum Informatik, Germany
SID, Wednesday, 15.10.2008, 13:30

The ability to predict extra-functional properties of software systems at design time based on system models is a cornerstone of software engineering’s way to become a true engineering discipline. Such predictions allow the systematic analysis of different design alternatives which leads to informed design decisions in contrast to the still frequently used ad hoc decisions.

Software components provide a mean to structure software systems, but also to reuse existing code parts in different contexts. Because of this, their documentation is usually assumed to be more enhanced including extra-functional specifications.

In the tutorial, we demonstrate the use of the Palladio Component Model (PCM) and its tools to model a component-based software system and some design alternatives. By applying the performance prediction methods available in the PCM, the tutorial demonstrates how the PCM helps in choosing between design options.

More detailed, the tutorial provides a conceptual overview on the PCM and its modelling artefacts. It introduces the modelling of components, architectures, resource environments and user behaviour as well as their enrichment with performance annotations. In a practical part, the PCM's tools and their usage are shown to analyse design options on a small example system.

Principles of Service-Oriented Architectures

Speaker: Prof. Dr. Gregor Engels, University of Paderborn, Germany
SID, Wednesday, 15.10.2008, 13:30

Service-oriented architectures (SOA) are promoted as the final answer to the long standing demand to bridge the gap between business needs and IT-solutions. Like object-orientation has been advocated as the overall structuring principle in the 90s, service-orientation is nowadays discussed as the magic bullet to align business requirements with software applications. As it is typical for any hype approach, a commonly agreed understanding of the basic principles and notions for service-oriented concepts is still missing. This leads often to misunderstandings and communication problems in IT-projects and, thus, jeopardizes the success of migration projects where existing architectures are advanced towards service-oriented architectures.

The tutorial aims at clarifying the understanding of basic principles, notions, languages as well as methods of a service-oriented approach. All introduced concepts are related to existing knowledge on software architectures, architectural styles, and architecture frameworks. Presented concrete methodical guidelines have been derived from successful industrial IT-projects and are related to Quasar Enterprise, the service-oriented development approach of Capgemini sd&m AG, Munich (Germany).

Panel Discussion

Extra-functional Contracts Versus Service Level Agreements

Panelists:
Ivica Crnkovic, Mälardalen University
Ian Gorton, Pacific Northwest National Laboratory
Clemens Szyperski, Microsoft
Wolfgang Theilmann, SAP Research
Moderator: Mircea Trifu, FZI Research Center for Information Technology
SID, Wednesday, 15.10.2008, 17:30 *[Tulla, Building 11.40]* p

The purpose of this panel is to bring together experts from the CBSE, SOA and GRID communities and to start a dialog about QoS, a topic common to all of these fields. Dependable QoS characteristics are of fundamental importance in order to further drive the economy in general and the industrialization of the IT sector in particular. In order to get a comprehensive view on QoS topics different disciplines have to synchronized, most important the disciplines of software engineering, service computing and Grid-like infrastructures. The possibility to model, predict, measure, guarantee and monitor certain QoS levels for critical services or components is not only a desirable feature but a necessity for all of these fields. This is anything but trivial, because QoS depends on so many factors: infrastructure including middleware and virtualization layers, the entire software stack including external components / services and usage profile. Because these communities focus on limited sets of specific issues, they have their own limited views on QoS. In order to realize the vision of end-to-end QoS, these views have to be reconciled into a comprehensive QoS model spanning several axes and covering design-time and run-time concerns, low-level (infrastructure), medium-level (software) and high-level (business) aspects, component / service aggregation and decomposition, different stakeholder perspectives, etc.

Organizers:
Dr. Wolfgang Theilmann, SAP AG, Germany
Mircea Trifu, FZI Forschungszentrum Informatik, Germany

Social Events

Receptions

We have organised two receptions for CompArch participants.

On Tuesday, we will have a reception in the Schloss Karlsruhe (palace of Karlsruhe). You find the palace on the map of Karlsruhe in your conference bag. Additionally, signs will lead you from the venue to the palace (10 minutes walking distance). The reception begins at 17:30.

On Wednesday, the Software Industrial Day will end with a reception in Tulla Foyer (see the Room Plan at the end of this booklet). The reception will begin at 19:00 right after the panel discussion.

Note: For drinks at the receptions, you will find two vouchers each in your conference bag. You can get one drink per voucher at the receptions. Additional drinks can be paid directly to the staff.

Steam Engine Train Trip to the Black Forest

On Thursday, we will go to **Bad Herrenalb** in the Black Forest with a steam engine train. In Bad Herrenalb, you will have time to explore the surroundings. Afterwards, we will have the conference dinner before heading back to Karlsruhe. Below, the schedule for the trip is given.

15:30 Buses to the train station (in Ettlingen) leave near the Schloss Karlsruhe (see indication on the campus map in Figure 1, signs will lead you there from the venue).

16:15 Departure of the steam engine train in Ettlingen

17:00 Exploring Bad Herrenalb. Choose whether you want to take a walk through the historic town of Bad Herrenalb, or whether you prefer to do a short hike in the surrounding woods. It is also possible to directly go to the Kurhaus restaurant and the surrounding park and have a coffee there (not covered by the social event fee). *See below for the options in Bad Herrenalb.*

18:00 Meeting at the Kurhaus for the Conference Dinner

20:45 Heading back to the Bad Herrenalb train station

21:10 Steam engine train departs back to Ettlingen

22:00 Arrival in Ettlingen. You can either take our buses back to Karlsruhe, or stay in picturesque Ettlingen a little longer and take a tram back.

Bad Herrenalb

There are three options for guided trips, starting at the train station in Bad Herrenalb (see Figure 2). Meeting point at the train station in Bad Herrenalb – umbrella with the indicated colour.

