05462 Abstracts Collection
Service Oriented Computing (SOC)
— Dagstuhl Seminar —

Bernd J. Krämer[1], Michael P. Papazoglou[2] and Francisco Cubera[3]

[1] FernUniversität in Hagen, DE
bernd.kraemer@fernuni-hagen.de
[2] Tilburg University, NL
mikep@kub.nl
[3] IBM TJ Watson Research, US
curbera@us.ibm.com

Abstract. From 15.11.05 to 18.11.05, the Dagstuhl Seminar 05462 "Service Oriented Computing (SOC)" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Service-oriented computing, service development life cycle, service composition, service quality, service management, service-oriented architecture, SOA

05462 Executive Summary – Service-Oriented Computing

This seminar attempted to overcome the present fragmentation of research efforts in the area of service-oriented computing by exchanging ideas, trying to find common ground in the form of a common research agenda, and sketch a technical road map capturing the state of the art in SOC research and identifying opportunities for conducting joint research. A brief Dagstuhl Manifesto on Service-Oriented Computing will be published in IEEE Internet early 2006. Starting point of the road map construction process could be the extended service-oriented architecture (xSOA) proposed by Papazoglou and Georgakopoulos in their introduction to the special issue on SOC in CACM 46(10), 2003. The extended SOA suggests a logical separation of basic service capabilities provided by the conventional SOA (for example, building simple applications) from more advanced service functionality (for example, composing services on the fly), and from the management of services (for example, managing service compositions). Starting from the xSOA architecture, the seminar organizers proposed

Dagstuhl Seminar Proceedings 05462
Service Oriented Computing (SOC)
http://drops.dagstuhl.de/opus/volltexte/2006/579

five themes to be jointly worked out in five different breakout groups. The intention was to work together towards a common objective, a white paper or services manifesto describing the state of the art and highlighting open problems and important research topics for the SOC community to work on in the future. The work group topics proposed were:

1. Service Foundations including topics like runtime infrastructure, architectures like the Enterprise Service Bus, modes of service delivery on PCs, palm tops and other hand-held devices, delivery networks, e.g., cable, UMTS, XDSL, or Bluetooth.
2. Composition of Services including QoS-driven composition, SLA (service level agreement) composition, dynamic composition etc.
3. Service Management addressing support for discovery, introspection, security and resources management, typical management functions, measurement, performance indicators, management infrastructure services and toolsets.
4. Service Development Life Cycle including service analysis, design methodologies, implementation techniques, construction and testing, provisioning, deployment, execution and monitoring, and business process modeling tools.
5. Cross-cutting concerns such as quality of service (QoS), semantics, non-functional characteristics, and others.

For each area a renowned expert in the respective field had prepared a warm-up presentation highlighting the state of the art in the respective field, open issues and interesting research topics. Under the direction of a group leader each breakout group produced a summary report and presented it to the audience at large. On the last day the structure of a SOC anthology was outlined and individual chapters were assigned to authors in the audience are proposed experts in the community. All group reports and a draft research reoadmap are accessible through DROPS.

Keywords: Service-oriented computing, seminar organization, extended SOA architecture

Joint work of: Cubera, Francisco Krämer, Bernd J.; Papazoglou, Michael P.

Business Scenario

A business-to-business scenario, which aimed to serve as a common example for the thematic work group sessions, is presented. It describes the activities of an order management process including client, supplier and trusted third-party roles in a supply-chain environment.

Keywords: B2B scenario, supply chain, order management process

Joint work of: Papazoglou, Michael P.

05462 Service-Oriented Computing: A Research Roadmap

This document presents a Services Research Roadmap that launches four pivotal, inherently related, research themes to Service-Oriented Computing (SOC): service foundations, service composition, service management and monitoring and service-oriented engineering. Each theme is introduced briefly from a technology, state of the art and scientific challenges standpoint. From the technology standpoint a comprehensive review of state of the art, standards, and current research activities in each key area is provided. From the state of the art the major open problems and bottlenecks to progress are identified.

