

06291 Abstracts Collection
**The Role of Business Processes in Service
Oriented Architectures**
— Dagstuhl Seminar —

Frank Leymann¹, Wolfgang Reisig², Satish R. Thatte³ and Wil van der Aalst⁴

¹ Universität Stuttgart, Institut für Architektur von Anwendungssystemen
70569 Stuttgart, Germany.

`frank.leymann@iaas.uni-stuttgart.de`

² Humboldt-Universität zu Berlin, Institut für Informatik
Unter den Linden 6, 10099 Berlin, Germany.

`reisig@informatik.hu-berlin.de`

³ Microsoft Research, Redmond, WA 98052, USA.

`satisht@microsoft.com`

⁴ Eindhoven University of Technology

P.O.Box 513, NL-5600 MB, Eindhoven, The Netherlands.

`w.m.p.v.d.aalst@tue.nl`

Abstract. The Dagstuhl seminar on *The Role of Business Processes in Service Oriented Architectures* (Seminar 06291) took place in July 2006 (16.07.2006-21.07.2006 to be precise). The seminar was attended by more than 40 experts from both academia and industry. Unlike most Dagstuhl seminars there was a high participation from industry (in particular from organizations developing software, e.g., IBM, SAP, Microsoft, Google, etc.). The focal point of the seminar was the marriage of business processes and service oriented architectures. This was reflected by the topics selected by the participants and their background.

Executive Summary

1 Motivation

More and more, applications are no longer built from scratch but by integrating pieces of software that have been built independently from each other. As a consequence, the various pieces of an application must be loosely coupled. Service oriented architectures (SOA) provide a general prescription and guidelines of how to loosely couple pieces of application functionality. Web services are a concrete instantiation of a service oriented architecture. Clearly, business processes are essential when aggregating loosely coupled functions into new application functionality. For the combination of business process technology and Web service technology the terms choreography or orchestration has been coined. These technologies are expected to become the foundational layer for tomorrow's information systems and are influencing already many application areas like Enterprise Application Integration, Software Engineering, Systems Management, Data provisioning, BPI, B2B – to name but just a few.

Software vendors today deliver products for modeling and executing choreographies. Research in this area is currently done scattered across different disciplines. The major goal of the seminar is to bring representatives from the different communities (from research, software vendors and users, of business processes and SOA) together for a first stocktaking, a joint in-depth understanding of the issues, to identify and prioritize the main research items, identify standardization needs, and to discuss demanding questions and open problems in detail. The areas to discuss include:

- Modeling Languages and Techniques for Business Processes
- Composition and refinement methods for Business Process Models
- Matching/searching process models
- Processes as constraints/semantics for Web services
- Wiring processes together (i.e., relation between BPEL and WS-CDL)
- Relation between BPEL abstract and executable processes
- Analysis of BPEL specifications
- Different transaction models in BPEL
- Business activity monitoring and process mining in the context of SOA
- Business Processes in Grid Architectures

The seminar clearly improved the understanding of the field of “Business Processes in SOA”. Furthermore, new collaborations between the different communities were triggered by this event.

2 Presentations

The following presentation were given during the seminar:



Fig. 1. Participants of the Dagstuhl seminar on “The Role of Business Processes in Service Oriented Architectures”

1. Mathias Weske, Hasso-Plattner-Institut, Potsdam
Towards Services-based Process Platforms
2. Peter Dadam, Universität Ulm
ADEPT and AristaFlow - Towards a New Dimension for Process-Aware Information Systems
3. Karsten Wolf, Universität Rostock
Controllability: A Soundness Criterion for Services
4. Kohei Honda, Queen Mary College, London
WS-CDL and Pi-Calculus
5. Jorge Cardoso, University of Madeira, Funchal
Business Process Complexity Analysis
6. Dieter König, IBM, Boeblingen, Germany
Service Composition: BPEL and SCA
7. Dimka Karastoyanova, U of Stuttgart, Germany
Semantics-Based BPEL Flexibility
8. Gerhard Pfau, IBM, Boeblingen, Germany
Management of Human Tasks
9. Satish Thatte, Microsoft, Redmond, USA
Windows Workflow Foundation: Overview and Role in SW-Stack

10. Ivana Trickovic, Alan Rickayzen, SAP, Walldorf, Germany
From Enterprise Services to Composite Applications
11. Chris Bussler, Cisco, Palo Alto, USA
SOA and BPM Technologies as Enterprise-class Computing Architecture Building Blocks
12. Sanjiva Weerawarana, WSO2, Sri Lanka
Revisiting Service Composition: Server-side Mashups
13. Oliver Guenther, Humboldt University Berlin, Germany
Price Formation for Composite Web Services
14. Guadalupe Ortiz Bellot, Centro Univ. de Merida, Spain
Aspect-Oriented Techniques for Web Services: A Model Driven Approach
15. Egon Börger, University of Pisa
An architecture for web service mediation and discovery
16. Gregor Hohpe, Google, San Francisco
Conversation Patterns
17. Carolin Letz, University of Münster, Germany
Web Service Matching - A Relational Approach
18. Kees van Hee, TU Eindhoven
Relationships between services, components and workflows
19. Peter Massuthe, HU Berlin
Operating Guidelines for Services
20. Jussi Vanhatalo, IBM Research, Zürich
Techniques for Business-Driven Development
21. Peter Dadam, Ulm
ADEPT tool demo
22. Bernhard Steffen, Universität Dortmund
Service-Oriented Design: The jABC Approach
23. Tiziana Margaria, Universität Potsdam
Application of the jABC Approach
24. Uwe Zdun, TU Wien
Patterns in SOA
25. Ekkart Kindler, Universität Paderborn
AMFIBIA and SOA
26. Jörg Desel, KU Eichstätt-Ingolstadt
Petri nets and SOA
27. Matthias Kloppmann, IBM
BPEL4people standard
28. Niels Lohmann, HU Berlin
Analyzing Interacting BPEL Processes: A Tool Demo
29. Ekkart Kindler, Universität Paderborn
AMFIBIA tool demo

3 Workshops and Panel

A substantial part of the week was reserved for interaction other than giving talks and discussions based on these talks. In total four half-day workshops and

one panel discussion were organized. The panel discussion was chaired by Satish Thatte and had the title “On the relevance and practicality of process modeling”.