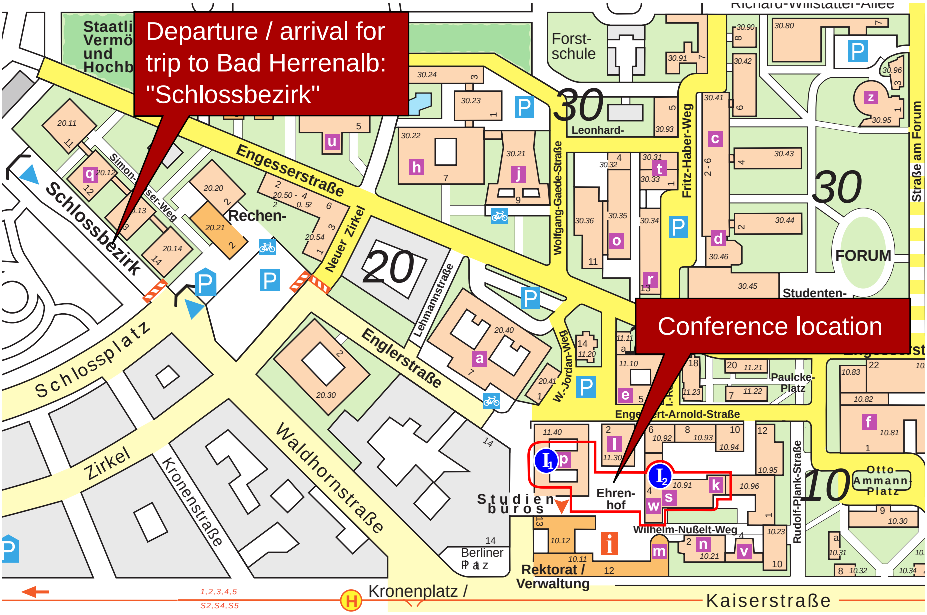


Fig. 1. Location of departure / arrival in Karlsruhe

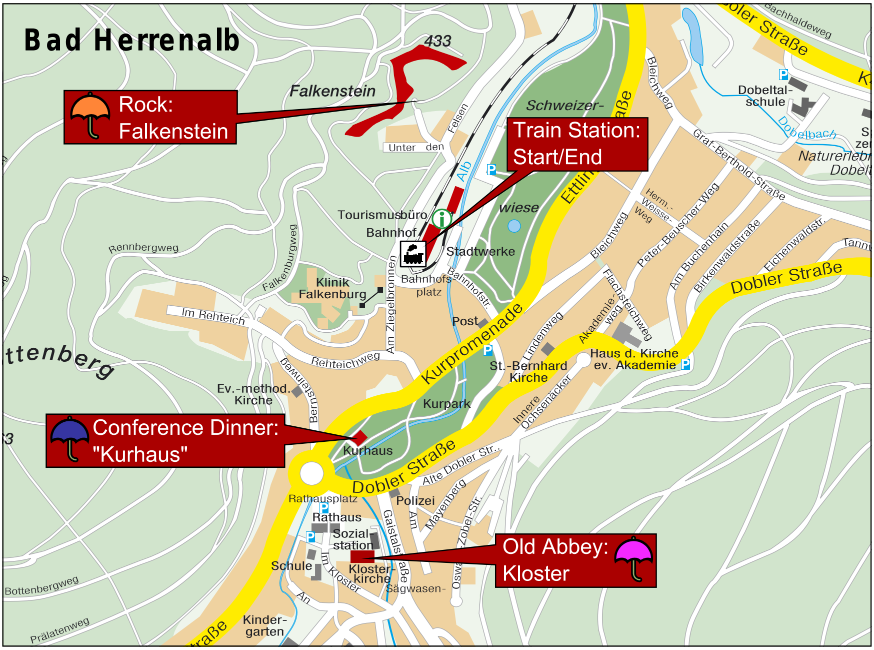


Fig. 2. Map of Bad Herrenalb

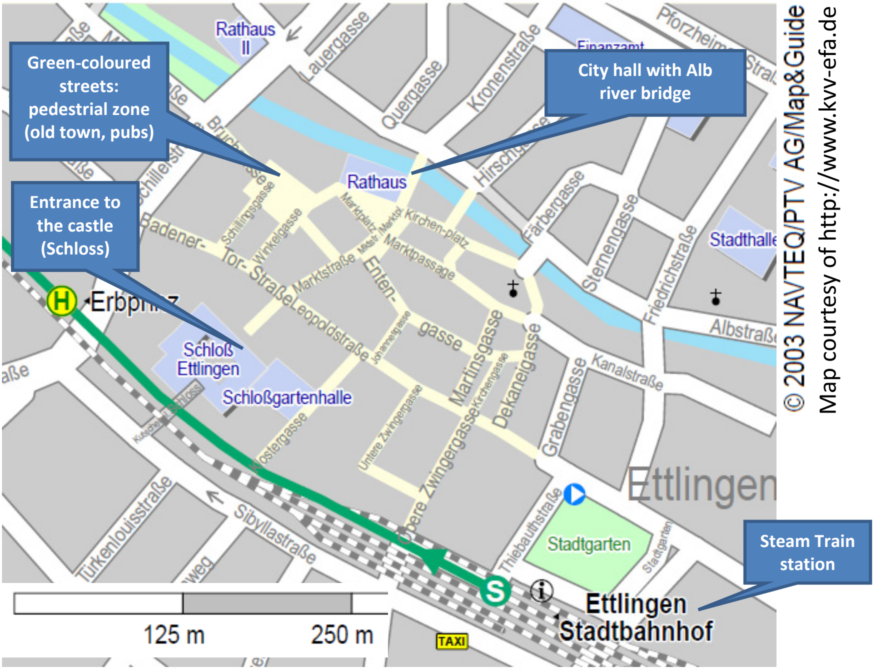
Trip options:

- (Orange umbrella) Falkenstein trail: “Sporty” hike to rock Falkenstein. View from the top on the Bad Herrenalb. (1 hour; 73 m difference in altitude; sturdy shoes recommended)
- (Pink umbrella) Walk to the old abbey ruin and church. Get to know the historic part of Bad Herrenalb. (1 hour)
- (Navy blue umbrella) Short path to the conference dinner location (“Kurhaus”). Have a coffee (not covered by the social event fee; 10 minutes walk)

All trips end in the “Kurhaus” where the conference dinner takes place.

Ettlingen

...for explorers who do not take the bus back to Karlsruhe...
The steam train terminates at “Ettlingen Stadtbahnhof” (also announced “Ettlingen Stadt”). After a walk, take light rail trains (trams), line S1 or S11, which leave “Ettlingen Stadtbahnhof” towards Karlsruhe (signed “Hochstetten” or “Neureut Kirchfeld”) at 22:28, 22:48, 23:08, 23:28, 23:48, 00:18, and the *final one at 00:48*.



© 2003 NAVTEQ/PTV AG/Map&Guide
Map courtesy of http://www.kvv-efa.de

Fig. 3. Map of Ettlingen

Further details:

- The 24h-ticket “Citykarte” is valid for the trip from Ettlingen to Karlsruhe (there is a 5-persons-variant of it as well). For single-ride tickets, you’ll need to pay for 3 zones, which is 2,50 EUR. There are ticket-vending machines at the station.
- The best stop to get off in Karlsruhe is Karlsruhe Marktplatz, right next to the pyramid. A tram line chart is inside the tram, stations are announced.
- “Ettlingen Erbprinz” stop is closer to the Ettlingen’s castle than “Ettlingen Stadt”
- If needed, a taxi stand is near “Ettlingen Stadtbahnhof” – ask for the fare! Taxis can be called at +49-(0)7243-77888, +49-(0)7243-78588, or the indicated number.

Note: An additional coloured map of Bad Herrenalb and Ettlingen will be handed out on Thursday.

Towards Food Supply in Karlsruhe

Johannes Stammel
FZI Research Center For Information Technology, Karlsruhe, Germany
stammel@fzi.de

Abstract Food supply is an important issue in nowadays research community, especially during long and exciting conference meetings. Therefore in this paper we propose a method of how to find good restaurants, bars, cocktail bars, lounges, cafés and other food suppliers in Karlsruhe.

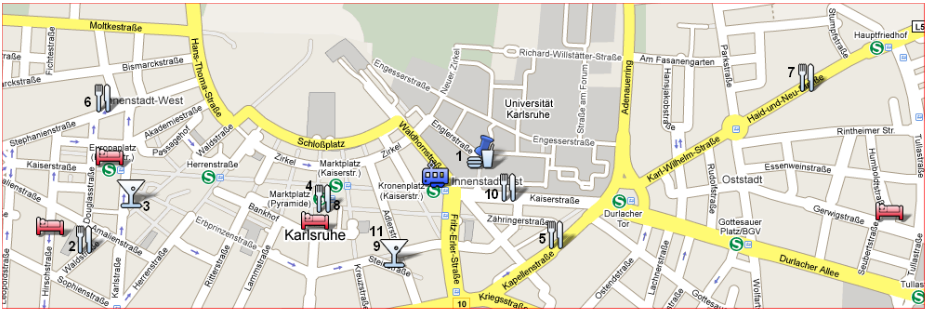


Fig. 4. Restaurant Map

Fast and Budget-Wise

Ballermann [1] Directly attached to university campus you find *Ballermann* where hot snacks and hot or cold drinks are served.

Name	Ballermann
Keywords	Snacks, Drinks
Address	Englerstr. 14, 76131 Karlsruhe
Phone	(07 21) 694417
Open	7 am. to 3 am.
Distance	Directly attached to conference venue
Web	http://www.ballermann1.de

Stövchen [2] Here you find a broad offer of breakfast and typical German food between 3 and 6 Euro. Warm food is served during whole day. It is well-known for its selection of Tarte Flambes. A wide range of *Hoepfner* beer flavors is available.

Name	Stövchen
Keywords	Breakfast, Lunch, Dinner, Beer, Tarte Flambes
Address	Waldstraße 54, 76133 Karlsruhe
Phone	(07 21) 2 92 41
Open	Su - Th 9 am. to 1 am. Fr - Sa 9 am. to 3 am.
Breakfast	Mo - Fr 9 to 12 am. Sa - So and public holidays 9 am. to 3 pm.
Distance	About 20 minutes from University
Tram	Lines 1, 4, S5, 3, S2
Tram Stop	Europaplatz
Hotels	Close to *Hotel Ambassador*
Web	http://www.stoevchen.com/index.htm

Popular Restaurants

Zum kleinen Ketterer [11] *(Professor's Hint I)* The *Zum kleinen Ketterer* is a restaurant serving good and solid food and down-home dishes. The host of this location is from England.

Name	Zum kleinen Ketterer
Keywords	Dinner, Beer, Wine
Address	Adlerstr. 34 / Corner to Markgrafenstr.
Phone	(0721) 354 00 99
Open	11.30 am. - 11.00 pm.
Distance	About 20 minutes from University.
Tram	Almost all lines
Tram Stop	Marktplatz
Hotels	Close to *Hotel Kaiserhof*
Web	http://www.kleiner-ketterer.de

Lehner's Wirtshaus [3] The *Lehner's Wirtshaus* is a cocktail bar and restaurant in Bavarian style. Typical Bavarian food is offered. There is a daily happy hour from 10:30 pm to 1:00 am with half-price cocktails.