During the during seminar each core theme was initially introduced by a leading expert in the field who described the state of the art and highlighting open problems and important research topics for the SOC community to work on in the future. These experts were then asked to coordinate parallel workgroups that were entrusted with an in-depth analysis of the research opportunities and needs in the respective theme. The findings presented in this summary report build on the advice of those panels of experts from industry and academia who participated in this Dagstuhl Seminar and met at other occasions during the past three years, e.g., at the International Conference on Service Oriented Computing (ICSOC, see www.icsoc.org). These experts represent many disciplines including distributed computing, database and information systems, software engineering, computer architectures and middleware and knowledge representation.

Keywords: Service-oriented computing, research road map, service foundations, service composition, service management, service monitoring, service-oriented engineering

Joint work of: Papazoglou, Michael P.; Traverso, Paolo; Dustdar, Schahram; Leymann, Frank; Krämer, Bernd J.

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/524

Towards Business Process Transparency

Marco Aiello (Università di Trento, I)

Service-Oriented Computing (SOC) is an emerging computing paradigm for building distributed information systems in which the concepts of distribution, openness, asynchronous messaging and loose coupling take a leading role. In this context, applications are built out of individual services that expose functionalities by publishing their interfaces into appropriate repositories, abstracting entirely from the underlying implementation. Published interfaces may be searched by other services or users and subsequently be invoked.

The interest in Service-Oriented Computing is a consequence of the shift from a vision of a web based on the presentation of information to a vision of the web as computational infrastructure, where systems and services can interact in order to fulfill users' requests programmatic view. Web Services (WS), the best-known example, are the realization of service-oriented systems based on open standards and infrastructures, extending the XML syntax.

There is a wide landscape of opportunities offered by Service-Oriented Computing: it is possible to create added value information systems by aggregating individual services; it is possible to (a) create inter- and intra-company information systems with relative ease by 'servicizing' existing systems, and (b) to achieve true interoperability and transparency at run-time. Nevertheless, Service-Oriented Computing is far from being a solution for all distributed systems' problems. A number of open key issues makes the research in the field interesting and challenging at the same time. Devising ways to (semi)-automatically compose services at design time or, even better, at run-time, would change the way we look at software engineering, and radically increase the speed of application development. How to discover existing services beyond signature matching and how to consider semantic equivalences among services is an open problem. All considerations of non-functional requirements are paramount for the user's satisfaction when interacting with services and need general and feasible solutions. Finally, there is a growing need to experiment with Service-Oriented Architectures to test the effectiveness and feasibility of the SOC approach in diverse areas such as electronic commerce, government, procurement, logistics, and many others.

If on the one hand, the adoption of the basic Web Service technology is blooming, on the other hand, the Service-Oriented community needs to prove that higher level functionalities are feasible. Service clients and providers would adopt service-oriented solutions more easily if they were enabled with abstract languages in which to express high level requests, requirements, or business rules. It is necessary to transfer the implementation transparency offered by service-oriented architectures to the end user and empower them with *process transparency*. Clients or providers need not know how every step is performed to achieve a given goal, but simply how to express the goal to be achieved or business rules that need to be fulfilled. For instance, having a user expressing his/her needs in the form of a high-level request and abstracting away from any implementation, any issue of availability of a service, or even of which services are necessary to complete the request, is the arrival point of SOC architectures. To achieve this, we need ways to automatically compose services, we need benchmarks to prove feasibility and effectiveness of any proposed approach given the relative youth of the SOC field.

Full Paper:
http://dit.unitn.it/ aiellom/pub.html

Contract Based Testing of Web Services

Michael Averstegge (FernUniversität in Hagen, D)

As Web services become more prevalent the inherent paradigm of loose coupling causes extra effort in testing. Context and enclosing process are not predictable and therefor not testable at design time, e.g. a Web service can be a substitute of any unavailable Web service . Since web services act as black boxes, contracts are the only formal functional specification visible to the client above the syntactic level offered by WSDL specification. In order to enable efficient pre-runtime tests, an overall contract for a dynamically created process must be createable automatically out of the contracts of the atomic Web services. If possible, this can only be done by an additive test layer wrapped around the Web services. Therefore the test is an extra layer acting as test-driver and constraint solver. We currently pursue research on the feasibility of creating an overall contract out of contracts of atomic Web services and testing them on process level.