The four workshops were:

- Workshop A: “Patterns” (Gregor Hohpe)
- Workshop B: “Components” (Kees van Hee and Dieter König)
- Workshop C: “What can Theory do for Practice and what does Practice need from Theory” (Jörg Desel and Wolfgang Reisig)
- Workshop D: “Process Mining/Monitoring Processes and Services” (Wil van der Aalst)

Each of the half-day workshops led to interesting insights and fruitful discussions. The results of the workshops are enclosed in the proceedings.

4 Conclusion

As indicated above, the seminar clearly improved the understanding of the field of “business processes in SOA”. Furthermore, new collaborations between the different communities were triggered by this event. Several authors contributed to the proceedings of this seminar. Moreover, there will be a special issue of International Journal of Business Process Integration and Management (IJBPIIM) based on this seminar.

Given the success of the seminar the organizers plan to organize a new seminar in one or two years.

Workshops and Panel Discussion

06291 Workshop A: Conversation Patterns

In a service-oriented architecture systems communicate by exchanging messages. Message passing provides for robust and loosely coupled interaction but it also provides less structure than traditional RPC models, which are based on a fairly rigid request-response interaction style. Instead, messages exchanged over time between a set of parties can form a multitude of conversations. An expressive contract between communicating parties should define a coordination protocol that describes which conversations are legal. Such a protocol can be expressed in different ways, for example through choreography or public endpoint process definitions. The purpose of conversations patterns is to document common forms of conversations in the design pattern format, highlighting design trade-offs and popular implementations.

As part of the Dagstuhl Seminar 06291 on The Role of Business Processes in Service Oriented Architectures in July 2006 we held a Workshop on Conversation Patterns. In this paper, we report on the results of this workshop.

Keywords: Patterns, Conversation, Orchestration, Choreography

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/828>

Workshop B: Component Models

Industry view: Dieter König summarized the main objectives of the component model defined by the Service Component Architecture (SCA⁵, also part of a presentation). SCA provides a programming model for building applications and systems based on a Service Oriented Architecture, which includes a model for the assembly of services, both tightly and loosely coupled, and a model for applying infrastructure capabilities to services and to service interactions, including security and transactions. The SCA model aims to encompass a wide range of technologies for service components and for the access methods which are used to connect them. For components, this includes different programming languages, as well as frameworks and environments commonly used with those languages. For access methods, SCA compositions allow for the use of various communication and service access technologies that are in common use, including Web services, Messaging systems and Remote Procedure Call.

Academia View: Kees van Hee summarized the main goals of a component model (also part of a presentation). Complex systems should be verified. Process models are important for this task; however, it needs to be defined how should

⁵ <http://www.osoa.org/display/Main/Home>

they be integrated into a component model. Components deliver services to their clients and therefore have a role as service provider. At the same time, they use services offered by other components, so they have often the role of service client. The overall system is a network of cooperating components, and can be considered as components themselves. Components may reside anywhere. Correctness criteria include at least the verification of proper completion (services should always be able to terminate with an end event, and they should not leave any unfinished work) and invariants (constraints on case and base objects should be valid as soon as a service has finished). Other properties that are important to be verified are those expressible in some temporal logic. A wish list for component model verification includes patterns for the generic components to be combined with application specific models to verify the complete behavior, which need to be verified in isolation and can be combined with application specific components using construction rules that guarantee correctness (the correctness-by-construction principle). Moreover, state space checking should be combined by reduction techniques. Symbolic model checking is needed. The answers of verification should be translated into the original frameworks of the engineers.

Workshop Discussion: the discussion was started using the following questions:

1. Do we need a component model at all?
Yes, it is the only way to master the complexity of large systems
2. If yes, where do we need it?
Business application composition; see scope definition
3. What is a component and what should be the right granularity?
See below; there are different levels of granularity
4. What should be covered or left out in a component model?
In: see below; left out: the runtime infrastructure
5. How can it be applied in practice, e.g. with respect to the development lifecycle?
Ran out of time before discussing this question
6. What does a composition model look like?
See below

Components should be coarse-grained service components, running in a back end system. Out of scope are e.g. UI components, OS, Bus, and persistence considerations. Components may have memory (stateful component), and they may be (part of) an installable unit.

The definition of components should include publicly visible aspects as the clearly specified function or service, two interfaces (message types and protocols, one offered to clients, and one required from suppliers. Externally observable behavior can be described by e.g. abstract BPEL or Petri Nets, and should include the client's and the supplier's perspectives. Quality of service (QoS) policies and/or service level agreements for aspects security, transaction, reliability, etc. should be included. A protocol validation of the implementation is desirable. The component implementation (BPEL, C#, ...) should be hidden.

Important features of a SOA for composing components include component discovery (static or dynamic binding), synchronous and asynchronous interactions between components, and connection/wiring between components. For the latter, note that the runtime environment itself is not part of the component model – it implements the connections and ensures the QoS specifications. Finally, component lifecycle aspects like version management (optional) and component configuration should be addressed.

Joint work of: Kees van Hee, Dieter König

Workshop C: What can Theory do for Practice and what does Practice need from Theory

This workshop started with two short talks.

First Wolfgang Reisig argued that modelling and models should play a more important role in the development of web services. Instead of summarizing his presentation, I will combine it with the results of the subsequent discussion.

A model is an intended abstraction from reality. So the quality of a model does not depend on its expressiveness but rather on its ability to precisely represent those aspects that the modeler intended to formalize. Bad formalisms force the modeler to be more specific or to be more detailed than necessary. For example, it should be possible to require some property in a declarative style rather than to provide some (arbitrary) imperative statements in a programming language which establishes the property.

Another very important criterion of modelling formalisms is the purpose of a model. The primary purposes are documentation of ideas, communication between humans and verification w.r.t. some specification. Obviously, the latter needs a formal semantics of the language. A model might *prescribe* a system which has to be constructed, it might *describe* an existing system or it might be used to *predict* properties of the system, which can be qualitative or quantitative properties.