Name	Lehner's Wirtshaus
Keywords	Dinner, Drinks, Cocktails
Address	Karlstraße 21, Ludwigsplatz
Open	11 am to 1 am.
Distance	About 20 minutes from University.
Tram	Almost all lines
Tram Stop	Europaplatz
Hotels	Close to *City Hotel*

Die Krone [4] *Die Krone* is a café house in modern Italian style with two floors. Warm food is served all day. Cocktails are offered. At Thursdays after 7 pm. cocktails are offered at half-price. WLAN can be used free of charge. Smoking room available.

Name	Die Krone
Keywords	Breakfast, Coffee, Dinner, Cocktails
Address	Karl-Friedrich-Straße 8, Marktplatz
Open	7:30 am to 1:00 am.
Distance	About 15 minutes from University.
Tram	Almost all lines
Tram Stop	Marktplatz
Hotels	Close to *Hotel Kaiserhof*
Web	http://www.krone-ka.de

Vogelbräu [5] *(Professor's Hint II)* The *Vogelbräu* is a local brewery of Karlsruhe. They serve an unfiltered beer, a *Pils* beer, and special beers depending on the season ranging from wheat beer to smoke beer. They provide good and solid food, down-home dishes, simple fair food from the region. The main house with a pleasant beer garden is located on the east side close to Karlsruhe city centre. A smoking area is provided. The *Vogelbräu* is a very attractive location and a must-see for visitors who like drinking beer.

Name	Vogelbräu
Keywords	Breakfast, Lunch, Dinner, Beer
Address	Kapellenstraße 50
Phone	(07 21) 37 75 71
Open	Mo - Su 10 am. to 1 am.
Breakfast	Mo - Fr 10 am. to 4 pm.
Distance	About 10 minutes from University
Tram	Lines 1, 2, S4, S5, 4, 5
Tram Stop	Durlacher Tor
Web	http://www.vogelbraeu.de/

Badisches Brauhaus [6] The *Badisches Brauhaus* is a local brewery, where freshly brewed beer is served as well as German-Alsatian food. The building has four levels which are connected via stairs and a slide.

Name	Badisches Brauhaus
Keywords	Lunch, Dinner, Beer
Address	Stephanienstraße 38-40
Phone	(07 21) 1 44 - 7 00
Open	Mo to Th: 11:30 am to 0:00 am. Fr and Sa: 11:30 am. to 01:00 am. Su: 11:00 am to 0:00 am.
Distance	About 20 minutes from University.
Tram	Line 1, 3, 4, S2, S5
Tram Stop	Europaplatz
Hotels	Close to *City Hotel*
Web	http://ka.stadtwiki.net/Badisch_Brauhaus

Hoepfner Burghof [7] The *Hoepfner Burghof* is a hotel and restaurant in the east part of the city. Hoepfner is the second largest brewery of Karlsruhe. Here you can try several kinds of beer, but also eat typical German food based on local customs.

Name	Hoepfner Burghof
Keywords	Lunch, Dinner
Address	Haid-und-Neu-Straße 18
Phone	(07 21) 6 18 34 00
Warm Food	Mo to Sa: (non-stop) 11.30 am. to 9.30 pm Su / Public Holidays: 11:30 am. to 2:00 pm., 6:00 pm to 9:00 pm.
Distance	About 15 minutes from University
Tram	Lines 4, 5
Tram Stop	Karl-Wilhelm Platz
Hotels	Close to *Hasen Hotel*
Web	http://www.hoepfner-burghof.de

Besitos Tapas [8] *Besitos* is a tapas bar in Spanish style. Warm food is served, especially tapas and other Spanish food. Classical and Mediterranean cocktails. Happy hour from 5 pm to 8 pm and after 10 pm.

Name	Besitos Tapas
Keywords	Dinner, Spanish food, Cocktails
Address	Marktplatz im Weinbrennerhaus
Open	7:30 am to 1:00 am.
Distance	About 15 minutes from University.
Tram	Almost all lines
Tram Stop	Marktplatz
Hotels	Close to *Hotel Kaiserhof*
Web	http://www.besitos.de

Deluxe

Alte Seilerei [10] *(Professor's Hint III)* The *Alte Seilerei* is a restaurant and wine house, where noble claims are satisfied. Fully-fledged menus are served and a wide selection of wine is offered.

Name	Alte Seilerei
Keywords	Dinner, Wine, Menus, Higher Price Segment
Address	Kaiserstraße 47
Phone	(07 21) 38 41 95 4
Open	Tu to Sa: 6 pm. to 0 am. Su and Mo: closed
Distance	About 5 minutes from University. Directly opposite of university main building
Tram	Line 1, 2, 3, 4, 5, S4, S5
Tram Stop	Kronenplatz
Web	http://ka.stadtwiki.net/Seilerei

Carlos Cocktails [9] At *Carlos Cocktails* a wide range of cocktails is served. Its owner Carlos has won several awards at national and international competitions.

Name	Carlos Cocktails
Keywords	Cocktails
Address	Markgrafenstraße 32, Lidellplatz
Distance	About 15 minutes from University.
Tram	Almost all lines
Tram Stop	Marktplatz
Hotels	Close to *Hotel Kaiserhof*

All locations can be also found in a google map (named *Comparch Restaurant Guide*) , which is linked from the CompArch Webpage.

Part II

Industrial Reports

Analyzing the Extensibility Options of Business Software Solutions

Marcus Echter

SAP AG
marcus.echter@sap.com

Abstract. Extensibility as a quality attribute plays a significant role in the context of business software. A valuable comparison between the extensibility options of different solutions is crucial for the right purchase decision, but not straightforward. Existing papers describe rather informal analyses that are not based on empirical studies. SAP AG applied the Goal Question Metric approach to develop a generic reference model in cooperation with the University of Karlsruhe. This model can be used for an empirical analysis of the extension options of enterprise software. A case study finally compared SAP's new "Business ByDesign" solution with two other competitors and showed the practical applicability and limitations of the aforementioned model.

1 Introduction

Enterprise software solutions have to be flexible enough to adapt to ever-changing requirements. This is crucial for a company's market success, as today's business processes heavily rely on software and are frequently subject to change. Over the years, SAP AG has developed enterprise software that offers various configuration and extension possibilities explicitly designed into the architecture of the solution. SAP joined the University of Karlsruhe (TH), which is famous for its research in software design and quality, to create a generic model for analyzing extension options of enterprise software. This model can also be used by companies that are willing to buy a new solution and want to know which one fits best to their requirements.

Currently, there are no empirical studies that compare the extension possibilities of enterprise solutions based on a formal model. [Par79] gives an overview on how to generally design software with regard to extensions and modularity. [Dom04] describes some basic adaptation options of ERP (Enterprise Resource Planning) systems with no reference to concrete implementations. [Uns04] and [Hut03] finally compare some select business solutions in an informal way. The aforementioned papers do not imply the usage of any kind of formal model or metrics for their analyses. To fill this gap, we applied the Goal Question Metric (GQM) approach [BCR94] by Basili et al. to design accurate metrics based on defined goals and questions. These can be used for a valuable analysis of extension and adaptation possibilities as shown in a case study.

The contribution of this paper is the presentation of a domain-specific reference model for comparing the extensibility options of enterprise software as well as the selection of questions and metrics it is based upon. Section 2 develops this model, its underlying questions and metrics, discusses the assumptions made and shows its practical applicability by performing an example comparison of three select business solutions.

2 Design of the Reference Model

The GQM approach. The Goal Question Metric (GQM) approach [BCR94] by Basili et al. is a systematic method for a goal-oriented derivation of metrics. These metrics are developed in a top-down fashion based on predefined goals and questions. The concrete values can then be interpreted with regard to the formulated questions. The *goal* defines the conceptional level and consists of three dimensions: *issue*, *object* and v*iewpoint*. Moreover, it answers a particular *purpose*. A set of *questions* on the operational level characterize the attributes of the object and refine the examined issue. Each question is assigned a number of *metrics* that answer it on a quantitative level. These metrics contain the actual data of the examination.

Overall Goal and Sub-Goals. The overall goal of our examination is quite straightforward and can be formulated as follows:

> Comparison (*Purpose*) between the extensibility options (*Issue*) of enterprise software solutions (*Object*) from the viewpoint of a customer (*Viewpoint*).

In order to manage the complexity of the domain, we decided to split the overall goal into sub-goals which each describe a particular aspect of the respective object. These extension categories have proven their practical relevance. From the viewpoint of a customer, *Business Configuration*, *User Interface* and *Data Model* extensions represent the most important sub-goals. For them, we developed the according questions and metrics which are presented in the next two paragraphs.