Keywords: Test, Web service, contract-based testing

ServiceMosaic Project: Modeling, Analysis and Management of Web Services Abstractions

Boualem Benatallah (Univ. of New South Wales, AU)

Although Web services provide abstractions to simplify the integration at lower levels of the interaction stacks (e.g., data syntax and communication protocols), where many of the issues have already been identified or even solved, they have not (yet) contributed to simplify integration at higher abstraction levels (e.g., data/message types and business-level interaction protocols). Generally stated, a business protocol specifies the ordering constraints on the message exchanges, which are allowed by the service in the interactions with other services. We developed a model-driven framework, called ServiceMosaic, for modeling, analysis and management of Web service abstractions.

Our framework builds upon existing protocol models and techniques to provide high level abstractions and operators for service protocols analysis and management. This framework has been implemented in a prototype platform, called ServiceMosaic, as a CASE toolset for modeling, analyzing, and managing service models including business protocols, orchestration, and adapters. The ServiceMosaic platform is developed using Java and J2EE technologies. We used the Eclipse platform. The ServiceMosaic toolset fits in the Eclipse platform as loosely-coupled plug-ins.

Our current work focus is on extending analysis and management techniques for timed protocols, and we will next concentrate on transactional aspects. We are also investigating business protocol discovery techniques to bring the benefits of protocols based interactions to services that do not explicitly model business protocols, or to interactions that involve groups of services. Finally, we plan to explore techniques for cataloging and analyzing previous adapters to improve the process of developing new adapters.

Keywords: Web services, business protocols, adapters, protocol discovery

05462 Summary Report from the Composition Group

Vincenzo D'Andrea (Università di Trento, I)

This report summarizes the activities of the working group in charge of discussing issues related to Service Composition. The group discussed the various aspects related to the composition of services. Composition is one the important aspects in service oriented computing: the general idea is that in building a new service, one can rely upon existing services which provide part of the needed operations. Several issues are related to this topic, from the research on automating the composition process, to the techniques for managing a composition, to the relationship with software (and service) reuse. These and other topics were touched in the two afternoon sessions of the group, with the goal of identifying the open research issues related to composition.

Keywords: Service composition

Joint work of: D'Andrea, Vincenzo; Benatallah, Boualem

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/520

Licensing Web Services

Vincenzo D'Andrea (Università di Trento, I)

Licensing and license models is a fundamental issue in software distribution. By these means, business relationships and strategies are built.

In order to address legal and commercial issues, the explosion of SOC demands a similar attention. THe nature of services [1] and of web services [2] precludes the adoption of a license model as it stands now for software. The dimensions of services that differ from software need to be analysed in order to define a licence model for services. Inter-Organizational boundaries, composition hierarchies, and service evolution are three examples of difficulties in defining such a model. Service licences have to be human radable as well as machine readable and formalisable, in order to allow licence composition.

- [1] the definition of what is a service is postponed

- [2] a service delivered via web protocols

Keywords: Service Licence

Joint work of: D'Andrea, Vincenzo; Gangadharan, G.

05462 Session Summary – "Service Management"

Asit Dan (IBM TJ Watson Research Center - Hawthorne, USA)

The primary focus of the service management track was to reach a common view on the state-of-the-art in service management, and to identify key issues related to service monitoring, configuration and management. The paper puts service management in the context of distributed systems management to derive commonalities and new challenges including monitoring and enforcement of service level agreements and management policies. It sketches a conceptual framework for service management, addresses related standards, and outlines possible trends and unsolved challenges of service management.