As a consequence of the former aspects, a good formalism should provide a single symbol (graphical symbol or name) for a single concept. This implies that the same concepts should be described with the same symbol, thus supporting understandability, but it also implies that a good formalism supports important concepts immediately and not via complex by-passes (e.g., a loop should be represented directly as a loop and not as a special conditional jump).

Concentrating on models for system design, it is essential to distinguish between validation (of the model and of its specification), i.e. the informal proof that a model faithfully describes the intended system on the chosen level of abstraction, and verification, i.e. the formal proof that the model satisfies the specification. Further, models and their formalisms have to be integrated in a methodology in such a way that refinement, i.e. change of abstraction level, is supported – by a single language or by sets of languages.

Finally, in practice one can observe that model based system development often ends with a model and a system. Thereafter, only the system is changed but not the model. Instead, it would be essential to keep the model up-to-date through the entire system live cycle. In other words, maintenance of software has to start with maintenance of models.

Most of the above holds for model based system development in general. There seems to be a tendency that previous mistakes are repeated for the development of service oriented architecture.

Therefore it is important to realize that the matter is not different for these architectures but rather that some additional specifics have to be considered.

Another presentation was given by Karsten Wolf. He presented previous and future research activities of his group and of a research consortium formed by his group, people from Eindhoven (Kees von Hee's group) and people from Berlin (Wolfgang Reisig's group). Emphasis of this presentation was on properties of Petri nets modelling open processes that are to be composed with the environment. There was no time to discuss these topics in detail.

Finally, the representatives from industry formulated their needs and their expectations from academia and vice versa. It turned out that in both directions there is no good match between the expectations from a side and the offered material / research results etc. from the other side. The industry is particularly interested in "prediction" via models. This holds for "risks", which turn out to be missing specifications in most cases (i.e., situations, where simulation discovers bad behavior which was not expected when writing the specification). In this sense, methods and models are requested for validation purposes. Moreover, models can be used to predict quantitative behavioral parameters such as throughput rates. In this sense, models are a means for optimization.

Industry is not that interested in formal results offered by academia – these do not solve the actual problems in the daily practice. Nevertheless, industry needs the universities for the education of their future employees and acknowledges a sound knowledge of the students in the application of formal methods. Therefore, indirectly new research results find their way to real applications.

In turn, industrial applications could support research activities in many ways. The most important aspect is that good scientific work comes together with a proof of relevance, i.e., a real life example or a case study which preferably solves a problem from industry. Often, real problems from industry also guide researchers in choosing their research direction.

It is also worth mentioning in this context that a considerable part of the research (theoretical and practical) is directly or indirectly financed by industry. Therefore, researchers are well motivated to keep contact to industry and to receive appropriate attention. Finally, money for research also comes from the state.

There is a tendency to support application oriented research much more than theory, although there is a common understanding that research in theory and in application strongly depend on each other.

Since the state representatives pay a lot of attention to industry, industry is asked to express solidarity with research in theory and in application. Good research results and well educated employees are a prerequisite for future economical success.

Joint work of: Jörg Desel, Wolfgang Reisig

Workshop D: Process Mining, Monitoring Processes and Services

In a service-oriented architecture, but also in classical enterprise systems, a variety of events (e.g., messages being sent and received) are being logged. This information can be used for *process mining* purposes, i.e., based on some event log it is possible to *discover* processes or to *check conformance*. The goal of process discovery is to build models without a-priori knowledge, i.e., based on sequences of events one can look for the presence or absence of certain patterns and deduce some process model from it. For conformance checking there has to be an initial model. One can think of this model as a “contract” or “specification” and it is interesting to see whether the parties involved stick to this model. Using conformance checking it is possible to quantify the fit (fewer deviations result in a better fit) and to locate “problem areas” where a lot of deviations take place.

Joint work of: van der Aalst, Wil

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/834>

Panel Discussion: On the Relevance and Practicality of Process Modeling

Following up on talks by Satish Thatte and Sanjiva Weerawarana which raised questions about the scope of application for formalized workflow and modeled processes, there was a post-dinner panel discussion on 17.07.06 titled “On the Relevance and Practicality of Process Modeling”, with Wil van der Aalst, Peter Dadam, Satish Thatte, and Sanjiva Weerawarana as the panelists. The overall point was to get the group to step outside the comfort zone of process modeling believers and discuss whether we are focusing on the right formalisms from a practical perspective and whether we are clear about the scenarios where modeled workflow is actually useful. Specifically, the discussion focused on the following two questions:

1. Why are data-driven (as opposed to control-flow focused) models not more prominent in business process modeling (both academic and industry). More generally, should we start with models that are inherently flexible and able to cope more easily with changes both at the instance level and at the model or type level rather than starting with (control flow) models that are hard to change?
2. What application areas are the right ones to model as formal processes? What about processes for handling exceptions? Are they amenable to modeling?

There was lots of lively debate about the first topic. Several points emerged.

One obvious candidate for flexible data driven systems is event-driven rule based systems typically using event-condition-action (ECA) rules. The events may be changes to the (data) state of systems of record or other kinds of business events such as signals from RFID devices or the arrival of B2B messages. Derivative events such as those inferred by complex-event-processing (CEP) systems monitoring real-time event streams are also relevant to this paradigm. Frank Leymann and others expressed concern about the lack of scalability and predictability of this approach as the number of rules grows. Satish Thatte pointed out that there are working systems such as the home grown system at the Nationwide Housing Society in the UK which uses a data driven ECA model for a very large and business critical mortgage processing application. A key issue here is whether the data that is used to drive the ECA rules is data owned by the process instance or by underlying line-of-business applications. Alan Rickayzen pointed out that duplicating application data into processes is not acceptable and data-driven models must account for this constraint. Wil van del Aalst pointed out that there are existing systems like FLOWer which do provide flexible data-driven models and these are in successful use in business environments in the Netherlands and other locations, but for some reason the major vendors and to some degree even the academic world have not paid sufficient attention to this promising technology. Peter Dadam expressed the view that control-driven models are in fact the right paradigm but we did not actually discuss the reasons. Wil and Bernhard Steffen also pointed out that although modeled processes have the image of being regimented it is in fact incorrect to assume that modeling is equivalent to regimentation since constraint based models among others allow predictability without regimentation by avoiding over-specification.

