facultad de informática

universidad politécnica de madrid

S-Cube-related non-CLIP Publications by Topic and year

S-Cube UPM team

TR Number S-Cube-RelatedPubs/2008 v. 124

Printing date: June 30, 2008

S-Cube-related non-CLIP Publications by Topic and year

Technical Report Number: S-Cube-RelatedPubs/2008 v. 124

Printing date: June 30, 2008

Authors

The CLIP Group
`clip@clip.dia.fi.upm.es` Universidad Politécnica de Madrid (UPM), Facultad de Informática, 28660 Boadilla del Monte, Madrid — Spain

Keywords

Publications, by topic, by year, S-Cube, CLIP Group

Abstract

This document provides a list of some publications related to the S-Cube project classified, independently, by *research topic* (as well as by type of publication – journal, book, invited, workshop, etc.) and by *year of publication.* We ask for understanding with missing items in the classification, missclassified papers, and relevant papers not appearing in this list, and at the same time will very much appreciate any pointers to them. This list is automatically generated out of a BiBTeX file and (if available) an electronic version of the paper. This means that adding large quantities of papers will be likely to take time.
You should not download papers from this site unless you are sure that you can download them legally from somewhere else – e.g., if your institution has an agreement with the publishers. The papers available here are not stored to bypass regulations, but to have references handy and centralized.
The original scripts and setup this site is based on were written and devised by M. Hermenegildo. A lot of time was invested by Dragan Ivanovic in generating the first version of the BiBTeX file with extended classification information regarding the topics the papers dealt with. Many people from the S-Cube NoE contributed with paper references.

Contents

Chapter 1

Publications by Topic

1 Publications in **Program Analysis**

Articles in Refereed Journals:

1. J. Cardoso. *Complexity analysis of BPEL Web processes.* Software Process: Improvement and Practice, Vol. 12, Num. 1, pages 35–49, Whiley-Interscience, 2007.

   Abstract: *Several organizations have already realized the potential of using WS-BEPL, the Process Execution Language for Web Services, to model the behavior of Web services in business processes. WS-BPEL provides a model for describing simple or complex interactions between business partners. In some cases, WS-BPEL process designs can be highly complex, due, for example, to the vast number of Web services carried out in global markets. High complexity in a process has several undesirable drawbacks; it may result in poor understandability, more errors, defects, and exceptions, leading to processes requiring more time to be developed, tested, and maintained. Therefore, excessive complexity should be avoided. Processes that are highly complex tend be less flexible, since it is more complicated to make changes to the process. The major goal of this article is to present two metrics to analyze the control-flow complexity (CFC) of WS-BPEL Web processes. The metrics are to be used at design-time to evaluate the complexity of a process design before implementation actually takes place.*

Articles in Refereed Conferences:

1. Moreno Marzolla, Raffaela Mirandola. *Performance Prediction of Web Service Workflows.* Software Architectures, Components, and Applications, Lecture Notes in Computer Science, Vol. 4880, pages 127–144, January 2008.

   Abstract: *Web Services play an important role in the SOA paradigm, as they allow services to be selected on-the-fly to build applications out of existing components. In this scenario, the BPEL notation can be used as an orchestration language which allows the user to describe interactions with Web Services in a standard way. The performance of a BPEL workflow is a very important factor for deciding which components must be selected, or to choose whether a given sequence of interactions can provide the requested quality of service. Due to its very dynamic nature, workflow performance evaluation can not be accomplished using traditional, heavy-weight techniques. In this paper we present a multi-view approach for the performance prediction of service-based applications encompassing both users and service provider(s) perspectives. As a first step towards the realization of this integrated framework we present an efficient approach for performance assessment of Web Service workflows described using the BPEL notation. Starting from annotated BPEL and WSDL specifications, we derive performance bounds on response time and throughput. In such a way users are able to assess the efficiency of a BPEL workflow, while service provider(s) can perform sizing studies or estimate performance gains of alternative upgrades to existing systems. To bring this approach to fruition we developed a prototype tool called bpel2qnbound, using which we analyze a simple case study.*

2. Dmytro Rud, Andreas Schmietendorf, Reiner Dumke. *Performance Modeling of WS-BPEL-Based Web Service Compositions.* Services Computing Workshops, pages 140–147, September 2006.

   Abstract: *This paper addresses quality of service aspects of Web service orchestrations created using WS-BPEL from the standpoint of a Web service integrator. A mathematical model based on operations research techniques and formal semantics of WS-BPEL is proposed to estimate and forecast the influence of the execution of orchestrated processes on utilization and throughput of individual involved nodes and of the whole system. This model is applied to the optimization of service levels agreement process between the involved parties*

3. Jorge Cardoso. *About the Data-Flow Complexity of Web Processes.* 6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility, pages 67–74, 2005.

   Abstract: *Abstract. Organizations are increasingly faced with the challenge of managing e-commerce and e-business applications involving Web services and Web processes. In some cases, Web processes design* *can be highly complex, due, for example, to the vast number of services carried out in global markets. High complexity in a process has several undesirable drawbacks, it may result in bad understandability, more errors, defects, and exceptions leading processes to need more time to develop, test, and maintain. Therefore, excessive complexity should be avoided. Flexibility and complexity are guiding principles in the design of business processes. In most business processes, flexibility and complexity are related inversely. Processes with a high complexity tend be less flexible, since it is more complicated to make changes to the process. In our previous work we have defined a metric to measure the control-flow complexity of Web processes. The major goal of this paper is to study the issues and establish the requirements for the development of a metric to analyze the data-flow complexity of Web processes.*

4. Xiang Fu, Tevfik Bultan, Jianwen Su. *Analysis of Interacting BPEL Web Services.* WWW'04: Proceedings of the 13th International Conference on World Wide Web, pages 621–630, ACM Press, 2004.