Keywords: Service management, service life-cycle management, service monitoring, policy enforcement, service level agreement

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/549

05462 Service Oriented Computing: Service Foundations

Schahram Dustdar (TU Wien, A)

The foundations of Service-Oriented Computing are concerned with the precise definition of a service. This is not only about providing a verbal explanation of what a service is, what can be and what cannot be considered a service, but rather it is about identifying the appropriate service model. Defining what service properties, requirements, and behaviors are relevant and possible for any generic service. Any task based on services has to then take into account the service model, in fact, it is shaped by the service model chosen. Think of service composition. In whichever way one wants to consider composition (static ad design-time, dynamic at execution time, etc.) one has to consider what are the basic blocks to build the composition out of. One has to know which properties and behaviors of the service are available to guarantee certain properties and behaviors of the composition. The report summarizes the results of the two days of brainstorming of this working group.

Keywords: Service Foundations

Joint work of: Dustdar, Schahram; Aiello, Marco

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/528

Semantic Web Services

Andreas Hess (Vrije Universiteit Amsterdam, NL)

Semantic Web Services promise a lot, but are not yet fit for use in the Real World. They do, however, address many of the issues also faced by the SOC community, and there is no need to throw out the baby with the water, so the SOC and SWS communities should talk!

Keywords: Semantic Web services

Towards A Meta-Model for Service Properties

Jens Hündling (Hasso-Plattner-Institut - Potsdam, D)

Service Oriented Computing offers a promising approach for global businesses and integrated, virtual enterprises that achieve common business goals. For realizing an exhaustive Service Oriented Architecture, some basic research is still necessary and a consensus has to be reached about key aspects.

We argue that a key aspect is a common model for enriched service descriptions, especially for all use cases related to service discovery. For automated discovery, these service descriptions need to be specified in a formal, computer-readable way. Additionally, properties of services that are beyond technical interface specifications should be modelled, e.g. by including Quality of Service (QoS) properties.

Keywords: Service Properties, Quality of Services, QoS, non-functional Properties, Service Discovery

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/529

Full Paper:
http://bpt.hpi.uni-potsdam.de/twiki/pub/Public/JensHuendling/Huendling-ModellingProperties_6pages.pdf

See also: Jens Hündling: Modelling Properties of Services. Proceedings of the First European Young Researchers Workshop on Service Oriented Computing April 21-22 - 2005, Leicester , U.K.

Automating Semantic Service Usage

Michael Klein (Universität Karlsruhe, D)

One major challenge in service oriented computing is an automated service binding. This goal can only be achieved with the help of an appropriate service description language that goes beyond a simple interface description.

In our work, we present such a language, the DIANE service description (DSD). It integrates additional information into the description that is necessary to automate the service usage process. First, we integrate unique and structured real world semantics into the description by using layered ontologies. Second, we integrate the configuration semantics into the service description, i.e. we show how the inputs and preconditions influence the result. This is done by using a purely state oriented service description where only state changes are described and and an explicit description of the exchanged messages is omitted. We use variables to provide information how the state can be configured. Third, we integrate preference semantics into request descriptions, i.e. we enable the requestor to integrate all of his preferences of the desired service into the description. This allows us to generate a personal matcher for each request.

Keywords: automated service binding, interface description, ontology

Services in Educational Technology

Bernd J. Krämer (FernUniversität in Hagen, D)

Educational technology has evolved in waves during the last 30 years, producing many innovations that finally ended up on the cemetery of educational technology. But as more and more learning management systems (LMS), learning environments and tools, learning object repositories (LOR), and educational content are put in place and made available on the Internet, it seems that learning technology will sustain and grow, in form or another.

One problem with the current situation in learning technology is a huge lack of interoperability. Student and course data are maintained in incompatible, often proprietary formats and platforms and tools are typically designed as closed solutions, which hinders students to find the best match in learning content and services or move between institutions and take their electronic study portfolio with them. In addition, networking and effective cooperation between educational institutions is prohibited. Learning technology standards aim to overcome this situation may be be too late to have a broad impact on existing solutions. In contrast, web services as a lightweight method to connect software systems over the Internet seem more likely to innovate educational technology by componentizing educational computing infrastructures; they make it easier to overcome the barriers of disparate infrastructures and help to implement the vision of broad access to information and services faster than standards can do.

At FernUniversität in Hagen we just started a few projects aiming to explore the potential of the service-oriented paradigm in higher education. Reliability, trust, security and personalization of services and information are of primary concern in this endeavor. In my own group we are currently investigating new ways of testing web services based on a contract model and improving existing service discovery solution though the incorporation and quality-of-service issues and a related metrics for ranking matches.