Questions. The questions below a certain sub-goal represent extension scenarios that are relevant for the customer according to the viewpoint of the overall goal. They are based on countless interviews with product managers and topic experts.

In the area of business configuration, there are two main questions that have to be answered:

1. Can a key-user configure the system?
2. Can new configuration content be added to the system?

The most complex category considered is user interface extensions. As the user can only interact with the system via some kind of user interface, he expects special adaptation and personalization features. This leads to the following questions:

1. Can a user personalize tables on the screen?
2. Can a user personalize his work environment?
3. Can a key-user adapt screen labels to customer-specific terminology?
4. Can a key-user adapt the page layout?
5. Can a new page with a new sub-menu be added?
6. Can mashups be created?

For data model extensions, we described once more two main extension scenarios:

1. Can a key-user adapt pre-defined business objects?
2. Can a key-user add a new business object to the system?

Metrics. As mentioned above, metrics characterize and quantify several aspects of a particular use case. In this work, we concentrated on ordinal metrics because they enable a relative comparison without complex calculations that are hard to validate, as found with cardinal metrics.

After due consideration, we decided to define five general metrics that are applicable to several questions and sub-goals (cf. Fig. 1).

Metrics	**Scale**
Feasibility	yes (x), no (-)
Power	++,+,o,-,--
Effort	low, medium, high
Implementation	Customer, Partner, Producer
Implementation (Customer)	End-User, Key-User

Fig. 1. General Metrics

Besides the general availability (*Feasibility*), the developed model places emphasis on the *power* and the realization *effort* of a particular feature. The power can be seen as the weighted set of supported sub-features, evaluated by a scale from “very low” (--) to “very high” (++). Example sub-features for the feature “Table Personalization” are hiding/rearranging columns, sorting or changing the table design. The effort is estimated by “low”, “medium” or “high”, depending on how easy a user can implement the desired extension scenario. The final two metrics consider the issue *who* can implement a scenario generally (Customer, Partner, Producer) and within a company (End-User, Key-User).

Assumptions/Limitations. It is not easy to model a complex application domain like the extensibility of a software solution. In this paragraph, we explicitly discuss the assumptions made as well as the limitations of the model.

Although the considered sub-goals were formulated separately, they are not independent from each other. For example, an extension field of a business object is quite useless if it cannot be displayed in the user interface. However, a full end-to-end extension consideration as desired by the customer can only be done in a limited fashion with our approach.

Our questions refer to the most important extension scenarios as required by the customer. Further questions could be formulated that are not part of our examination.

Finally, the considered metrics cannot capture each and every aspect of a certain extension scenario. Some aspects like the look-and-feel of a user interface are inherently hard to metricize. In our examination, we therefore focused on the two main aspects “power” and “effort”.

Case Study. After having developed the aforementioned model, we showed its practical applicability in a case study. In this we compared SAP’s new midmarket solution “SAP Business ByDesign” with two competitors – Salesforce CRM and Oracle Fusion – with regard to their extension options. The comparison required a deep understanding of the extensibility features of the different solutions so that the generic template could be filled with concrete data. After all, the developed model proved valuable as a basis for detailed analyses and gave a clear tendence on the supported extension features in the respective categories.

3 Conclusions

We presented a generic model for analyzing the extensibility options of enterprise software solutions. It is based on extension categories that serve as subgoals of the overall analysis. For the three most important categories *Business Configuration, User Interface* and *Data Model*, we formulated questions that represent typical adaptation and extension use cases as required by the customer. The questions are answered by five generic metrics that are condensed into the two main indices *power* and *effort*. This model was then applied to a concrete comparison between three select business solutions. The case study showed that the model, despite some limitations, is a valid common template for an extensibility analysis. It can be applied by enterprise software vendors to delimit their products from competitors as well as by customers to help in doing the right purchase decision.

In our analysis, we included five categorial metrics. These metrics only permit simple relative comparisons and not complex aggregated analyses as enabled by cardinal metrics. If the latter are used, more sophisticated metrics like LOC can be designed that imply numerical measurements in the system. Furthermore, our examination concentrated on three select extension categories and did not span the wide area of process flexibility in particular. All these issues could be addressed in future studies.

References

[BCR94] Victor Basili, Gianluigi Caldiera, and H.Dieter Rombach. *The Goal Question Metric Approach.* In: Marciniak, J.J. (ed.) Encyclopedia of Software Engineering, pp. 528–532, John Wiley & Sons, New York, 1994.

[Dom04] Axel Domschke. *Anpassungsmöglichkeiten von Informationssystemen am Beispiel von ERP-Systemen.* FHWT Vechta/Diepholz, 2004.

[Hut03] Andreas Huthmann. *ERP-Systeme, Vergleich - Sage KHK, Navision, Bäurer*. University of Marburg, 2003.

[Ora] Oracle Fusion. http://www.oracle.com/fusion.

[Par79] David Parnas. *Designing Software for Ease of Extension and Contraction.* In: IEEE Transactions on Software Engineering, vol.SE-5, pp.128-138, 1997.

[Sal] Salesforce CRM. http://www.salesforce.com.

[Sap] SAP Business ByDesign. http://www.sap.com/businessbydesign.

[Uns04] Martin Unsöld. *Markt der Standardanwendungssoftware für ERP mit einem Vergleich der konzeptionellen Struktur der Angebote.* University of Würzburg, 2004.

Evaluating Failure Propagation in the Application Landscape of a Large Bank

Josef Lankes[1], Florian Matthes[1], Tarmo Ploom

[1] Technische Universität München

1 Motivation and Initial Situation

Today, enterprises operate a large number of applications providing critical support to the business. These applications form, when taken together, the application landscape, which can be seen as an important asset, providing essential support to business processes, but sometimes also acting as a limiting factor.

Important quality attributes depend not only on architecture and implementation of specific applications. The support an application landscape can deliver to business also depends on how the applications are integrated in the landscape.

This article focuses on failure propagation in an application landscape, which affects the availability at which the applications offer their specific services.

We applied metrics we introduced in [LS1] for evaluating failure propagation aspects in an application landscape on two proposals stakeholders from a large bank created to limit failure propagation. The evaluation was targeted at the subset of the landscape application supporting private banking, specifically the one located on the mainframe. This subset of the application landscape consists of 255 applications, organized into 75 subdomains, which are themselves organized into 18 domains. Together, the applications amount to about 12 millions lines of PL/1 code.

1.1 Proposals for Limiting Failure Propagation

Figure 1 shows, how these proposals intend to limit failure propagation.

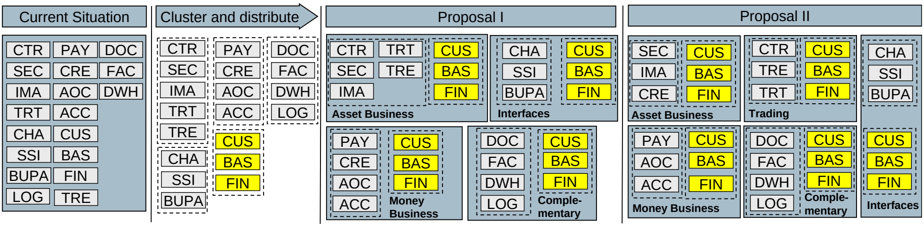


Fig. 1. Domains and their distribution in the proposals limiting failure propagation

Both proposals rely on making changes to how domains (small rectangles in Figure 1) are deployed and communicate. Thereby, the domains are organized into so-called domain clusters (rectangles with slashhed lines) along functional concerns. In the as-is landscape, one platform (large rectangle with full line) hosts all domains. Contrastingly, the proposals distribute the domain clusters to different platforms, as illustrated on the right side of Figure 1: Proposal I introduces the platforms Money Business, Asset Business, Interfaces, which offers functionality to customers and suppliers, and Complementary, which contains non-banking functionality. Each platform hosts a specific domain cluster and a replication of the fundamentals-cluster, which provides basic data and services.

The platforms are independent, therefore, only asynchronous communication is allowed between them. The data used by the fundamentals-domains is replicated between their different deployments. Only one deployment of a fundamentals-domain is allowed to change its data, the other deployments are restricted to read-only access.

The two proposals differ in the number of platforms they create. Proposal II uses an additional platform, into which it puts an replication of the fundamentals-domains, and the domains CTR, TRT, and TRE.

2 Metrics for Assessing Failure Propagation on Application Landscapes

In order to evaluate the two proposals in respect to their ability to limit failure propagation in the application landscape, we used the metrics introduced in [LS1]. The metrics are calculated on detailed information about the targeted subset of the application landscape, which was structured as shown by Figure 2.