The discussion on relevant application areas for modeled processes surfaced a variety of very different views among the participants. Johannes Klein expressed the opinion that in the real world workflow is peripheral to e-mail and ERP systems as a mode of driving business processes. Bernhard Steffen emphasized the importance of predictability of process flow and results in situations such as Ikea's warehouse workflow. Satish Thatte pointed out that in most cases it is in fact regimented work behavior such as that of warehouse workers that provides the best examples of successful workflow applications.

Joint work of: Satish Thatte

Talks, Demos and Other Contributions

SOA and BPM Technologies as Enterprise-class Computing Architecture Building Blocks

Christoph Bussler (Cisco Systems - San Jose, USA)

SOA (Service-Oriented Architecture) and its surrounding technologies are the currently available silver bullets in the computing architecture space. While SOA provides important architectural abstractions, business process management (BPM) is necessarily a key component for implementing business process functionality for large-scale process-oriented enterprises.

Processes are not simply linear chains of events in enterprises. There exist concurrent actions on ongoing processes that mutually influence each other. These concurrent actions are processes themselves and they have to maintain consistent process and data states. For example, the process ‘change purchase order’ is a process that influences the process ‘submit purchase order’ and both have to cooperate while being consistent. In addition, processes have to implement variants based on specific business domain semantics expressed in entities like order types, order or selling regions, or routes to market. One order process can cover all these various domain semantics aspects, but it has to clearly provide the correct behavior for the domain semantics. Alternatively there could be a process each whereby all processes share common parts as subprocesses. Orthogonal to the pure processing logic that ensures that the process progresses there are policies that processes have to obey during execution. Policies can be specific variations of the process logic based on the selling region, customer type or order volume.

In summary, processes are multidimensional behavioral specifications that require dedicated system support in order to execute consistently and correctly.

From an enterprise architecture perspective processes are only one building block in addition to services and business rules that implement the overall business logic. These building blocks have to be ‘isolated’ from external computation by dedicated externalization interfaces that follow a canonical data model to ensure precise semantics and to allow keeping agreed upon service levels. When the enterprise architecture requires the use of external services a dedicated incorporation mechanism has to be put in place that enforces a uniform model of service invocation and access of external services.

The purchase order process at Cisco Systems is a great source of insights and requirements that pose requirements to the process and SOA community in order to improve not only the non-functional but also the functional capabilities. Based on these requirements an enterprise architecture is developed that follows the idea of a process-oriented architecture where the process is in the center of the architecture and services and business rules are enablers of processes as well as complementary architecture components.

Keywords: BPM, SOA

An architecture for web service mediation and discovery

Egon Börger (Università di Pisa, I)

In this talk we define a high-level model to mathematically capture the semantical meaning of abstract Virtual Providers (VP), their instantiation and their composition into rich mediator structures.

We will show how this model can be successfully applied to two application scenarios, Web Service protocol mediation and Semantic Web Service Discovery.

Joint work with M. Altenhofen, A. Friesen, J. Lemcke from SAP Research Germany

Keywords: Virtual Provider, Mediator, Protocol Mediation, Semantic Web Service Discovery

Approaches to Compute Workflow Complexity

Jorge Cardoso (University of Madeira - Funchal, P)

During the last 20 years, complexity has been an interesting topic that has been investigated in many fields of science, such as biology, neurology, software engineering, chemistry, psychology, and economy. A survey of the various approaches to understand complexity has lead sometimes to a measurable quantity with a rigorous but narrow definition and other times as merely an ad hoc label. In this paper we investigate the complexity concept to avoid a vague use of the term ‘complexity’ in workflow designs. We present several complexity metrics that have been used for a number of years in adjacent fields of science and explain how they can be adapted and use to evaluate the complexity of workflows.

Keywords: Workflow, Complexity, Business Processes, Reengineering

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/821>

ADEPT & AristaFlow - Towards a New Dimension for Process-Aware Information Systems

Peter Dadam (Universität Ulm, D)

The target of the ADEPT project is to develop the fundamentals for a WF technology which makes process-aware applications easy to implement and which is much more flexible than today’s systems.

Very challenging in this context is to achieve this in an efficient manner and without violating consistency and robustness.

The AristaFlow project complements these activities by designing and implementing a integrated development environment which will allow to compose new processes in a plug & play like fashion.

The talk will illustrate the “technological vision” we are trying to make reality in our research and will explain the technological approaches taken in the ADEPT project and the AristaFlow project in order to meet these goals. Among other things, the developed framework for dynamic WF changes, which enables both, the quick and correct propagation of WF type changes to in-progress WF instances and the ad-hoc adaptation of single WF instances will be presented.

If enough time and interest, the presentation will be complemented by a life demonstration of the experimental ADEPT process management system and the schema evolution and process instance migration facility, which will be part of the ADEPT2 process management system which is currently under implementation at the University of Ulm.

Keywords: Adaptive process management, ad-hoc deviation, process schema evolution, process

Schedulability of Composed Services

Jörg Desel (KU Eichstätt-Ingolstadt, D)

Services are programs triggered by process actions. These programs are often considered functional; they compute single output data from single input data. Reactive programs, consuming and producing strings of data items concurrently, cause additional problems. In this talk, I consider the problem of bounded channels between services; can reactive services be scheduled in such a way that the channels between services do not overflow? I discuss solutions to this problem in the setting of Petri net models. In particular, efficiently decidable necessary conditions for schedulability are presented.

Pricing Web Services

Oliver Günther (HU Berlin, D) This paper focuses on the challenges associated with composing and pricing web services. We present the results of an online experiment, where subjects were confronted with a variety of choices and decisions relating to web service markets and service composition. Our analysis shows that people expect the price of a composite web service to be lower than the sum of the prices of the elementary services, i.e., users are not willing to pay for aggregation by a third party. To obtain a viable business model for composed web services, non-standard pricing mechanisms, such as auctions and negotiations, possibly supported by electronic agents, have to be taken into consideration. Usage-based pricing schemes, combined with an option to switch to a flat subscription, seem most appropriate to penetrate the developing web service market.