Keywords: Educational technology, interoperability, web services, learning technology standards

What does Service-oriented Computing really mean?

Dominik Kuropka (Hasso-Plattner-Institut - Potsdam, D)

If you take a closer look at current service-oriented architectures and their standards like Web-services and compare them to other distributed computing approaches, then you will come to the following conclusion: Technically SOC does not provide any new possibilities or solutions which were not already available or implementable with the old approaches. It would be for example not a problem to implement most services by using for example CORBA. For this reason we have to conclude that the major innovation in SOC is not a technical one, but more a philosophical one. The philosophical innovation is the move from the object-oriented paradigm to a service-oriented one.

Keywords: SOC, OOP

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/522

Web Service Capacity Management

Heiko Ludwig (IBM TJ Watson Research Center - Hawthorne, USA)

Service capacity refers to volume of service provided at a defined level of service. Service Capacity Management has been primarily the province of system management, mostly detached of the WS infrastructure as an issue of a service provider. This detached way of managing capacity works only if one of three conditions is true: (1) The client capacity requirements are known, e.g., because service and clients are in the same organization or management domain. (2) Client behavior changes at a rate that a provider can add or remove capacity in time. (3) Failure to provide service capacity is no issue.

If clients require capacity commitments from service providers, an agreement between client and service must be established. Answering capacity request on the service provider side mainly requires the integration of management functions with the container of the service. Client side capacity management of service capacity, however, is not widely addressed. Clients need functions to estimate requirements of capacity, contracting function to create agreements and aggregate capacity from different providers. Furthermore, acquired capacity must be managed at runtime. This entails a rich field of research for a more powerful, capacity aware service client model.

Keywords: Web services, capacity, management

05462 Session Summary – ”Cross Cutting Concerns”

Heiko Ludwig (IBM TJ Watson Research Center - Hawthorne, USA)

A session on cross cutting concerns of Web services necessarily addresses a multitude of domains, unlike the other ”core” working groups focusing on development and life-cycle, foundations, composition, and management. A cross-cutting issue is a topic that relates to a multiple or all of the core subjects but cannot be address in the context of a single core workgroup.

Keywords: Cross-cutting concerns, semantics, quality of service, definition, business web services, external effects

Joint work of: Ludwig, Heiko; Petrie, Charles

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/526

Service-Oriented Design: The jABC Approach

Tiziana Margaria (Universität Göttingen, D)

Reviewing our 10 years of experience in service engineering for telecommunication systems from the point of view of Service-Oriented Design then and now, we observe that much is common to the two communities. We aim in our current research at establishing a link to the notions used by the service-oriented programming (SO) community.

We are convinced that combined approaches, that blend the flexibility of the current SO-scenario with the rigour and semantic standardization culture of the telecommunication community will dramatically increase the productivity of the development of a large class of software systems. Incremental formalization and automatic verification techniques may be again the key to achieving confidence and reliability for services that interact and interoperate on a large distributed scale.

Keywords: Service-Oriented Design, Telecommunication Services, Service platforms

Joint work of: Margaria, Tiziana; Steffen, Bernhard; Reitenspieß, Manfred

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/521

Operating Guidelines for Services

Peter Massuthe (HU Berlin, D)

In the service-oriented architecture (SOA), we distinguish three roles of service owners: service providers, service requesters, and service brokers.

Each service provider publishes information to the broker about how requesters can interact with its service. Thus, the broker can assign a fitting service provider to a querying requester.

We propose the information published to the broker to be *operating guidelines*. Operating guidelines are essentially communication instructions for the service requester. We present an automata-theoretic approach that is centered around operating guidelines and is capable of implementing all tasks arising in the SOA.