   Abstract: *This paper presents a set of tools and techniques for analyzing interactions of composite web services which are specified in BPEL and communicate through asynchronous XML messages. We model the interactions of composite web services as conversations, the global sequence of messages ex- changed by the web services. As opposed to earlier work, our tool-set handles rich data manipulation via XPath expressions. This allows us to verify designs at a more detailed level and check properties about message content. We present a framework where BPEL specifications of web ser- vices are translated to an intermediate representation, followed by the translation of the intermediate representation to a verification language. As an intermediate representation we use guarded automata augmented with unbounded queues for incoming messages, where the guards are expressed as XPath expressions. As the target verification language we use Promela, input language of the model checker SPIN. Since SPIN model checker is a finite-state verification tool we can only achieve partial verification by fixing the sizes of the input queues in the translation. We propose the concept of synchronizability to address this problem. We show that if a composite web service is synchronizable, then its conversation set remains same when asynchronous communication is replaced with synchronous communication. We give a set of sufficient conditions that guarantee synchronizability and that can be checked statically. Based on our synchronizability results, we show that a large class of composite web services with unbounded input queues can be completely verified using a finite state model checker such as SPIN.*

Technical Reports and Manuals:

1. Oliver Kopp, Rania Khalaf, Frank Leymann. *Reaching Definitions Analysis Respecting Dead Path Elimination Semantics in BPEL Processes.* Num. 2007/04, Institut für Architektur von Anwendungssystemen, November 2007.

   Abstract: *The Business Process Execution Language for Web Services (BPEL) is a workflow language geared towards Service Oriented Computing. BPEL provides basic workflow capabilities, such as the ability to impose control on a set of activities via explicit links, as well as advanced features such as recovery and event handling. The area of workflow often requires understanding the data dependencies as well as the control dependencies between activities. This aids in business process design as well as in analysis and rengineering. BPEL, however, uses shared variables to model data. Activities read and write data to these variables, i.e. there is no explicit data link construct. In order to draw out the flow of data between the activities of a BPEL process, one must therefore perform data-flow analysis on that process. To address this problem, we present an algorithm that statically determines such data dependencies. "Statically" means, that we do not determine the dependencies per process instance at runtime, but on the process model without creating any instances. Mainstream data-flow analysis techniques are presented in [ALSU06, Muc97, NNH04]. However, these techniques cannot be directly applied to a BPEL process, since BPEL supports both parallelism and dead path elimination (DPE, [CDG+ 03, OAS07, CKLW03, BK05, LR00]). DPE is a technique used in BPEL to propagate activity disablement down paths that can no longer be executed. The algorithm in this paper consists of reaching definitions analysis dealing with DPE, enabling it to reduce the number of data dependencies when compared to approaches not dealing with DPE.*

2 Publications in **Programming Language Design**

Articles in Refereed Conferences:

1. Luciano Baresi, Sam Guinea. *Towards Dynamic Monitoring of WS-BPEL Processes.* ICSOC, pages 269–282, 2005.

   Abstract: *The intrinsic flexibility and dynamism of service-centric applications preclude their pre-release validation and demand for suitable probes to monitor their behavior at run-time. Probes must be suitably activated and deactivated according to the context in which the application is executed, but also according to the confidence we get on its quality. The paper supports the idea that significant data may come from very different sources and probes must be able to accommodate all of them. The paper presents: (1) an approach to specify monitoring directives, called monitoring rules, and weave them dynamically into the process they belong to; (2) a proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time; (3) a user-oriented language to integrate data acquisition and analysis into monitoring rules.*

3 Publications in **Constraint Programming**

Articles in Refereed Journals:

1. Christel Baier, Marjan Sirjani, Farhad Arbab, Jan J. M. M. Rutten. *Modeling component connectors in Reo by constraint automata.* Sci. Comput. Program., Vol. 61, Num. 2, pages 75–113, 2006.

   Abstract: *In this paper we introduce constraint automata and propose them as an operational model for Reo, an exogenous coordination language for compositional construction of component connectors based on a calculus of channels. By providing composition operators for constraint automata and defining notions of equivalence and refinement relations for them, this paper covers the foundations for building tools to address concerns such as the automated construction of the automaton for a given component connector, equivalence checking or containment checking of the behavior of two given connectors, and verification of coordination mechanisms.*

2. George A. Papadopoulos, Farhad Arbab. *Coordination Models and Languages.* Advances in Computers, Vol. 46, pages 330–401, 1998.

   Abstract: *A new class of models, formalisms and mechanisms has recently evolved for describing concurrent and distributed computations based on the concept of "coordination". The purpose of a coordination model and associated language is to provide a means of integrating a number of possibly heterogeneous components together, by interfacing with each component in such a way that the collective set forms a single application that can execute on and take advantage of parallel and distributed systems. In this chapter we initially define and present in sufficient detail the fundamental concepts of what constitutes a coordination model or language. We then go on to classify these models and languages as either "data-driven" or "control-driven" (also called "process-" or "task-oriented"). Next, the main existing coordination models and languages are described in sufficient detail to let the reader appreciate their features and put them into perspective with respect to each other. The chapter ends with a discussion comparing the various models and some conclusions.*

Articles in Refereed Conferences:

1. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, Marco Montali, Sergio Storari, Paolo Torroni. *Computational Logic for Run-Time Verification of Web Services Choreographies: Exploiting the OCS-SI Tool.* WS-FM, pages 58–72, 2006.

   Abstract: *In this work, we investigate the feasibility of using a framework based on computational logic, and mainly defined in the context of Multi-Agent Systems for Global Computing (SOCS UE Project), for modeling choreographies of Web Services with respect to the conversational aspect. One of the fundamental motivations of using computational logic, beside its declarative and highly expressive nature, is given by its operational counterpart, that can provide a proof-theoretic framework able to verify the consistency of services designed in a cooperative and incremental manner. In particular, in this paper we show that suitable "Social Integrity Constraints", introduced in the SOCS social model, can be used for specifying global protocols at the choreography level. In this way, we can use a suitable tool, derived from the proof-procedure defined in the context of the SOCS project, to check at run-time whether a set of existing services behave in a conformant manner w.r.t. the defined choreography.*

2. Laurent Fribourg, Marcos Veloso Peixoto. *Concurrent Constraint Automata.* International Logic Programming Symposium, 1993.

   Abstract: *We address the problem of the specification and the proof of properties of concurrent systems which manipulate an unbounded number of data. We propose an approach based on an extended notion of automata, called "Concurrent Constraint Automata (CCA)". A CCA is an automaton with constraints and synchronous communication. By automata with constraints, we mean a state machine whose actions and states contain parameters that take their value in the set of natural numbers. (...)*

4 Publications in **Programming Language Implementation**

Articles in Refereed Conferences:

1. Tevfik Bultan, Xiang Fu, Jianwen Su. *Tools for Automated Verification of Web Services.* ATVA, pages 8–10, 2004.

5 Publications in **All topics related to software services**

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

2. Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann. *Service-Oriented Computing: State of the Art and Research Challenges.* IEEE Computer, To Appear, November 2007.