Keywords: Composite web services, pricing

Joint work of: Günther, Oliver; Tamm, Gerrit; Leymann, Frank

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/822>

Conversation Patterns

Gregor Hohpe (Google Inc. Mountain View, USA)

One of the tenets of service-oriented development is that services interact through the exchange of messages. Messages provide for robust and loosely coupled interaction. But what happens when your components need to interact in more sophisticated ways than just saying “here’s a message, see ya!”? This is where conversations, choreography, and orchestrations come in. This paper describes the starting point for a catalog of patterns for conversations between services.

Keywords: WS-CDL Choreography Conversation

WS-CDL and Pi-Calculus

Kohei Honda (Queen Mary College - London, GB) This talk is about our experience in the dialogue between industry and theory, in the context of W3C’s working group on a description language for web services called choreography description language (WS-CDL). We shall discuss how such a dialogue can not only benefit industry but also can stimulate a theoretical research.

Participation in W3C’s standardisation process started two years ago, when Steve Ross-Talbot, the chair of W3C’s web service WGs and co-chair of W3C’s CDL working group, invited three pi-calculus experts, Robin Milner, Nobuko Yoshida and me, to the CDL WG. CDL is intended as a description language for web services, Steve’s idea is that both generally and for specific features, the pi-calculus foundation can be very useful.

Our participation in WG resulted in a standard which will become a W3C recommendation soon. It has also led to a new principle of programming which consists of two distinct ways for describing communication-centric software, one centring on global message flows and another centring on end-point behaviour, distilled as formal calculi. The two paradigms of description share a common feature, structured representation of communications. The global calculus originates from CDL itself, while the local calculus is based on the pi-calculus, one of the representative calculi for communicating processes.

The theoretical part of the talk will illustrate these two frameworks using simple but non-trivial examples, present the static and dynamic semantics of the calculi, and show a basic theory of end-point projection — that any well-formed description in the global calculus has a precise representation in the local calculus.

The talk will discuss potential forms of collaboration between industry and theory, what merits such collaboration may have for practical engineering, and how this can give positive feedbacks to academic research.

Keywords: WS-CDL choreography pi-calculus types programming

Enhancing WS-flow Flexibility through Parameterization and Using Semantic Web Service Technologies

Dimka Karastoyanova (Univ. Stuttgart, D)

Compositions of Web Services (WSs) can be implemented using a process based approach. Such compositions are also called WS-flows. They are similar to workflows but describe a process in terms of two dimensions only: control flow and functions. The control flow predefines the execution order of tasks in a process. Activities are implemented by functions, which in the case of WS-flows are only WSs. BPEL is a language for describing WS-flows and is the de-facto standard at the moment.

Since both dimensions of WS-flows are fixed during design time the flexibility of these processes is limited. Moreover, none of the existing approaches towards flexibility are supported in a standardized manner.

Flexibility can be improved by enabling independence of processes on the functions dimension. This can be done by substituting port type and operation names of WSs involved in a process with parameters. The greatest benefit from this approach is achieved if parameters are substituted with concrete values during run time.

Substituting parameters during process run time is impossible with the existing WS-flows meta-model, because there is no available mechanism to compute the parameter values.

This requires an extension of the existing meta-model with a modelling element to stand for such a mechanism, called parameter evaluation strategy. The parameter evaluation strategy is assigned to an interaction activity in a process model during build time, but is executed during run time. This enabled dynamic binding not only to WS ports, but rather to WS port types, too. This means that all instances of the same process model could potentially use services with different port type and operation names. With this approach both the “avoid change” approach to flexibility and “adaptation” are supported; the latter requires tool support. In the presented work we use four main mechanisms to compute parameter values: static, prompt, fromVariable and query. They exhibit different degrees of flexibility and automation. The fully automated mechanism is represented by the strategy called “query”. It can utilize any semantic WS technology. Semantic WS descriptions describe a WS in terms of its capabilities, rather than its interface. Similarly, the requirements to a type of service to be used in a process model can be described using a semantic description, which is the input data required by the query strategy. Whenever a parameterized activity is executed and it has the query strategy assigned, the semantic description of the requirements is used to discover WSs compliant with these requirements. Once a compliant service is selected its port type and operation names are used to perform the actual invocation.

The parameter computation mechanisms must be supported by the process execution environment (e.g. a BPEL engine). The discovery and selection of semantic WSs is performed by a piece of middleware, typically denoted by the

name enterprise service bus. A prototypical implementation of an engine supporting the parameter evaluation strategy element in BPEL and a service broker supporting discovery and selection of OWL-S services will be demonstrated as a part of the talk.

Keywords: WS-flows, parameterization, parameter substitution, Semantic Web services

AMFIBIA and SOA

Ekkart Kindler (Universität Paderborn, D)

Today, there are many different formalisms and notations for modelling business processes. Though most of the formalisms have their justification, the plethora of notations makes it hard to compare and to exchange business process models among different tools.

AMFIBIA (A Meta-model For the Integration of Business process modelling Aspects) sets out to capture the basic aspects of business process models and to define their concepts independently from a particular formalism and notation, and then map different formalisms to these basic concepts. This way, business process models can be compared with each other, and it will be even possible, to integrate and combine different formalisms in a single workflow engine. Currently, we implement a prototype of a workflow engine, which supports the concepts of AMFIBIA.

Since the development of AMFIBIA started quite late in the history of workflow management, it might not have strong impact on existing workflow management systems. The concepts of AMFIBIA, however, should be applicable to SOA, where formalism independence is even more important.

The talk presents the ideas and concepts of AMFIBIA and intends to trigger a discussion on the aspects of SOA and the aspects and concepts that need to be captured in SOA.

Keywords: Aspect oriented modelling, Formalism independence, BPM

Full Paper:

<http://wwwcs.uni-paderborn.de/cs/kindler/research/AMFIBIA/>

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/823>

Demo: EPC Tools - a tool (and platform) for simulating and analysing EPCs

Ekkart Kindler (Universität Paderborn, D)

Due to its non-locality, the semantics of Event driven Process Chains (EPCs) is sometimes a bit awkward; some EPCs do not even have a semantics, which we call unclean. EPC Tools help checking whether a EPC is clean.

Moreover, it helps simulating EPCs and analysing some of its properties such as soundness.