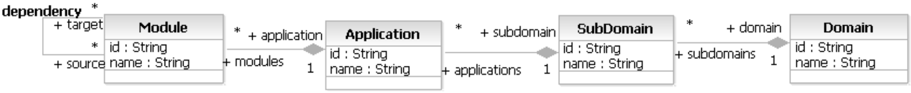


Fig. 2. Information model of the data available about the application landscape

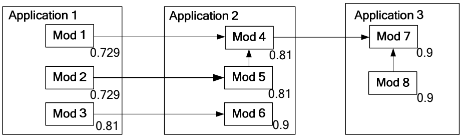


Fig. 3. Calculating failureProbability on information according to Figure 2

Figure 3 exemplifies, how the metrics were calculated on data structured as shown in Figure 2. This calculation procedure assumes, that a module fails, if one of the modules it depends on, also transitively, is no longer able to render its services, as the respective Application has failed. For calculating the failureProbability of a module m, the algorithm starts at m, and derives the set of modules which are, also transitively, called by m. Then, it derives the set of Applications, in which these modules are located. These are the Applications which need to be operational for the module under consideration being able to render its services. n is the size of this set. Assuming, that Applications fail independently with availability[1] A , the failureProbability of m is $1 - A^n$.

3 Evaluating Proposals for Limiting Failure Propagation

failureProbability, as described in Section 2, has been calculated for all modules. The probability of the complementary events 1-failureProbability(m), interpreted as an availability of the respective module, was averaged over the respective Domains. These values (called averageServiceAvailability) were calculated for both the as-is application landscape, Proposal I, and Proposal II.

Figure 4 shows these results, visualizing each one of the three scenarios as a line, with the domains on the x-axis, and the y-coordinate of the line showing the averageServiceAvailability of the respective domain. If a proposal has multiple deployments of a domain, the respective graph visualizes an average value.

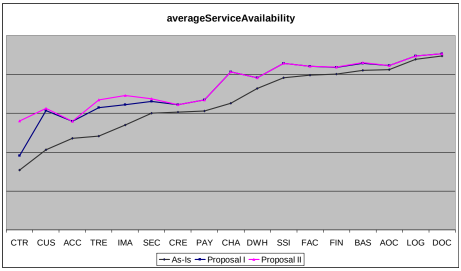


Fig. 4. Comparing the as-is landscape, Proposal I and Proposal II

In interpreting these results, the stakeholders went into two directions. On the one hand, the evaluation results gave them new insights into their proposals, showing especially, that the domains benefit largely differently from the proposals. Especially two domains, SEC and CRE, did not improve as expected.

On the other hand, the stakeholders discussed the assumptions underlying the evaluation. One point related to the assumption that dependencies crossing platform borders can be replaced by messaging, and then be considered having only minor impact on availability. This assumption can be disputed. If a dependency reads data, the need for this data does not disappear when messaging is

[1] Via the shared availability, the approach focuses on the landscape, and not on characteristics of specific Applications. However, it could be extended into this direction.

introduced. If the data is not returned within a specified time, this still constitutes a failure. However, as the dependency information does not contain the directions of the data flows, above evaluations are still used as an (possibly optimistic) approximation. This confirmed to the stakeholders, that projects need improved data about the application landscape. They limited this statement not only to above proposal comparison, but directed it at (IT-) projects in general.

Moreover, above evaluation did not consider an effect which can be exepected from the replication of basic functionality: Reduced likelihood of large failure events, which affect a high share of modules. In order to assess this effect, failure distributions were estimated[2]. These distributions indicate the probability of failures involving differently large shares of the modules in the landscape. They are shown for the as-is landscape in Figure 5, and for Proposal I in Figure 6, and clearly indicate, that Proposal I reduces the likelihood of large failures.

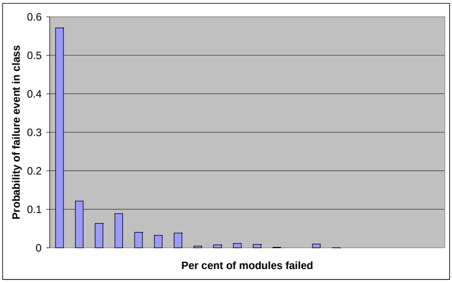


Fig. 5. As-Is

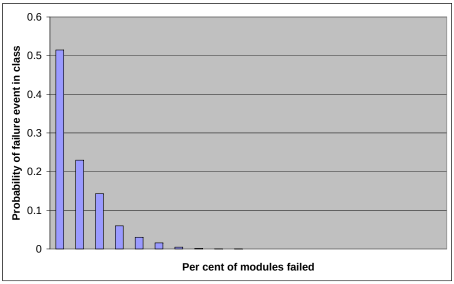


Fig. 6. Proposal 1

4 Resume and Outlook

Besides providing the stakeholders with information about their proposals, the metrics analyses helped them to refine their questions about the proposals. For example, reduced likelihood of large failures was not directly discussed before the metrics analyses. This resulted in a more systematic discussion of approaches for limiting failure propagation, based on explicit assumptions and information.

Thus, above described case showed, that metrics at the application landscape level are a suitable aid in the evolution of the respective architectures. Currently, we are researching into approaches for assessing quality attributes related to throughputs, latencies, but also changeability at application landscape level.

References

[LS1] Lankes, J., Schweda, C.: Using Metrics to Evaluate Failure Propagation in Application Landscapes. Multikonferenz Wirtschaftsinformatik, 1827–1838 (2008)

[2] via Monte Carlo simulations, exact calculation was too computation intensive

Index-based Process and Software Quality Control in Agile Development Projects

Nicole Rauch and Eberhard Kuhn and Holger Friedrich

andrena objects ag
[Nicole.Rauch,Eberhard.Kuhn,Holger.Friedrich]@andrena.de

Abstract. In software development, it is important to assure a high level of process and software quality. In the agile context, suitable approaches to measure and analyze these aspects are hard to find. CMMI and SPICE are too heavyweighted, while EN ISO 9001 is too lightweight. To fill this gap, andrena objects ag developed ISIS, a navigation system for process and software quality management that combines several metrics. It is based on more than 200 person years of software development experience. ISIS proved its practicability in several customer projects and was certified according to EN ISO 9001 in late 2007.

1 Introduction

For software companies, a high quality level of their products is important, allowing for efficient maintenance and extensibility. Market success relies on an optimal cost-benefit-ratio. Over the past years, andrena objects ag has established a continuous quality improvement process. To achieve a high software quality, the developers' qualification, motivation, and creativity are crucial. For a satisfactory process quality, andrena applies XP [Bec03] and Scrum [Sch04]. In addition, quantitative quality measurements and analyses are highly desirable.

The agile development process followed at andrena is characterized by few but guiding rules. Existing approaches like CMMI [CKS06] and SPICE [Loo07] are far too restrictive and inefficient to be applied to an agile process and thus are aloof the cost-benefit-optimum. EN ISO 9001 [Cas06], on the other hand, defines too few rules to be useful as guidance in software development since the metric "customer satisfaction" is the only index controlling the process. In real projects, the vast amount of available metrics [Kan02] is widely being ignored due to time restrictions. Life sciences, for example, follow a different path. To evaluate the water quality, they also feature innumerable metrics, but the analysis only takes place based on a highly restricted set of *indicators*. This led us to designing a method allowing systematic quantitative control and optimization of the process and software quality based on a small number of metrics. Our goal is to develop software of adequate quality while at the same time being highly productive.

The contribution of this paper is the presentation of the ISIS navigation system as well as the selection of metrics and tools it is based upon. Section 2 introduces ISIS, its metrics and underlying tools, presents the integration into the Scrum process and EN ISO 9001 and gives an experience report.

Metric	**In PQI**	**In SQI**
customer satisfaction	17 %	
Number of bugs	15 %	
Deviation from approximated time usage	11 %	
Test coverage	Δ 13 %	23 %
Packages in cycles	Δ 11 %	19 %
Average Component Dependency (Class)	Δ 9 %	16 %
Classes > 20 methods	Δ 6 %	10.5 %
Methods > 15 LOC	Δ 6 %	10.5 %
Cyclomatic Complexity (num. methods > 5)	Δ 10 %	17.5 %
Compiler warnings	Δ 2 %	3.5 %

Fig. 1. Metrics incorporated into process and software quality index (PQI and SQI).