   Abstract: *Service-oriented computing promotes the idea of assembling application components into a network of services that can be loosely coupled to create flexible, dynamic business processes and agile applications that span organizations and computing platforms. An SOC research road map provides a context for exploring ongoing research activities.*

3. Ardagna, D., Pernici, B.. *Adaptive Service Composition in Flexible Processes.* Software Engineering, IEEE Transactions on, Vol. 33, Num. 6, pages 369–384, June 2007.

   Abstract: *In advanced service oriented systems, complex applications, described as abstract business processes, can be executed by invoking a number of available Web services. End users can specify different preferences and constraints and service selection can be performed dynamically identifying the best set of services available at runtime. In this paper, we introduce a new modeling approach to the Web service selection problem that is particularly effective for large processes and when QoS constraints are severe. In the model, the Web service selection problem is formalized as a mixed integer linear programming problem, loops peeling is adopted in the optimization, and constraints posed by stateful Web services are considered. Moreover, negotiation techniques are exploited to identify a feasible solution of the problem, if one does not exist. Experimental results compare our method with other solutions proposed in the literature and demonstrate the effectiveness of our approach toward the identification of an optimal solution to the QoS constrained Web service selection problem*

4. Bianculli, D., Ghezzi, C., Spoletini, P.. *A Model Checking Approach to Verify BPEL4WS Workflows.* Service-Oriented Computing and Applications, 2007. SOCA '07. IEEE International Conference on, To Appear, June 2007.

   Abstract: *The increasing diffusion of service oriented computing in critical business transactions demands reliability and correctness of the workflow logic representing web service orchestrations. We present an approach for the formal verification of workflow-based compositions of web services, described in BPEL4WS. Workflow processes can be verified in isolation, assuming that the external services invoked are known only through their interface. It is also possible to verify that the actual composition of two or more processes behaves correctly. We can verify deadlock freedom, properties expressed as data-bound assertions written in WS-CoL, a specification language for web services, and LTL temporal properties. Our approach is based on* *the software model checker Bogor, whose language supports the modeling of all BPEL4WS constructs. We provide an empirical evaluation of our approach and we compare the results with other BPEL4WS model checking tools.*

5. Chun Ouyanga, Eric Verbeekb, Wil M.P. van der Aalst, Stephan Breutel, Marlon Dumas, Arthur H.M. ter Hofstede. *Formal semantics and analysis of control flow in WS-BPEL.* Science of Computer Programming, Vol. 67, Num. 2-3, pages 162–198, April 2007.

   Abstract: *Web service composition refers to the creation of new (Web) services by combining functionalities provided by existing ones. A number of domain-specific languages for service composition have been proposed, with consensus being formed around a process-oriented language known as WS-BPEL (or BPEL). The kernel of BPEL consists of simple communication primitives that may be combined using control-flow constructs expressing sequence, branching, parallelism, synchronization, etc. We present a comprehensive and rigorously defined mapping of BPEL constructs onto Petri net structures, and use this for the analysis of various dynamic properties related to unreachable activities, conflicting messages, garbage collection, conformance checking, and deadlocks and lifelocks in interaction processes. We use a mapping onto Petri nets because this allows us to use existing theoretical results and analysis tools. Unlike approaches based on finite state machines, we do not need to construct the state space, and can use structural analysis (e.g., transition invariants) instead. We have implemented a tool that translates BPEL processes into Petri nets and then applies Petri-net-based analysis techniques. This tool has been tested on different examples, and has been used to answer a variety of questions.*

6. Roberto Lucchi, Manuel Mazzara. *A pi-calculus based semantics for WS-BPEL .* Journal of Logic and Algebraic Programming, Vol. 70, Num. 1, pages 96–118, January 2007.

   Abstract: *Recently, the term Web services orchestration has been introduced to address some issues related to Web services composition, that is the way of defining a complex service out of simpler ones. Several proposals for describing orchestration for business processes have been presented in the last years and many of these languages make use of concepts as long-running transactions and compensations for coping with error handling. WS-BPEL 2.0, the most credited candidate for becoming a standard, provides three different mechanisms allowing to cope with abnormal situations: exception, event and compensation handling. This complexity makes it difficult to formally define the framework, thus limiting the formal reasoning about the designed applications. In this paper we advocate that three different mechanisms for error handling are not necessary and we formalize a novel orchestration language based on the idea of event notification as the unique error handling mechanism. To this end, we formally define the three BPEL mechanisms in terms of our calculus. It is possible to take advantages of this formal description in two ways. Firstly, this language represents by itself a proposal of simplification for WS-BPEL 2.0 including an unambiguous specification. Secondly, an implementor of an actual WS-BPEL 2.0 orchestration engine could implement simply this single mechanism providing all the remaining ones by compilation. With this attempt we intend to give a concrete contribute towards the improvement of the quality of the BPEL specification, the applicability of BPEL itself and the implementation of real orchestration engines. Finally, as a case study we consider some of the hundreds of open issues met by the WS-BPEL designers and we propose a solution making use of the experience gained developing our algebra.*

7. D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, P. Plebani. *PAWS: A Framework for Executing Adaptive Web-Service Processes.* IEEE Software, To Appear, 2007.

   Abstract: *The processes with adaptive Web services framework couples design-time and runtime mechanisms to flexibly and adoptively execute managed Web-services-based business processes.*

8. Stefano Bistarelli, Ugo Montanari, Francesca Rossi, Francesco Santini. *Unicast and Multicast Qos Routing with Soft Constraint Logic Programming.* CoRR, Vol. abs/0704.1783, 2007.