Service Oriented Architectures: Approaches, Technologies and Research Issues

michael P. Papazoglou (Tilburg University, NL)

SOA is an emerging approach that addresses the requirements of loosely coupled, standards-based, and protocol-independent distributed computing. Typically business operations running in an SOA comprise a number of invocations of these different components, often in an event-driven or asynchronous fashion that reflects the underlying business process needs. To build an SOA a highly distributable communications and integration backbone is required. This functionality is provided by the Enterprise Service Bus (ESB) that is an integration platform that utilizes web services standards to support a wide variety of communications patterns over multiple transport protocols and deliver value-added capabilities for SOA applications.

This paper reviews technologies and approaches that unify the principles and concepts of Service Oriented Architecture with those of event-based programming. The paper also focuses on the ESB and describes a range of functions that are designed to offer a manageable, standards-based SOA backbone that extends middleware functionality throughout by connecting heterogeneous components and systems and offers integration services.

Finally, the paper proposes an approach to extend the conventional SOA to cater for essential ESB requirements that include capabilities such as service orchestration, "intelligent" routing, provisioning, integrity and security of message as well as service management. The layers in this extended SOA, in short xSOA, are used to classify research issues and current research activities.

Keywords: Service oriented architecture, asynchronous and event-driven processing, application and service integration, enterprise bus

See also: to appear with VLDB Journal

Extending the Service Oriented Architecture

Michael P. Papazoglou (Tilburg University, NL)

The basic SOA model implements concepts such as service registration, discovery, and load balancing of service requests. However, the basic SOA does not provide facilities for ensuring consistency across the organization, high availability of services, security of non-public services and information, orchestration of multiple services as part of composite applications, or metadata management – all essential requirements for business-quality services. In fact essential SOA applications requirements, suggest that the basic approach be extended to support requirements for business-quality services and advanced functional capabilities such as service orchestration, ”intelligent” routing, provisioning, and service management. It should also guarantee the integrity and security of messages. Such overarching concerns are addressed by the extended SOA (xSOA).

The xSOA is an attempt to streamline, group together and logically structure the functional requirements of complex applications that make use of the service-oriented computing paradigm. The xSOA is a stratified logical service-based architecture that has three planes with the bottom plane utilizing the basic SOA constructs while the service composition and management planes are layered on top of it.

Keywords: Service-oriented computing, services oriented architectures, web services composition, web services management

Full Paper:
http://www.bijonline.com/index.cfm?section=article&aid=159

05462 Summary Report on "Service Design and Development"

Barbara Pernici (Politecnico di Milano, I)

This paper reports on the work group discussions on the theme "Service Design and Development". The sessions took place during the Dagstuhl seminar on Service Oriented Computing held in Schloss Dagstuhl in November 2005. The group discussed, in particular, modeling and methodological issues and different perspectives on the service development life cycle. It also set up a draft research agenda on these issues.

Keywords: Service design, development, integration, models, life cycle

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/525

Limits and Potential of AI Service Planning

Charles Petrie (Stanford University, USA)

Abstract: AI Service Planning is often critiqued for the wrong reasons, based upon specific technologies. On the other hand, there are fundmental limits to this technique for generating web service-based processes.

The traditional process and planning communities tend to look at service composition differently. The former tend to start with process definitions, focusing on a protocol of message exchange, iterating over the results, and analysing its properties until the result is some a new reusable process, which may be characterized as a composite service. The planning community tends to focus on achieving a specific goal by integrating some set of web services, resulting in a one-time process instance. Desirable properties result from constraints on process generation. Proper message exchange is a by-product of planning.

One computational issue is how far service planning can be extended, because general process synthesis is a kind of program synthesis. A practical issue is whether all of the information used in traditional process synthesis can be represented so that planning could generate the same processes. Finally, it is not clear that synthesizing general reusuable processes is always necessary: ultimately, reusuable workflows may in many cases be replaced by one-time process synthesis.

Keywords: AI Planning Service Composition Process

In the era of globalization business processes span across enterprises...

Alexander Rausch (Hasso-Plattner-Institut - Potsdam, D)

In the era of globalization business processes span across enterprises. IT has to face the challenge of mergers, outsourcing and temporary joint ventures. IT projects must be ready for change during all phases of the life cycle.