2 ISIS

System Overview. To supplement the agile development methods Scrum and Test-Driven Development, andrena developed ISIS, a navigation system for quality management. ISIS' main component is the project logbook. Characteristics regarding process and software quality are recorded, condensed, and documented in time series. It offers a continuous comparison to predetermined quality goals. Erroneous trends can instantaneously be counteracted. ISIS supports developer teams in keeping their projects on track. ISIS also offers a high degree of transparency to the project management and to the IT management. They have immediate access to objectified indicators for process and software quality.

Included Metrics. ISIS is based on a number of metrics, i.e. indices that describe certain aspects of source code. The metrics included in ISIS should easily be collected and interpreted, and it should be hard to manipulate them. Furthermore, the selected indicators should cover the overall quality as well as possible. To keep the analysis manageable, only a small set of indicator metrics was selected for the evaluation of the software quality. We regard software quality as being holistic, that is, we assume to be able to draw conclusions regarding the whole by only looking at parts. The same holds for the process quality. Although this assumption still lacks scientific validation, we noticed in many code reviews a strong correlation between the quality of architecture, design, and coding of software. Fig. 1 lists the included indicator metrics. The condensation into the two central control indices *process quality index* and *software quality index* applies a heuristic based on andrena's long standing experience in object-oriented software development.

Customer satisfaction is a highly integrated metric. It is decisive for customer oriented services such as software development. The external software quality is indicated by the *number of bugs.* A bug is defined as behavior that deviates from a given specification and that occurs at the customer site. Unfortunately, developers massively repress bugs instead of benefiting from the potential of learning and avoidance that can be utilized by collecting and analyzing programming errors. The *deviation from the approximated time usage* was added for two reasons. It is important for a customer to have some indication for the cost of a task, and an adequate task approximation is essential in planning. Another indicator for the external software quality as well as for the maintainability and extensibility of the code is *test coverage. Packages in cycles* and *average component dependency* both indicate the architecture quality. The former represents the quality at a medium scale while the latter indicates the modularization at the level of subsystems. The *class size* represents an application's design quality as well as its maintainability and extensibility. *Method length* and *Cyclomatic complexity* indicate the readability and maintainability of the code and therefore the coding quality, while *compiler warnings* indicate the workmanship.

The first three indicators are pure process metrics, while the others are software metrics. The process quality index (PQI) is based on the values of the three process metrics in a given time interval as well as on the changes of the software metrics in the same interval (indicated by Δ in Fig. 1). This way, the process quality is determined by current values as well as by recent improvements. The software metrics underlie the software quality index (SQI) at a given moment. For example, 20 % of PQI and 35 % of SQI are based on the architecture quality.

The indicators and the indicated properties can also be regarded as aesthetic criteria, e.g. regarding the proportion of the whole and its parts (modularization, design, class and attributes), symmetry (architecture) or wellformedness (class size, method length).

Tool Zoo. We apply several tools to measure the indices and to evaluate the results as well as for the historiography and visualization. The software metrics are measured by two tools: *Sotograph* [Sot] performs automated static analysis, historiography, delta functions, manual identification, and removal of weaknesses, while *EclEmma* (freeware) is used to measure the average test coverage and to identify local deficits at the level of classes and methods. To capture and historicize programming errors, the *BugCollector* (developed by andrena) is used. The central instrument for the integration, condensation, visualization and historiography is the *project logbook* (developed by andrena). For each datapoint, the Sotograph results are automatically integrated.

Integration into Scrum. Scrum [Sch04] is an agile method to manage work in a socially complex environment. Two meetings are defined which serve the continuous improvement of the process. For both meetings, ISIS provides substantiated input that quantifies qualitative changes. Concrete measurements for improvement are determined and their implementation is being supervised.

ISIS and EN ISO 9001. EN ISO 9001, besides measuring the customer satisfaction, requires at least one management report per year. ISIS exceeds these requirements by far. In a monthly report addressed to management and customer, each team presents the quality indices, their interpretation and measurements for improvement, if necessary. A *transparent production* is the result.

User Experience. The introduction of ISIS initiates an intensive discussion of software quality, metrics and production processes and establishes quality awareness. Measurements can be taken immediately. Transparent production fosters confidence by management and customer and leads to steadier production. In all andrena projects the process and software quality was significantly increased by ISIS. An indispensable prerequisite is a learning-oriented no blame culture. Due to the limited number of indicators and extensive automation, it only takes us about one hour per month to determine the quality indices.

3 Conclusions

We presented ISIS, a quality management tool developed and used by andrena. This tool is based on a select number of metrics that are condensed into two main indices representing the process and the software quality. This tool is being applied to all projects at andrena. To our experience it captures the quality of a piece of software at a given moment accurately and follows its development process sensitively. Therefore, software companies that are interested in producing and maintaining high-quality software are likely to benefit from applying ISIS.

We have not yet been able to find adequate indicators for some aspects. For example, the *working productivity*, which is often being measured by ΔLOC per time or function points per time, has not yet been integrated. We generally regard ΔLOC to be inappropriate: When refurbishing existing systems, one generally observes a reduction of LOC due to the removal and cleanup of duplicated code, duplicated logic and unspeakable constructions. In the development of new applications, this metric fosters a tendency to using copy and paste. Another important software metric is *duplicated code*. It is not easily being measured and interpreted. Future activities will concentrate on integrating this metric.

References

[Bec03] Kent Beck. *Extreme Programming.* Addison Wesley, 2003.

[Cas06] Michael Cassel. *ISO 9001 Qualitätsmanagement prozessorientiert umsetzen.* Hanser Fachbuchverlag, 2006.

[CKS06] Mary Beth Chrissis, Mike Konrad, and Sandy Shrum. *CMMI: Guidelines for Process Integration and Product Improvement.* Addison Wesley, 2006.

[Kan02] Stephen H. Kan. *Metrics and Models in Software Quality Engineering.* Addison-Wesley, 2002.

[Loo07] Han Van Loon. *Process Assessment and ISO/IEC 15504: A Reference Book.* Springer, 2007.

[Sch04] Ken Schwaber. *Agile Project Management with Scrum.* Microsoft Press, Redmont, Washington, 2004.

[Sot] Sotograph. http://www.software-tomography.de/html/sotograph.html.

Quality Considerations in SAP Architectures [1]

Wolfgang Theilmann*, Roger Kilian-Kehr*

* SAP Research, CEC Karlsruhe, Vincenz-Prießnitz-Str. 1,
76131 Karlsruhe, Germany
{wolfgang.theilmann, roger.kilian-kehr}@sap,com

Abstract. This paper provides an industrial perspective on the relationship between architectural issues and quality concerns in SAP. We show that quality concerns matter for different kinds of architectures addressing different perspectives on SAP systems such as technology, business logic and business view. Furthermore, the system lifecycle also plays an important role as architectures and kind of quality questions evolve along this lifecycle. We derive 3 key requirements for future system architectures and the modeling of quality characteristics: (1) the ability to deal with underspecified environments, (2) the embedding into the development process and (3) joining of programming models with architectures.

Keywords: SAP architectures, system lifecycle, quality concerns, performance, programming models

1 Introduction

Quality concerns have always played an important role in the design of business applications at SAP [1]. However, current cost pressure in IT industries has lead to an even more important role of quality considerations in the development process and the design of SAP architectures.

The costs of IT systems are mainly determined by 3 factors: cost of engineering, cost of provisioning and cost of operation. Exploiting economies of scale significant further cost reduction can be achieved by increasing the effort in developing solutions of high quality which then allow for provisioning and operation at lower costs. Of course, this only works out if there is a positive tradeoff between the additional effort and the saved provisioning/operation costs.

This paper provides an industrial perspective on the relationship between architectural issues and quality concerns in SAP. This is done in the following steps. Section 2 introduces the solution lifecycle and shows when and by whom quality concerns are dealt with. Section 3 explains some key characteristics of SAP architectures and how these address quality requirements. Section 4 presents a complexity assessment for different quality and sketches specific requirements for

[1] The research leading to these results is partially supported by the European Community's Seventh Framework Programme ([FP7/2001-2013]) under grant agreement n° 216556

efficiency issues. Last, Section 5 concludes with a summary and some key requirements for future system architectures and the modeling of quality characteristics.

2 Solution Lifecycle

Quality concerns need to be dealt with along the complete lifecycle of a solution. The following figure addresses the main phases and steps of this lifecycle at SAP.