   Abstract: *We present a formal model to represent and solve the unicast/multicast routing problem in networks with Quality of Service (QoS) requirements. To attain this, first we translate the network adapting it to a weighted graph (unicast) or and-or graph (multicast), where the weight on a connector corresponds to the multidimensional cost of sending a packet on the related network link: each component of the weights vector represents a different QoS metric value (e.g. bandwidth, cost, delay, packet loss). The second step* *consists in writing this graph as a program in Soft Constraint Logic Programming (SCLP): the engine of this framework is then able to find the best paths/trees by optimizing their costs and solving the constraints imposed on them (e.g. delay ¡ 40msec), thus finding a solution to QoS routing problems. Moreover, c-semiring structures are a convenient tool to model QoS metrics. At last, we provide an implementation of the framework over scale-free networks and we suggest how the performance can be improved.*

9. Giorgia Lodi, Fabio Panzieri, Davide Rossi, Elisa Turrini. *SLA-Driven Clustering of QoS-Aware Application Servers.* IEEE Trans. Software Eng., Vol. 33, Num. 3, pages 186–197, 2007.

   Abstract: *In this paper, we discuss the design, implementation, and experimental evaluation of a middleware architecture for enabling service level agreement (SLA)-driven clustering of QoS-aware application servers. Our middleware architecture supports application server technologies with dynamic resource management: application servers can dynamically change the amount of clustered resources assigned to hosted applications on-demand so as to meet application-level quality of service (QoS) requirements. These requirements can include timeliness, availability, and high throughput and are specified in SLAs. A prototype of our architecture has been implemented using the open-source J2EE application server JBoss. The evaluation of this prototype shows that our approach makes possible JBoss' resource usage optimization and allows JBoss to effectively meet the QoS requirements of the applications it hosts, i.e., to honor the SLAs of those applications.*

10. Fabrizio Montesi, Claudio Guidi, Roberto Lucchi, Gianluigi Zavattaro. *JOLIE: a Java Orchestration Language Interpreter Engine.* Electr. Notes Theor. Comput. Sci., Vol. 181, pages 19–33, 2007.

    Abstract: *Service oriented computing is an emerging paradigm for programming distributed applications based on services. Services are simple software elements that supply their functionalities by exhibiting their interfaces and that can be invoked by exploiting simple communication primitives. The emerging mechanism exploited in service oriented computing for composing services -in order to provide more complex functionalities- is by means of orchestrators. An orchestrator is able to invoke and coordinate other services by exploiting typical workflow patterns such as parallel composition, sequencing and choices. Examples of orchestration languages are XLANG [IBM, "XLANG: Web Services for Business Process Design," http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm] and WS-BPEL [OASIS, "Web Services Business Process Execution Language Version 2.0, Working Draft," http://www.oasis-open.org/committees/download.php/10347/wsbpe specification-draft-120204.htm]. In this paper we present JOLIE, an interpreter and engine for orchestration programs. The main novelties of JOLIE are that it provides an easy to use development environment (because it supports a more programmer friendly C/Java-like syntax instead of an XML-based syntax) and it is based on a solid mathematical underlying model (developed in previous works of the authors [N. Busi, R. Gorrieri, C. Guidi, R. Lucchi and G. Zavattaro, Towards a formal framework for Choreography, in: Proc. of 3rd International Workshop on Distributed and Mobile Collaboration (DMC 2005) (2005), N. Busi, R. Gorrieri, C. Guidi, R. Lucchi and G. Zavattaro, Choreography and orchestration conformance for system design, in: Proc. of 8th International Conference on Coordination Models and Languages (COORDINATION'06), LNCS to appear, 2006, C. Guidi and R. Lucchi, Mobility mechanisms in service oriented computing, in: Proc. of 8th International Conference on on Formal Methods for Open Object-Based Distributed Systems (FMOODS'06), LNCS to appear, 2006]).*

11. Mike P. Papazoglou, Benedikt Kratz. *Web services technology in support of business transactions.* Service Oriented Computing and Applications, Vol. 1, Num. 1, pages 51–63, 2007.

    Abstract: *Advanced business applications typically involve well-defined business functions such as payment processing, shipping and tracking, determining new product offerings, granting/extending credit, managing market risk and so on. These reflect commonly standard business functions that apply to a variety of application scenarios. Although such business functions drive transactional applications between trading partners they are completely external to current Web services transaction mechanisms and are only expressed as part of application logic. To remedy this situation, this paper proposes a business-aware Web services transaction model and support mechanisms, which is driven by common business functions. The model allows expressing* *business functions such as payment and credit conditions, delivery conditions, business agreements stipulated in SLAs, liabilities and dispute resolution policies. It allows blending these business functions with QoS criteria such as security support to guarantee integrity of information, confidentiality, and non-repudiation.*

12. Wenhui Sun, Jinyu Zhang, Feng Liu. *WS-SLA: A Framework for Web Services Oriented Service Level Agreements.* Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference on, To Appear, May 2006.

    Abstract: *Service level agreements (SLA) becomes the prevailing business model for delivery of a large number of products and services in telecommunication and IT arenas. We presents an integrated service level agreement model applied to operating supporting system (OSS), on the basis of WfMC workflow reference model and service oriented architecture (SOA). And an application scenario instance of SLA system architecture is described by a set of operations models by Business Process Execution Language (BPEL)*

13. Christel Baier, Marjan Sirjani, Farhad Arbab, Jan J. M. M. Rutten. *Modeling component connectors in Reo by constraint automata.* Sci. Comput. Program., Vol. 61, Num. 2, pages 75–113, 2006.

    Abstract: *In this paper we introduce constraint automata and propose them as an operational model for Reo, an exogenous coordination language for compositional construction of component connectors based on a calculus of channels. By providing composition operators for constraint automata and defining notions of equivalence and refinement relations for them, this paper covers the foundations for building tools to address concerns such as the automated construction of the automaton for a given component connector, equivalence checking or containment checking of the behavior of two given connectors, and verification of coordination mechanisms.*

14. Mike P. Papazoglou, Willem-Jan van den Heuvel. *Service-oriented design and development methodology.* Int. J. Web Eng. Technol., Vol. 2, Num. 4, pages 412–442, 2006.

    Abstract: *Service Oriented Architectures (SOA) are rapidly emerging as the premier integration and architectural approach in contemporary, complex, heterogeneous computing environments. SOA is not simply about deploying software: it also requires that organisations evaluate their business models, come up with service-oriented analysis and design techniques, deployment and support plans, and carefully evaluate partner/customer/supplier relationships. Since SOA is based on open standards and is frequently realised using Web Services (WS), developing meaningful WS and business process specifications is an important requirement for SOA applications that leverage WS. Designers and developers cannot be expected to oversee a complex service-oriented development project without relying on a sound design and development methodology. This paper provides an overview of the methods and techniques used in service-oriented design and development. The aim of this paper is to examine a service development methodology from the point of view of both service producers and requesters and review the range of elements in this methodology that are available to them.*

15. Groenmo, R., Jaeger, M.C.. *Model-driven semantic Web service composition.* Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific, To Appear, December 2005.