Keywords: Event driven Process Chains; EPCs, semantics, analysis

Full Paper:

<http://www.upb.de/cs/kindler/publications/copies/CFK-EPK05.pdf>

See also: Nicolas Cuntz, Jörn Freiheit, and Ekkart Kindler: On the semantics of EPCs: Faster calculation for EPCs with small state spaces. In: M. Nüttgens, F. J. Rump (eds.): EPK 2005: Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Hamburg. Proceedings, pp. 7-23, December 2005. Also: CEUR Workshop Proceedings, Vol. 167. In: M. Nüttgens, F. J. Rump (eds.): EPK 2005: Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Hamburg. Proceedings, pp. 7-23, December 2005. Also: CEUR Workshop Proceedings, Vol. 167

BPEL4People Overview

Matthias Kloppmann (IBM - Böblingen, D)

Human user interactions are currently not covered by the Web Services Business Processes Execution Language (WS-BPEL), which is primarily designed to support automated business processes based on Web services. In practice, however, many business process scenarios require user interaction.

BPEL4People describes scenarios where users are involved in business processes, and defines appropriate extensions to WS-BPEL to address these.

Keywords: BPEL People BPEL4People

Joint work of: Kloppmann, Matthias; König, Dieter; Leymann, Frank; Pfau, Gerhard; Rickayzen, Alan; von Riegen, Claus; Schmidt, Patrick; Trickovic, Ivana

Service Composition - WS-BPEL and SCA

Dieter König (IBM - Böblingen, D)

Web Services-Business Process Execution Language (WS-BPEL) provides a model for describing the behavior of a business process based on stateful, long-running interactions between the process and its partners. The interaction with each partner occurs through Web service interfaces, and the structure of the relationship at the interface level is encapsulated in a partner link.

Service Component Architecture (SCA) provides an assembly model for services and a model for applying infrastructure capabilities to services, such as security and transactions.

The SCA WS-BPEL Client and Implementation model specifies how any WS-BPEL process can be used as the implementation of a component within SCA assembly model. Conversely, it specifies how to use WS-BPEL to implement any SCA component type that uses only WSDL interfaces to define services and references, possibly with some SCA specific extensions used in process definition.

In this session, we discuss the relationships between WS-BPEL and SCA and show how WS-BPEL processes can be used as SCA service implementations in a Service Oriented Architecture.

Keywords: Business Processes, WS-BPEL, Web Services, Service Composition, Service Components, Service Component Architecture, SCA

Web Service Matching - A Relational Approach

Carolin Letz (Universität Münster, D)

With an increasing number of Web services the time saved by using an existing service instead of implementing it anew is made up by the time spend searching for an adequate service provider, e.g., in an outsourcing scenario.

We propose a simple relational Web service description that allows to analyze service matching. We show

- (i) that the result of a match computation between two Web services can be regarded as a new Web service and
- (ii) that the result of the match computation can be used to establish a hierarchy of matches that indicates if the match is useful or not.

Keywords: Web Service Matchmaking

Joint work of: Letz, Carolin; Vossen, Gottfried

Analyzing Interacting BPEL Processes: A Tool Demo

Niels Lohmann (HU Berlin, D)

In the talk/demonstration, theoretical results on open workflow nets (oWFN), e.g. controllability (the question whether a proper partner exists) and operating guidelines (a characterization of all proper partners) are linked to BPEL.

A tool chain is presented to automatically analyze BPEL processes: firstly BPEL2oWFN, a tool translating BPEL processes to Petri net models (e.g. oWFN or simple P/T nets) is demonstrated. The generated nets are then analyzed by Fiona. Fiona checks whether the given net is controllable and if it is generates a partner. Furthermore, the operating guideline can be calculated. A small example with a subtle design flaw shows the value of the tool chain

Keywords: BPEL, Petri nets, oWFN, controllability, operating guidelines, tool support

See also:

<http://www.informatik.hu-berlin.de/top/tools4bpel/tools.html>

Analyzing Interacting BPEL Processes

Niels Lohmann (HU Berlin, D)

This paper addresses the problem of analyzing the interaction between BPEL processes. We present a technology chain that starts out with a BPEL process and transforms it into a Petri net model. On the model we decide *controllability* of the process (the existence of a partner process, such that both can interact properly) and compute its *operating guideline* (a characterization of all properly interacting partner processes). A case study demonstrates the value of this technology chain.

Keywords: Business process modeling and analysis; Petri Nets

Joint work of: Lohmann, Niels; Massuthe, Peter; Stahl, Christian; Weinberg, Daniela

See also: Analyzing Interacting BPEL Processes. In Schahram Dustdar, José Luiz Fiadeiro, and Amit Sheth, editors, Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006, Proceedings, volume 4102 of Lecture Notes in Computer Science, pages 17-32, September 2006. Springer-Verlag.

An Algorithm for Matching Nondeterministic Services with Operating Guidelines

Peter Massuthe (HU Berlin, D)

Interorganizational cooperation is more and more organized by the paradigm of services. Service-oriented architectures (SOA) provide a general framework for service interaction. SOA describe three roles of services, the service provider, the service requester, and the service broker, together with the three operations publish, find, and bind.

We provide a formal method based on nondeterministic automata to model services and their interaction. In this paper, we restrict ourselves to finite and acyclic automata. We suggest operating guidelines as a convenient and intuitive artifact to realize the publish operation. In our approach, the find operation reduces to a matching problem between the requester's service and the published operating guidelines. If matching services are actually bound together, our approach guarantees deadlock-free communication. In this paper, matching of deterministic as well as nondeterministic automata with operating guidelines is presented.

Keywords: Services, SOA, Formal Methods, (Nondeterministic) Automata, Operating guidelines, Matching

Joint work of: Massuthe, Peter; Wolf, Karsten

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/824>

Aspect-Oriented Techniques for Web Services: a Model-Driven Approach

Guadalupe Ortiz (Centro Univ. de Merida, E)

Web Service technologies offer a successful way for interoperability among applications, but in order to tackle the entire web service life cycle, it is necessary to face how to model systems based on service functionality and also how to add extra-functional properties to modelled services. In this regard, we propose first of all a versatile and simple UML profile based on the Service Component Architecture specification for modelling services, in order to provide a model environment in which to add extra-functional properties.