Modelling a business process requires an in-depth knowledge about the business context, the interaction between business partners and the dependencies to other business processes. From a business point of view it does not matter whether a certain element of a process chained is deployed in the one or other enterprise. IT must provide a platform to easily move elements of the process chain between enterprises.

The designer of a business process requires tools for assembling a network of process elements. He breaks down the overall business process to a level where he can map the resulting elements to existing services. On the other hand the service provider must have an in-depth understanding of the underlying functionality. Either he has implemented this functionality from scratch or he has composed this functionality by assembling existing underlying services.

Two different development paradigms come into play. Modelling a business process requires a top-down approach where business knowledge is crucial and technical knowledge is neglectable. Service enabling requires an in-depth analysis of existing applications and the subsequent renovation. The latter requires skills about a goal-oriented componentization of software.

The life cycle imposes additional challenges. Traditional version control does not work in a loosely coupled world of orchestrated services. High availability is necessary. For each functionality several services must be available and exchangeable. Perhaps the marketplaces of the last internet hype will come up again.

Decentralized Service Discovery in BRICKS

Thomas Risse (Fraunhofer Institut - Darmstadt, D)

Service registration and discovery play an important role in service-oriented architectures. Many system components are available only through a web-service interface, and we would like to achieve dynamic bindings of components during run-time. Therefore, service registration and discovery mechanisms help in finding an appropriate component/service that provides needed functionalities.

Within the BRICKS project (IST 507457) the service registration and discovery component follows the standards of WSDL and UDDI. Since one aim of the BRICKS project is to build a decentralized architecture, we cannot simply take an existing centralized UDDI implementation and deploy it in the system. Hence we keep the standard interface, and build a decentralized UDDI repository on top of our decentralized XML storage. By doing this we avoid that the UDDI repository can be a single point of failure in the system. Furthermore, as out decentralized XML storage is self-organizing we can reduce the maintenance costs of the infrastructure.

Keywords: Service Discovery, UDDI, Peer-to-Peer, decentralized XML storage

A model-based approach for the development of automotive software based on services and abstraction levels

Sabine Rittmann (TU München, D)

The emerging interoperability among software systems, especially across operational boundaries, results in challenging problems. The heterogenity of distributed systems and the complex interactions between those have to be dealt with. A promising approach to handle this intricacy is the upcoming service-oriented paradigm. Here, a service is a piece of functionality, like the opening of the power windows of a car, for example. Therefore, the focus lies on the system behavior, and not on the system structure (as it is the case with the traditional component based approach).

Most of the work on service-orientation defines a service merely by a syntactic list of procedures which are called by a client (or more general: by a requester). However, this is not sufficient. The problem is that complex interactions and the interplay of functions, which is necessary to provide a thorough understanding of the system, are lost sight of. Local views on pieces of the (distributed) system functionality do not allow for an overview of the various relationships between system entities. This is especially true for domains where systems are characterized by a high amount of interactions between functional modules (so-called multi-functional systems), like the automotive domain.

We work on an innovative model-based design approach for the development of automotive software. In contrast to low-level modeling approaňches (such as Matlab/Simulink or ASCET-SD, which mainly focus on technical aspects of the system), high-level modeling concepts are introduced to represent HW-/SW-architectures within a set of consecutive abstraction levels. The system of abstraction levels supports the inheritance of model information from abstract levels down to concrete levels and the refinement of this information at each level. Thus the gap between (informal) requirements and the implementation is reduced.

In this set of abstraction levels, services are used as a means to formalize functional requirements. The system is first specified by a merely behavioural view abstracting from structural issues (like software components and hardware details). As a result, the interplay of functions can be understood more easily and functional inconsistencies can be detected quite early in the development process. Additionally, since the higher levels abstract form technical details, reuse of models will be possible in a very easy way. The presented approach is currently developed within the project ”mobilSoft” (funded by the Bavarian Government under grant number IuK 188/001).

A list of publications on this work can be found under:
http://www4.in.tum.de/ rittmann/public.html.