    Abstract: *As the number of available Web services increases there is a growing demand to realise complex business processes by combining and reusing available Web services. The reuse and combination of services results in a composition of Web services that may also involve services provided in the Internet. With semantically described Web services, an automated matchmaking of capabilities can help identify suitable services. To address the need for semantically defined Web services, OWL-S and WSML have been proposed as competing semantic Web service languages. We show how the proposed semantic Web service languages can be utilized within a model-driven methodology for building composite Web services. In addition we combine the semantic-based discovery with the support for processing QoS requirements to apply a ranking or a selection of the candidates. The methodology describes a process which guides the developer through four phases, starting with the initial modelling, and ending with a new composite service that can be deployed and published to be consumed by other users.*

16. Ma, K.J.. *Web services: what's real and what's not?.* IT Professional, Vol. 7, Num. 2, pages 14–21, March 2005.

Abstract: *The idea of abstracted, well-defined, and ubiquitously invokable services replacing proprietary interprocess communications has been a goal of system designers for a long time. The rise of Web services has led to a lot of misconceptions about how they can and cannot support the holy grail of a service-oriented architecture (SOA). This article seeks to put Web services in perspective, explaining their current capabilities and what industry can expect from them in the near term. It gives an overview of how technologies such as the Extensible Markup Language (XML), XML schemas, Extensible Stylesheet Language Transformations (XSLT), the Simple Object Access Protocol (SOAP), the Web Services Description Language (WSDL), and universal description, discovery, and integration (UDDI) fit into the equation for an SOA.*

17. Wil M. P. van der Aalst. *Business alignment: using process mining as a tool for Delta analysis and conformance testing.* Requir. Eng., Vol. 10, Num. 3, pages 198–211, 2005.
Abstract: *Increasingly, business processes are being controlled and/or monitored by information systems. As a result, many business processes leave their "footprints" in transactional information systems, i.e., business events are recorded in so-called event logs. Process mining aims at improving this by providing techniques and tools for discovering process, control, data, organizational, and social structures from event logs, i.e., the basic idea of process mining is to diagnose business processes by mining event logs for knowledge. In this paper we focus on the potential use of process mining for measuring business alignment, i.e., comparing the real behavior of an information system or its users with the intended or expected behavior. We identify two ways to create and/or maintain the fit between business processes and supporting information systems: Delta analysis and conformance testing. Delta analysis compares the discovered model (i.e., an abstraction derived from the actual process) with some predefined processes model (e.g., the workflow model or reference model used to configure the system). Conformance testing attempts to quantify the "fit" between the event log and some predefined processes model. In this paper, we show that Delta analysis and conformance testing can be used to analyze business alignment as long as the actual events are logged and users have some control over the process.*

18. Schahram Dustdar, Wolfgang Schreiner. *A survey on web services composition.* IJWGS, Vol. 1, Num. 1, pages 1–30, 2005.
Abstract: *Due to the web services' heterogeneous nature, which stems from the definition of several XML-based standards to overcome platform and language dependence, web services have become an emerging and promising technology to design and build complex inter-enterprise business applications out of single web-based software components. To establish the existence of a global component market, in order to enforce extensive software reuse, service composition experienced increasing interest in doing a lot of research effort. This paper discusses the urgent need for service composition, the required technologies to perform service composition. It also presents several different composition strategies, based on some currently existing composition platforms and frameworks, re-presenting first implementations of state-of the-art technologies, and gives an outlook to essential future research work.*

19. Papazoglou, M.P., van den Heuvel, W.-J.. *Web services management: a survey.* Internet Computing, IEEE, Vol. 9, Num. 6, pages 58–64, November/December 2005.
Abstract: *Solutions based on service-oriented architectures are promising in that they leverage common services and enable collaborative business processes that cross organizational boundaries. However, because Web services applications can span multiple hosts, operating systems, languages, and enterprises, it's problematic to measure, control, and manage application availability and performance. In addition to discussing the relationship of Web services management to traditional distributed systems management, this survey explores various Web services management approaches and their underlying architectural concepts.*

20. James Pasley. *How BPEL and SOA Are Changing Web Services Development.* IEEE Internet Computing, Vol. 9, Num. 3, pages 60–67, 2005.
Abstract: *As the use of Web services grows, organizations are increasingly choosing the Business Process Execution Language for modeling business processes within the Web services architecture. In addition to orchestrating organizations' Web services, BPEL's strengths include asynchronous message handling, reliability, and recovery. By developing Web services with BPEL in mind, organizations can implement aspects of the service-oriented architecture that might previously have been difficult to achieve.*

21. Tao Yu, Kwei-Jay Lin. *Service selection algorithms for Web services with end-to-end QoS constraints.* Inf. Syst. E-Business Management, Vol. 3, Num. 2, pages 103–126, 2005.

    Abstract: *Web services are new forms of Internet software that can be universally deployed and invoked using standard protocols. Services from different providers can be integrated into a composite service regardless of their locations, platforms, and/or execution speeds to implement complex business processes and transactions. In this paper, we study the end-to-end QoS issues of composite services by utilizing a QoS broker that is responsible for selecting and coordinating the individual service component. We design the service selection algorithms used by QoS brokers to construct the optimal composite service. The objective of the algorithms is to maximize the user-defined utility function value while meeting the end-to-end delay constraint. We propose two solution approaches to the service selection problem: the combinatorial approach, by modeling the problem as the Multiple Choice Knapsack Problem (MCKP), and the graph approach, by modeling the problem as the constrained shortest path problem in the graph theory. We study efficient solutions for each approach.*

22. Neal Leavitt. *Are Web Services Finally Ready to Deliver?.* IEEE Computer, Vol. 37, Num. 11, pages 14–18, November 2004.