Secondly, a new UML profile is proposed to model and reuse the said extra-functional properties in service models. The implemented models based on these profiles will be independent from a final implementation language or platform, thus it is necessary to specify a particular type of model to convert the independent one into in a subsequent step.

In order to meet this requirement an object, an aspect and a policy based models are proposed as the intermediate step between the independent model and the final code. We acknowledge that there are tools available which convert Java models into web services' Java code, and it is not our aim to build a tool to fulfil this requirement. Regarding extra-functional properties, aspect-oriented techniques allow them to be easily modularized and reused; in this respect, properties are implemented as aspects in a totally transparent way and avoid the need to modify service code in an intrusive manner on adding extra-functional properties, improving our system maintenance. Furthermore, this way traceability between the model and the code is perfectly maintained in both directions.

This work has been developed thanks to the support of CICYT under contract TIN2005-09405-C02-02.

Keywords: Web Service, Model Driven Development, Aspect-Oriented Techniques, Extra-Functional Property, Service Component Architecture, Ws-Policy

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/826>

Towards UML Modelling Extra-Functional Properties in Web Services and their Clients

Guadalupe Ortiz (Centro Univ. de Merida, E)

Web Services provide our systems with a platform independent and loosely coupled implementation environment, being time to face how the named systems can be modelled. Service Component Architecture (SCA) allows us to define services independently of the final implementation technology; however, it does not integrate the remaining development stages. Model Driven Architecture provides a method to face all stages in development from the platform independent

model to final code, although it is not specific to service technologies. Regarding web service extra-functional properties, WS-Policy establishes how to describe them in a loosely coupled manner; however the loosely coupled environment is not always maintained when modelling or implementing these properties, which can be solved by using aspect-oriented techniques. In this paper, we propose to use a model driven approach for extra-functional properties in SCA service based models, where generated code will consist of the policy description and an aspect-oriented implementation.

Keywords: Extra-Functional property, web service, UML modeling, aspect-oriented techniques, WS-policy, service component architecture

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/825>

Human Task Manager – Involving people in SOA-based Applications

Gerhard Pfau (IBM - Böblingen, D)

The involvement of people in service compositions is a relatively new facet of Service-Oriented Architecture (SOA), expanding the ways software can model how humans work and interact in a business. Human tasks, as recently introduced with WebSphere Process Server, provide the infrastructure for human-based Web services. They are based on the staff functions known from classical human workflow systems, with their functionality extended to include sophisticated escalation and notification capabilities, as well as the ability to specify WebSphere Portal-based user interfaces for human tasks.

This presentation provides an introduction to Human Task Manager, a component delivered as part of WebSphere Process Server used to manage and execute human tasks. It outlines the major usage scenarios of human tasks and shows how to involve people in service choreographies as service originators, or as a human-based “service implementation”. We also discuss purely human-centric scenarios, where people create human tasks for other people, without the involvement of any automated service. Furthermore we compare the different scenarios supported by pre-modeled human tasks versus ad-hoc defined tasks, and demonstrate how ad-hoc defined human tasks can be used for ad-hoc human collaboration. Finally we show that offering human tasks as standalone components (standalone human tasks) complements the capabilities of human tasks embedded in business processes (inline human tasks). While standalone human tasks offer re-use capabilities across SOA applications, inline human tasks have access to the context of the surrounding business process, and thus enable classical human workflow scenarios like the “separation of duties” (aka the “four-eyes principle”).

Keywords: Human Workflow, Human Tasks, task, business process, BPEL, WS-BPEL, ad-hoc, collaboration

From Enterprise Services to Composite Applications

Alan Rickayzen (SAP - Walldorf, D)

SAP introduced Enterprise SOA as a foundation for service-enabled business solutions. Business solutions are built from independent, re-usable building blocks called Enterprise Services. This presentation explains what Enterprise Services are, how they are used to build cross-system applications, and the Enterprise Service Community Process, SAP established recently, for creating enterprise service definitions jointly with partners and ISVs.

Joint work of: Rickayzen, Alan; Trickovic, Ivana

A SOA-Based Architecture Framework

Christian Stahl (HU Berlin, D)

In this paper we present first results of a SOA-based architecture framework. The architecture framework is required to be close to industry standards, especially to service component architecture (SCA), language independent (i.e. it is adoptable) and the building blocks of a system, activities and data, are first class citizens. We present a meta model of the architecture framework and discuss its concepts in detail.

Keywords: SOA, architecture framework

Joint work of: Aalst, Wil van der; Beisiegel, Michael; Hee, Kees van; König, Dieter; Stahl, Christian

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/827>

Business process experience for the masses

Satish R. Thatte (Microsoft Corp. - Redmond, USA)

Real world business processes are a complex web of structured business data, business logic, workflows, and business events, together with a large amount of “unstructured” document centric collaborative work where the creative non-stereotyped aspects of business process get accomplished. The gap that exists today between the structured and unstructured worlds is a serious barrier to productivity and the full realization of the value of software assets. The talk will discuss the nature of the problem and some of the work underway at Microsoft to bridge the two worlds.

From Enterprise Services to Composite Applications

Ivana Trickovic (SAP - Walldorf, D)

SAP introduced Enterprise SOA as a foundation for service-enabled business solutions. Business solutions are built from independent, re-usable building blocks called Enterprise Services. This presentation explains what Enterprise Services are, how they are used to build cross-system applications, and the Enterprise Service Community Process, SAP established recently, for creating enterprise service definitions jointly with partners and ISVs.

Joint work of: Rickayzen, Alan; Trickovic, Ivana

Techniques for Business-Driven Development

Jussi Vanhatalo (IBM Research - Zürich, CH)

The talk explores the emerging paradigm of business-driven development, which presupposes a methodology for developing IT solutions that directly satisfy business requirements and needs. At the core of business-driven development are business processes, which are usually modeled by combining graphical and textual notations.

During the business-driven development process, business-process models are taken down to the IT level, where they describe the so-called choreography of services in a Service-Oriented Architecture. The derivation of a service choreography based on a business-process model is simple and straightforward for toy examples only. For realistic applications, many challenges at the methodological and technical level have to be solved. The talk explores these challenges and describes selected solutions that have been developed by the research team of the IBM Zurich Research Laboratory.