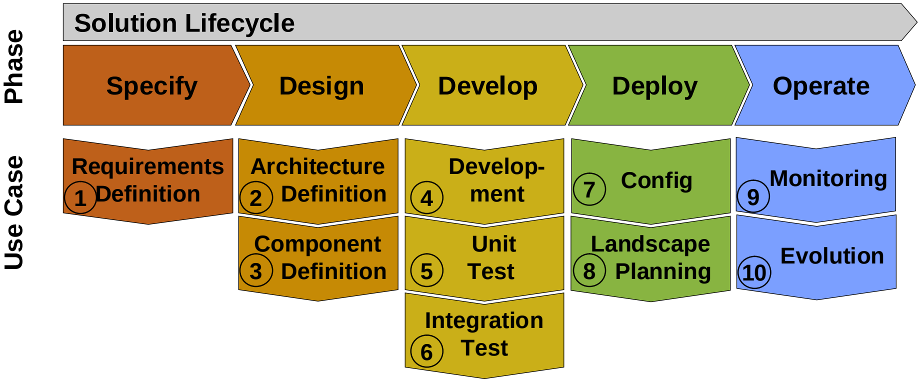


Fig. 1. Basic *solution lifecycle* at SAP (cycles, feedback loops and partial new development not shown in this figure).

Along this lifecycle the following main roles deal with the following issues:

1. Solution Manager provides estimated quality KPIs.
2. System architect specifies assumed characteristics of building blocks.
3. Component Designer relates requested quality with characteristics of underlying components.
4. Developer refines previously assumed quality characteristics,
5. Developer derives actual quality characteristics from unit tests.
6. Tester validates requirements from solution manager.
7. Consultant/Customer specifies actual artifacts to be used (=> first TCO estimate).
8. Consultant/Customer specifies actual landscape (=> final TCO determination).
9. Administrator observes system whether quality targets are met,
10. Consultant/Customer explores impact of planned changes.

Noteworthy, that this lifecycle is a quite simplified abstraction in 2 ways: First, the actual solution lifecycle includes various cycles and feedback loops which allow for an iterative development process that can benefit from early prototypes and tests and feed their results back into earlier stages of the lifecycle. Second, actual development almost never starts from scratch but faces an already existing system which needs to be modified or extended. This obviously limits the freedom but also the uncertainty for all the involved stakeholders.

3 Characteristics of SAP Systems

SAP systems range from midsize to very large size systems, the latter consisting of several hundred million lines of code and deployed on hundreds of distributed compute nodes. In addition to this sheer size, there are a couple of other relevant characteristics which are briefly sketched below.

Several kinds of Architectures. There is no single architecture of an SAP system that covers all perspectives of the relevant stakeholders. Specific architectures are

- the *technology platform* [2], providing generic IT platform and integration functionality,
- the *business process platform*, providing general business logic which can be flexibly used, combined and composed,
- *applications architectures*, representing actual and possibly customer or industry specific solutions,
- *service architectures*, embedding an IT solution into a bigger business service solution, and
- *system landscapes*, describing the actual infrastructure that operates an IT solution and its configuration.

All these architectures provide a different view on different quality concerns for different stakeholders and at different granularity. A proper development process must ensure that information flows correctly between these different perspectives and overall quality concerns can be properly managed and achieved.

System (Development) Paradigms. SAP has adopted the paradigm of service-orientation for developing and providing business functionality. The so-called Enterprise SOA approach [3] goes far beyond regular Web services as Enterprise services feature clear business semantics (they are structured according to a harmonized enterprise model based on business objects, process components, and global data types), quality and stability (they safeguard a stable interface for future versions) and adherence to standards (they are based on open standards such as WSDL and UN/CEFACT CCTS).

Furthermore, a sound model-based development approach has been chosen which provides multiple rich models for both business (integration scenarios, process components) and IT perspective (business logic, integration logic, configuration).

Customer Engagement. Customers are involved in the specification of new SAP solutions already in the very first phases of a solution lifecycle. This co-development serves for creating solutions which eventually meet market needs in terms of functionality but also quality requirements. However, the (quality) requirements and environment of specific customers are largely unknown at design time. This significantly increases the difficulty at design time to predict quality properties and costs of only vaguely known target environments. Consequently, traditional SAP systems require thorough go-live check at a customer site before they can become operational.

Quality Concerns @SAP. Following, we sketch how the most important quality aspects are addressed in SAP.

Scalability is clearly a cross-cutting concern that spans across all kinds of architectures. However, scalability is largely solved by the technology platform which allows for linear scalability of the application server cluster.

Availability is largely achieved in the technology platform by a fault tolerant setup of application server & database.

Responsiveness (response time) is again a cross-cutting concern. There is a global requirement of 1-2s maximal response time for interactive applications. This requirement is broken down via budgeting to architectural layers and components.

Efficiency (resource consumption) is another cross-cutting concern. A sizing formula [4] allows relating usage profile per applications with resource demands. There are currently running efforts to build a sizing repository (for components). However, actual customer requirements for efficiency can be very specific and SMEs in particular are very cost-sensitive. This motivates current research efforts for multi tenancy support, i.e. resource sharing between different customers/tenants.

Extensibility (as part of maintainability) is mainly addressed at the business process platform. Here a dedicated framework part of the architecture assures various degrees of seamless extensibility.

Portability is solved by the technology platform by avoiding usage of any hardware/OS/DB-specific features.

Other quality aspects such as usability and security are mainly addressed by development guidelines and do not directly reflect in any architecture.

4 Complexity & Challenges

The following figure qualitatively summarizes the complexity of the various quality concerns at SAP in terms of architectural complexity (i.e. how many architectures are involved) and process complexity (i.e. how many stakeholders and lifecycle phases are involved).

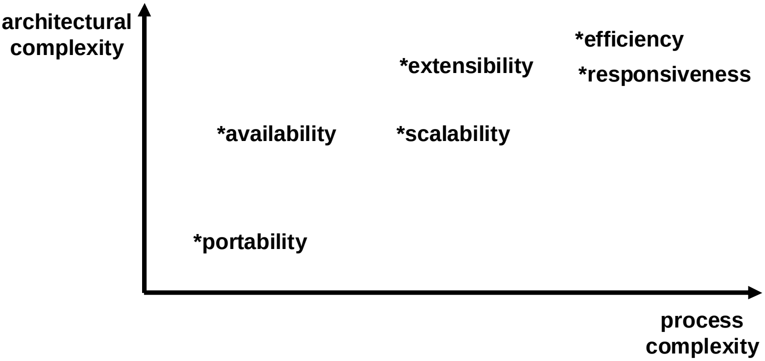


Fig. 2. *Architecural and process complexity* of quality concerns at SAP.

As illustrating example, we sketch some specific challenges for managing efficiency.

First, there is the underspecified environments which means that (a) concrete deployment and infrastructure (hardware, DB, OS) are unknown at design time, (b) customer requirements/behaviour unknown at design time and still underspecified at go-live time, (c) actual control flow is known vaguely (at design time) and slightly better (at testing time – scenario-based testing) again better after business configuration and even better at run-time, (d) component developers are focussed on one architectural layer while non-functional characteristics of lower layers are only vaguely specified and subject to change, (e) the number of configuration & usage variants prevents from exhaustive testing and (f) scenarios of dynamic service composition are even harder to predict.

Second, the various architectures and programming models are just loosely coupled which means that no formal/provable relationship between architecture models and programming artefacts exists. It is currently unclear whether a closer coupling is feasible at all with general purpose programming languages.

Third, technical expertise on non-functional behaviour of artefacts is widely spread and poorly formalized, so it’s hard from an overall perspective to say who knows/does what and when.

5 Conclusions

Quality issues play in increasing role for SAP systems in order to allow for more cost effective provisioning and operation of these. Facing the size and complexity of SAP systems they are extremely hard to properly manage. Future software/system architectures and their associated models could play an important role for better management of these quality issues. In order to realize that the following key requirements need to be solved:

- Specification/prediction of quality aspects must be supported in underspecified environments (e.g. infrastructure, service composition, usage profile unknown). The decoupling of the roles of software and service provider (only latter one knows actual execution environment, customer requirements and service wiring) must be acknowledged. Last, quality characteristics in flexible service composition need better support.
- Adequate quality management requires joining programming models and architectures where abstractions on the one side meet with abstractions on the other.
- Quality management requires deep embedding into the development process with clear specification who knows/does what and when. Adhoc solutions do not scale for large organisations/systems.