    Abstract: *Web services, in brief, are a framework of software technologies designed to support interoperable machine-to-machine interaction over a network. Companies on different systems can use Web services to exchange information online with business partners, customers, and suppliers. Various standards organizations and industry consortia are developing Web services specifications without a unifying authority. Organizations such as the World Wide Web Consortium (W3C), the Organization for the Advancement of Structured Information Standards (OASIS), the Liberty Alliance Project, and the Web Services Interoperability Organization (WS-I) have developed or reviewed numerous standards. A primary goal of Web services is to unlock a new generation of e-commerce applications.*

23. Xiulan Yu, Long Zhang, Ying Li, Ying Chen. *WSCE: a flexible Web Service Composition Environment.* Web Services, 2004. Proceedings. IEEE International Conference on, To Appear, July 2004.

    Abstract: *In this paper, we propose the concepts of virtual partner and inspector into the Web services composition. Virtual partner, as an IT level concept, is a Web service (pseudo Web service) using the same interface with the actual partner but different binding message. A virtual partner can be invoked directly by a business process described by BPEL, so that the BPEL programmer can test both application's functionality and non functionality performance early in the development cycle to avoid any problems in the final runtime, or test the selection of their partners in business level design. The IT virtual partners provide developers with a range of the techniques which let them explore every aspect of their program. Inspector is proposed when using the third-party process engine. An inspector itself is also a Web service. The programmer can register any required output information in it. The IT virtual partner and the inspector concepts have been integrated in our WSCE, a flexible Web Services Composition Environment for a business process. WSCE is a prototype of autonomic modeling and simulation environment. With the help of a third-party BPEL engine, it provides programmer with concepts and tools to facilitate business process programming.*

24. Liangzhao Zeng, Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.. *QoS-aware middleware for Web services composition.* Software Engineering, IEEE Transactions on, Vol. 30, Num. 5, pages 311–327, May 2004.

    Abstract: *The paradigmatic shift from a Web of manual interactions to a Web of programmatic interactions driven by Web services is creating unprecedented opportunities for the formation of online business-to-business (B2B) collaborations. In particular, the creation of value-added services by composition of existing ones is gaining a significant momentum. Since many available Web services provide overlapping or identical functionality, albeit with different quality of service (QoS), a choice needs to be made to determine which services are to participate in a given composite service. This paper presents a middleware platform which addresses the issue of selecting Web services for the purpose of their composition in a way that maximizes user satisfaction expressed as utility functions over QoS attributes, while satisfying the constraints set by the user and by the structure of the composite service. Two selection approaches are described and compared: one based on local (task-level) selection of services and the other based on global allocation of tasks to services using integer programming.*

25. Andrew W. Cooke, Alasdair J. G. Gray, Werner Nutt, James Magowan, Manfred Oevers, Paul Taylor, Roney Cordenonsi, Rob Byrom, Linda Cornwall, Abdeslem Djaoui, Laurence Field, Steve Fisher, Steve Hicks, Jason Leake, Robin Middleton, Antony J. Wilson, Xiaomei Zhu, Norbert Podhorszki, Brian A. Coghlan, Stuart Kenny, David O'Callaghan, John Ryan. *The Relational Grid Monitoring Architecture: Mediating Information about the Grid.* J. Grid Comput., Vol. 2, Num. 4, pages 323–339, 2004.

    Abstract: *We have developed and implemented the Relational Grid Monitoring Architecture (R-GMA) as part of the DataGrid project, to provide a flexible information and monitoring service for use by other middleware components and applications. R-GMA presents users with a virtual database and mediates queries posed at this database: users pose queries against a global schema and R-GMA takes responsibility for locating relevant sources and returning an answer. R-GMAs architecture and mechanisms are general and can be used wherever there is a need for publishing and querying information in a distributed environment. We discuss the requirements, design and implementation of R-GMA as deployed on the DataGrid testbed. We also describe some of the ways in which R-GMA is being used.*

26. Mariya Koshkina, Franck van Breugel. *Modelling and verifying web service orchestration by means of the concurrency workbench.* SIGSOFT Softw. Eng. Notes, Vol. 29, Num. 5, pages 1–10, ACM, 2004.

    Abstract: *Verification techniques like model checking, preorder checking and equivalence checking are shown to be relevant to web service orchestration. The Concurrency Workbench of the New Century (CWB) is a verification tool that supports these verification techniques. By means of the Process Algebra Compiler (PAC), the CWB is modified to support the BPE-calculus. The BPE-calculus is a small language, based on BPEL4WS, to express web service orchestration. Both the syntax and the semantics of the BPE-calculus are formally defined. These are subsequently used as input for the PAC. As output, the PAC produces modules that are incorporated into the CWB so that it supports the BPE-calculus and, hence, provides a verification tool for web service orchestration.*

27. Milanovic, N., Malek, M.. *Current solutions for Web service composition.* Internet Computing, IEEE, Vol. 8, Num. 6, pages 51–59, November/December 2004.

    Abstract: *Web service composition lets developers create applications on top of service-oriented computing's native description, discovery, and communication capabilities. Such applications are rapidly deployable and offer developers reuse possibilities and users seamless access to a variety of complex services. There are many existing approaches to service composition, ranging from abstract methods to those aiming to be industry standards. The authors describe four key issues for Web service composition.*

28. Marco Pistore, Marco Roveri, Paolo Busetta. *Requirements-Driven Verification of Web Services.* Electr. Notes Theor. Comput. Sci., Vol. 105, pages 95–108, 2004.

    Abstract: *We propose a requirements-driven approach to the design and verification of Web services. The proposed methodology starts from a requirements model, which defines a business domain at a "strategic" level, describing the participating actors, their mutual dependencies, goals, requirements, and expectations. This business requirements model is then refined into a business process model. In this refinement, definitions of the processes carried out by the actors of the domain are added to the model in the form of BPEL4WS code. We show how to exploit model checking techniques for the verification of the specification, both at the requirements and at the process level. At the requirements level, model checking is used to validate the specification against a set of queries specified by the designer; at the process level, it is used to verify if the BPEL4WS processes satisfy the constraints described in the requirements model.*

29. W. van der Aalst. *Don't go with the flow: Web services composition standards exposed.* IEEE Intelligent Systems, To Appear, Jan/Feb 2003.