Keywords: Services, service-orientation, model-based development, automotive domain, abstraction levels

eScience Grid Services

Heinz W. Schmidt (Monash University, AU)

Over the past decade eScience infrastructures in networks, computing and software have matured and brought grids to the science communities. Grids involve distant collaboration, virtual presence, massive data stores, high-performance computing facilities, large scientific instruments and the accompanying software platforms and middlewares dealing with modelling, experiment, simulation and legacy data and packages.

In analogy to the power grid, eScience grids promise 'plug-in' science anywhere anytime: real-time access and control to global resources for the individual scientist as well as orchestrated coordination of science capabilities, teams and flagship projects.

However access to the vast resources encounters road blocks: software is not scalable, there are few paths from experiments run an a local workstation with small data sets to those consuming tera or soon petabytes of event data from synchrotrons or geoscience datastores and running simulations on several clusters around the world.

This presentation looks at some of the characteristics of eScience grids and grid services scenarios. It then summarises some of the key challenges and problems that eScience grid services architectures will have to overcome and sketches a roadmap for projects in this space.

Keywords: eScience, grid, service, quality, performance, availability, service oriented architecture

Mindmap of Day 2 Discussions

Heinz W. Schmidt (Monash University, AU)

Freemind map summarising presentations (5 themes) and workout groups - obviously a personal view ;) This is temporary and will be replaced eventually by something more consolidated.

(freemind is a small-footprint opensource mind mapping package).

Keywords: Mindmap, service orientation

Full Paper:
http://www.csse.monash.edu.au/ hws/soa-mm/SO.html

What a service is (Clickable mind map)

Heinz W. Schmidt (Monash University, AU)

My view of service software as something overlapping component software but also significantly different and with characteristics far beyond components - an attempt to integrate some of what was said at the workshop.

Apparently there is no agreement what a service is. Perhaps those of us in the components and OSA community should offer a definition in the upcoming book? My view of components is well documented in the literature and reflected here. Some of the aspects we need for example in predictable assembly for engineering components (in software-intensive HW/SW systems) go beyond best-practice industrial component systems and, I believe, have parallels in the OSA world. Examples include rich definitions, process/protocol orientation, QoS inclusion in contracts etc.

Keywords: Service, component, interface, contract, QoS

Full Paper:
http://www.csse.monash.edu.au/ hws/soa-mm/service.html

Business-driven SOA

Stefan Tai (IBM TJ Watson Research Center - Hawthorne, USA)

Web services as a technology platform and Web services-based SOA as an architectural style have reached a level of maturity that can be considered "good enough" for many application integration problems, including enterprise-scale distributed systems. However, measuring and evaluating the business impact of Web services-based SOA (when migrating to SOA or after migration) remains unaddressed; yet, the business value and propositions are the critical driving factors for service-oriented computing. We argue that in addition to the current Web services stack non-technical aspects such as labor, organization, legal, governance and new means to address these – cross-cutting to foundations/infrastructure, composition, management and lifecycle – must be considered to constitute a notion of "business service", which can be used to better create and capture the business impact that Web services-based SOA have for organizations.

Challenges of Service Engineering in Virtual Organisations

Christian Zirpins (Universität Hamburg, D)

In the multi-disciplinary context of distributed information systems technology for virtual organisations there is a clear need to target combined aspects of software engineering methodology and distributed middleware technology. In virtual organisations, business processes are decomposed w.r.t. core competencies of its partner organisations.

These assets are continually re-integrated w.r.t. changing requirements of customers and markets. Each such constellation implies a specific interaction procedure between the partners that has to be implemented by an operative process-chain. The more dynamic, the more the virtual organisation relies on integrated ICT.

Service-oriented Grid architectures promise effective means to share functions and processes. Yet, gaps remain between technology and its application: the abstractions (e.g. Web Services) are too low-level to be efficiently applied by business engineers and software engineering methodology to develop mission-critical systems on top is usually missing. Hence, two research goals emerge: (i) high-level abstractions for organisational interaction procedures that enables efficient realisation of operational process-chains and (ii) software engineering methodology to develop information systems that enforce and control organisational constellations.

Keywords: Information Systems and Software Engineering Methodology for Networked Organisations, Service-Oriented Architecture and Grid Computing