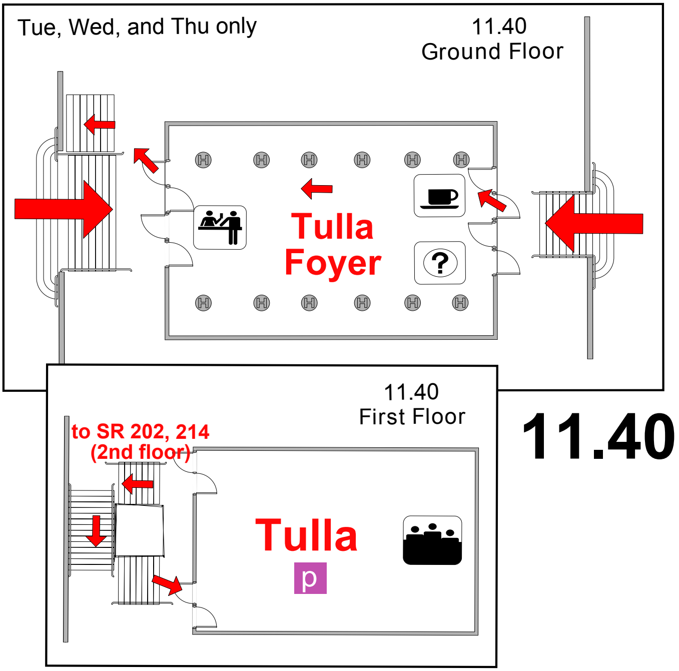
However, all these requirements and possible related measures need a careful assessment of the tradeoff between required additional engineering effort vs. saved provisioning and operation costs. For having a business case a clear return on investment (ROI) needs to be specified.

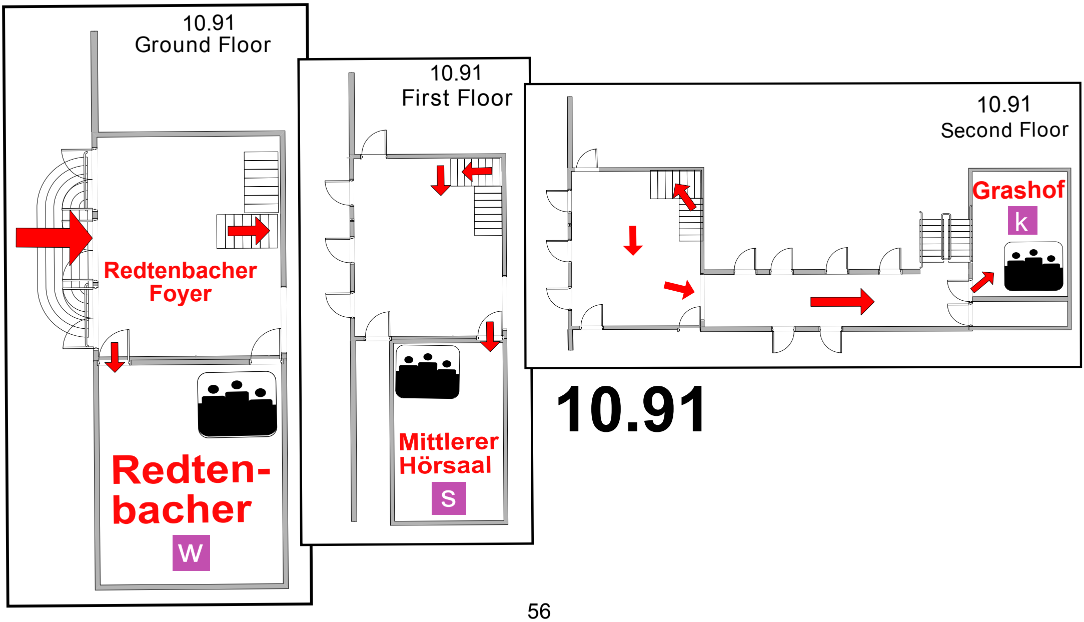
References

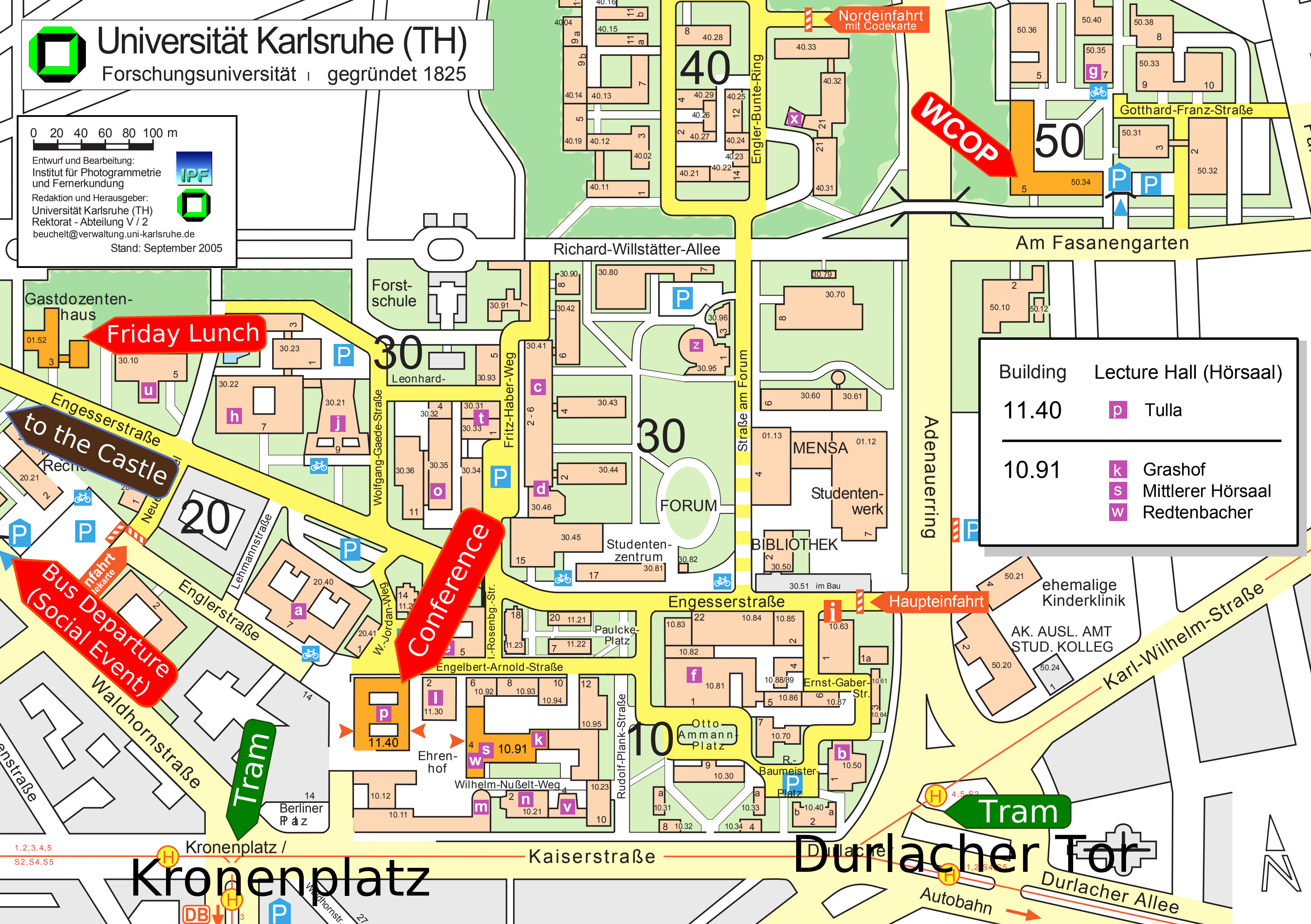
1. SAP Web-Site, http://www.sap.com
2. SAP NetWeaver platform, https://www.sdn.sap.com/irj/sdn/nw-products
3. SAP Enterprise SOA, https://www.sdn.sap.com/irj/sdn/enterprisesoa
4. Susanne Janssen and Ulrich Marquard: Sizing SAP Systems, SAP Press, ISBN 978-1-59229-156-4, http://www.sap-press.com/product.cfm?account=&product=H2904

Room Plans

- Lecture Hall
- Registration
- ? Information
- ☕ Refreshments







Universität Karlsruhe (TH)

Forschungsuniversität | gegründet 1825

0 20 40 60 80 100 m

Entwurf und Bearbeitung:
Institut für Photogrammetrie und Fernerkundung

Redaktion und Herausgeber:
Universität Karlsruhe (TH)
Rektorat - Abteilung V / 2
beuchelt@verwaltung.uni-karlsruhe.de

Stand: September 2005

Building	Lecture Hall (Hörsaal)
11.40	p Tulla
10.91	k Grashof s Mittlerer Hörsaal w Redtenbacher

Platinum Supporters

Silver Supporters

Bronze Supporters

General Supporters