    Abstract: *The recently released Business Process Execution Language for Web Services (BPEL4WS) is said to combine the best of other standards for web services composition such as WSFL from IBM and XLANG of Microsoft. BPEL4WS allows for a mixture of block structured and graph structured process models thus making the language expressive at the price of being complex. Although BPEL4WS is not such a bad proposal by itself, it is remarkable how much attention this standard receives while the more* *fundamental issues and problems such as semantics, expressiveness, and adequacy do not get the attention they deserve. Having a standard is a very good idea. However, there are too many of them and most of them die before becoming mature. A simple indicator of this development is the increasing length of acronyms: PDL, XPDL, BPSS, EDOC, BPML, WSDL, WSCI, ebXML, and BPEL4WS are just some of the acronyms referring to various standards in the domain. Another problem is that these languages typically have no clearly defined semantics. The only way to overcome these problems is to critically evaluate the so-called standards for web services composition, i.e., Don't go with the flow!*

30. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski, Alistair P. Barros. *Workflow Patterns.* Distributed and Parallel Databases, Vol. 14, Num. 1, pages 5–51, 2003.
    Abstract: *Differences in features supported by the various contemporary commercial workflow management systems point to different insights of suitability and different levels of expressive power. The challenge, which we undertake in this paper, is to systematically address workflow requirements, from basic to complex. Many of the more complex requirements identified, recur quite frequently in the analysis phases of workflow projects, however their implementation is uncertain in current products. Requirements for workflow languages are indicated through workflow patterns. In this context, patterns address business requirements in an imperative workflow style expression, but are removed from specific workflow languages. The paper describes a number of workflow patterns addressing what we believe identify comprehensive workflow functionality. These patterns provide the basis for an in-depth comparison of a number of commercially availablework flow management systems. As such, this paper can be seen as the academic response to evaluations made by prestigious consulting companies. Typically, these evaluations hardly consider the workflow modeling language and routing capabilities, and focus more on the purely technical and commercial aspects.*

31. Gian Luigi Ferrari, Stefania Gnesi, Ugo Montanari, Marco Pistore. *A model-checking verification environment for mobile processes.* ACM Trans. Softw. Eng. Methodol., Vol. 12, Num. 4, pages 440–473, 2003.
    Abstract: *This article presents a semantic-based environment for reasoning about the behavior of mobile systems. The verification environment, called HAL, exploits a novel automata-like model that allows finite-state verification of systems specified in the π-calculus. The HAL system is able to interface with several efficient toolkits (e.g. model-checkers) to determine whether or not certain properties hold for a given specification. We report experimental results on some case studies.*

32. Xavier Franch, Juan Pablo Carvallo. *Using Quality Models in Software Package Selection.* IEEE Software, Vol. 20, Num. 1, pages 34–41, 2003.
    Abstract: *The growing importance of commercial off-the-shelf software packages requires adapting some software engineering practices, such as requirements elicitation and testing, to this emergent framework. Also, some specific new activities arise, among which selection of software packages plays a prominent role. All the methodologies that have been proposed recently for choosing software packages compare user requirements with the packages' capabilities. There are different types of requirements, such as managerial, political, and, of course, quality requirements. Quality requirements are often difficult to check. This is partly due to their nature, but there is another reason that can be mitigated, namely the lack of structured and widespread descriptions of package domains (that is, categories of software packages such as ERP systems, graphical or data structure libraries, and so on). This absence hampers the accurate description of software packages and the precise statement of quality requirements, and consequently overall package selection and confidence in the result of the process. Our methodology for building structured quality models helps solve this drawback.*

33. Alexander Keller, Heiko Ludwig. *The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services.* J. Network Syst. Manage., Vol. 11, Num. 1, 2003.
    Abstract: *We describe a novel framework for specifying and monitoring Service Level Agreements (SLA) for Web Services. SLA monitoring and enforcement becomes increasingly important in a Web Service environment where enterprises rely on services that may be subscribed dynamically and on demand. For economic and practical reasons, we want an automated provisioning process for both the service itself as well as the SLA management system. It measures and monitors the QoS parameters, checks the agreed-upon*

service levels, and reports violations to the authorized parties involved in the SLA management process. The Web Service Level Agreement (WSLA) framework, our approach to these issues, is targeted at defining and monitoring SLAs for Web Services. Although WSLA has been designed for a Web Services environment, it is applicable as well to any inter-domain management scenario such as business process and service management or the management of networks, systems and applications in general. The WSLA framework consists of a flexible and extensible language based on XML Schema and a runtime architecture comprising several SLA monitoring services, which may be outsourced to third parties to ensure a maximum of accuracy. WSLA enables service customers and providers to unambiguously define a wide variety of SLAs, specify the SLA parameters and the way how they are measured, and relate them to managed resource instrumentations. Upon receipt of an SLA specification, the WSLA monitoring services are automatically configured to enforce the SLA. An implementation of the WSLA framework, the SLA Compliance Monitor, is publicly available as part of the IBM Web Services Toolkit.

34. Shuping Ran. *A model for web services discovery with QoS.* SIGecom Exch., Vol. 4, Num. 1, pages 1–10, ACM, 2003.

    Abstract: *Web services technology has generated a lot interest, but its adoption rate has been slow. This paper discusses issues related to this slow take up and argues that quality of services is one of the contributing factors. The paper proposes a new Web services discovery model in which the functional and non-functional requirements (i.e. quality of services) are taken into account for the service discovery. The proposed model should give Web services consumers some confidence about the quality of service of the discovered Web services.*

35. Menasce, D.A.. *QoS Issues in Web Services.* Internet Computing, IEEE, Vol. 6, Num. 6, pages 72–75, November/December 2002.

    Abstract: *Quality of service (QoS) is a combination of several qualities or properties of a service, such as: availability is the percentage of time that a service is operating; security properties include the existence and type of authentication mechanisms the service offers, confidentiality and data integrity of messages exchanged, nonrepudiation of requests or messages, and resilience to denial-of-service attacks; response time is the time a service takes to respond to various types of requests; Response time is a function of load intensity, which can be measured in terms of arrival rates (such as requests per second) or number of concurrent requests. QoS takes into account not only the average response time, but also the percentile of the response time; and throughput is the rate at which a service can process requests. QoS measures can include the maximum throughput or a function that describes how throughput varies with load intensity. The QoS measure is observed by Web services users. These users are not human beings but programs that send requests for services to Web service providers. QoS issues in Web services have to be evaluated from the perspective of the providers of Web services and from the perspective of the users of these services.*

36. Nicholas R. Jennings, Peyman Faratin, A. R. Lomuscio, Simon Parsons, Michael Wooldridge, Carles Sierra. *Automated Negotiation: Prospects Methods and Challenges.* Group Decision and Negotiation, To Appear, 2001.