Towards a services-based process platform

Mathias Weske (Hasso-Plattner-Institut - Potsdam, D)

The talk bases on the state of the art in business process management and highlights challenges imposed by service oriented architectures. The design, configuration, and enactment phases of the business process lifecycle need to be re-investigated in the light of service oriented architectures, in particular dynamic binding of service implementations to service specifications. Paths to create service compositions based on mappings of service parameters to a domain ontology are investigated. The considerations are illustrated by a usage scenario and a service-based process platform developed in the context of Adaptive Services Grid, a European research project in FP6.

Joint work of: Weske, Mathias

Towards Services-based Process Platforms

Mathias Weske (Hasso-Plattner-Institut - Potsdam, D)

Service oriented software architectures that are aware of business processes will form the core of operational IT landscapes in the future.

This contribution starts with an introduction of the state of the art in service oriented architectures. A concrete case study identifies central requirements that are not satisfied by these architectures so far. The authors argue that semantically rich descriptions of services are essential to tap the full potential of service oriented architectures in enterprise environments. This regards matchmaking and binding of services, integration of new services as well as the cost-efficient development of added value services by composing semantically described basic services. This paper introduces a semantic service platform that implements dynamic matchmaking, composition and binding of semantically described services. Finally its functionality and possible application scenarios are outlined.

Keywords: Business process platform, Semantic Services, Service-oriented architectures

Joint work of: Dominik Kuroпка and Mathias Weske

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/831>

Controllability: A Soundness Criterion for Services

Karsten Wolf (Universität Rostock, D)

Soundness is an established correctness criterion for workflow nets. Open workflow nets extend workflow nets with an interface. They are suitable for modelling services. Controllability lifts the concept of soundness to open workflow nets.

We present several variations of the controllability problem. In particular, we distinguish centralized and decentralized environments for a service. For deciding controllability, we synthesize a partner service to a given service. In some settings, we can show that a most permissive partner service exists.

Keywords: Service, soundness, controllability, partner synthesis

Modeling Process-Oriented Integration of Services Using Patterns and Pattern Primitives

Uwe Zdun (TU Wien, A)

Service-oriented architectures are increasingly used in the context of business processes. However, the proven practices for process-oriented integration of services are not well documented yet. In addition, modeling approaches for

the integration of processes and services are neither mature nor do they exactly reflect the proven practices. We propose a pattern language for process-oriented integration of services to describe the proven practices that we have mined in a large number of industrial case studies. Next, a concept based on pattern primitives is introduced to (semi-)formally model those patterns. A pattern primitive is a fundamental, formalizable modeling element that represents a pattern. We present a catalog of pattern primitives that are formally modeled using OCL constraints and map those primitives to the patterns in the pattern language of process-oriented integration of services.

Model-Driven and Pattern-Based Integration of Process-Driven SOA Models

Uwe Zdun (TU Wien, A)

Service-oriented architectures (SOA) are increasingly used in the context of business processes. However, the modeling approaches for process-driven SOAs do not yet sufficiently integrate the various kinds of models relevant for a process-driven SOA – ranging from process models to software architectural models to software design models. We propose to integrate process-driven SOA models via a model-driven software development approach that is based on proven practices documented as software patterns. We introduce pattern primitives as an intermediate abstraction to formally model the participants in the solutions that patterns convey. To enable model-driven development, we develop domain-specific modeling languages for each kind of process-driven SOA model - based on formal meta-models that are extended with the pattern primitives. The various process-driven SOA models are integrated in a model-driven tool chain via the meta-models. Our tool chain validates the process-driven SOA models with regard to the constraints given by the meta-models and primitives.

Keywords: Process-Driven SOA, Pattern, Pattern Primitives

Joint work of: Zdun, Uwe; Dustdar, Schahram

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/820>

Relationships between services, components and workflows

K.M. van Hee (TU Eindhoven, NL)

Although workflow models play an important role in modeling business processes and their orchestration, their role in software engineering is still limited. In the SOA (service-oriented architecture) approach to software engineering components deliver services. The services they deliver are built from other services that are “outsourced” to other components. Therefore one of the key functions of a component is the orchestration of these outsourced services. The orchestration is performed by workflows.

In the talk we will make a step towards a modeling framework where these concepts fit together in a natural way and which allows for verification.

Keywords: Petri nets, SOA

Choreography Conformance Checking: An Approach based on BPEL and Petri Nets

Wil van der Aalst (TU of Eindhoven, NL)

Recently, languages such as BPEL and WS-CDL have been proposed to describe interactions between services and their behavioral dependencies.

The emergence of these languages heralds an era where richer service descriptions, going beyond WSDL-like interfaces, will be available. However, what can these richer service descriptions serve for? This talk discussed the use of event logs of web services and behavioral service descriptions as input for process mining and conformance checking. Conformance checking is the act of verifying whether or not one or more parties stick to an agreed-upon behavior, by observing their actual behavior as recorded in message logs. This talk shows that it is possible to translate BPEL business abstract processes to Petri nets and to relate SOAP messages to transitions in the Petri net. The approach has been implemented in the ProM framework.

Keywords: Process mining, SOAP messages, BPEL, Petri nets

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/830>

DecSerFlow: Towards a Truly Declarative Service Flow Language

Wil van der Aalst (TU of Eindhoven, NL)

The need for process support in the context of web services has triggered the development of many languages, systems, and standards.

Industry has been developing software solutions and proposing standards such as BPEL, while researchers have been advocating the use of formal methods such as Petri nets and pi-calculus. The languages developed for service flows, i.e., process specification languages for web services, have adopted many concepts from classical workflow management systems. As a result, these languages are rather procedural and this does not fit well with the autonomous nature of services. Therefore, we propose DecSerFlow as a Declarative Service Flow Language. DecSerFlow can be used to specify, enact, and monitor service flows. The language is extendible (i.e., constructs can be added without changing the engine or semantical basis) and can be used to enforce or to check the conformance of service flows. Although the language has an appealing graphical representation, it is grounded in temporal logic.

Keywords: DecSerFlow, LTL, service flows, web services, SOA

Joint work of: van der Aalst, Wil; Pesic, Maja

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2006/829>