37. Frank E. Ritter, Richard M. Young. *Embodied models as simulated users: introduction to this special issue on using cognitive models to improve interface design.* Int. J. Hum.-Comput. Stud., Vol. 55, Num. 1, pages 1–14, 2001.

    Abstract: *Cognitive models provide a means for applying what is known from psychology to the design of interfaces, thereby improving their quality and usability. Existing uses of models include predicting time and errors for users to perform tasks, acting as embedded assistants to help users perform their tasks, and serving as surrogate users. Treating the design of human-computer interfaces as a form of engineering design requires the development and application of user models. A recent trend is for models to be built within the fixed framework of a cognitive architecture, which has been extended by the addition of simulated eyes and hands, enabling the construction of embodied models. Being embodied allows models to interact directly with interfaces. The resulting models can be used to evaluate the interfaces they use, and serve as explanations of users' behavior. The papers in this Special Issue point to a new route for the future, one in which models built within embodied cognitive architectures provide information for the design of better interfaces.*

38. George A. Papadopoulos, Farhad Arbab. *Coordination Models and Languages.* Advances in Computers, Vol. 46, pages 330–401, 1998.

Abstract: *A new class of models, formalisms and mechanisms has recently evolved for describing concurrent and distributed computations based on the concept of "coordination". The purpose of a coordination model and associated language is to provide a means of integrating a number of possibly heterogeneous components together, by interfacing with each component in such a way that the collective set forms a single application that can execute on and take advantage of parallel and distributed systems. In this chapter we initially define and present in sufficient detail the fundamental concepts of what constitutes a coordination model or language. We then go on to classify these models and languages as either "data-driven" or "control-driven" (also called "process-" or "task-oriented"). Next, the main existing coordination models and languages are described in sufficient detail to let the reader appreciate their features and put them into perspective with respect to each other. The chapter ends with a discussion comparing the various models and some conclusions.*

Articles in Refereed Conferences:

1. Sheng Chen, Liang Bao, Ping Chen. *OptBPEL: A Tool for Performance Optimization of BPEL Process.* Software Composition, pages 141–148, 2008.

Abstract: *The Business Process Execution Language (BPEL) is now a de facto standard for specifying and executing business process for web service composition and orchestration. As more and more web services are composed using BPEL, tuning these compositions and gain better performance becomes increasingly important. This paper presents our approach for optimizing the BPEL process and introduces OptBPEL, a tool for performance optimization of BPEL process. The approach starts from the optimization of synchronization structure concerning link in BPEL. After that, some concurrency analysis techniques are applied to obtain further performance improvement. Finally, we give some experiments and prove the efficiency of these optimization algorithms used in OptBPEL.*

2. Kyriakos Kritikos, Dimitris Plexousakis. *Evaluation of QoS-Based Web Service Matchmaking Algorithms.* International Conference on Services Computing, 2008.

Abstract: *Web Service (WS) discovery is a prerequisite for achieving WS composition and orchestration. Although a lot of esearch has been conducted on the functional discovery of WSs, the proposed techniques fall short when faced with the foreseen increase in the number of (potentially functionally- equivalent) WSs. The above situation can be resolved with the addition of non-functional (Quality of Service (QoS)) discovery mechanisms to WS discovery engines. QoS-based WS matchmaking algorithms have been devised for this reason. However, they are either slow - as they are based on ontology reasoners - or produce inaccurate results. Inaccuracy is caused both by the syntactic matching of QoS concepts and by wrong matchmaking metrics. In this paper, we present two Constraint Programming (CP) QoS-based WS discovery algorithms for unary constrained WS specifications that produce accurate results with good performance. We also evaluate these algorithms on matchmaking time, precision and recall in different settings in order to demonstrate their efficiency and accuracy.*

3. Kyriakos Kritikos, Dimitris Plexousakis. *OWL-Q for Semantic QoS-based Web Service Description and Discovery.* First International Joint Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, November 2007.

Abstract: *Semantic Web Services are emerging for their promise to produce a more accurate and precise Web Service discovery process. However, most of research approaches focus only on the functional part of semantic Web Service description. The above fact along with the proliferation of Web Services is highly probable to lead to a situation where Web Service registries will return many functionally-equivalent Web Service advertisements for each user request. This problem can be solved with the semantic description of QoS for Web Services. QoS is a set of non-functional properties encompassing performance and network-related characteristics of resources. So it can be used for distinguishing between functionally-equivalent Web Services. Current research approaches for QoS-based Web Service description are either syntactic or poor or non-extensible. To solve this problem, we have developed a rich and extensible ontological specification called OWL-Q for semantic QoS-based Web Service description. We analyze all OWL-Q parts and reason*

that rules should be added in order to support property inferencing and constraint enforcement. Finally, we line out our under-development semantic framework for QoS-based Web Service description and discovery.

4. Cinzia Cappiello, Marco Comuzzi, Pierluigi Pleban. *On Automated Generation of Web Service Level Agreements .* Advanced Information Systems Engineering, Vol. 4495, Springer Verlag, June 2007.

   Abstract: *Before a service invocation takes place, an agreement between the service provider and the service user might be required. Such an agreement is the result of a negotiation process between the two parties and defines how the service invocation has to occur. Considering the Service Oriented Computing paradigm, the relationship among providers and users is extremely loose. Traditional agreements are likely to concern long term relationships and to be manually performed. In this paper, we propose a model to generate service level agreement on-the-fly. Just before the invocation commences, the quality of the service is negotiated in order to generate a service level agreement tied to that specific invocation. Such an approach relies on a quality model that supports both users requirements and providers capabilities definition.*

5. Jonatha Anselmi, Danilo Ardagna, Paolo Cremonesi. *A QoS-based selection approach of autonomic grid services.* SOCP '07: Proceedings of the 2007 workshop on Service-oriented computing performance: aspects, issues, and approaches, pages 1–8, ACM, 2007.

   Abstract: *The Web service composition (WSC) is the process of building an instance of an abstract workflow by combining appropriate Web services that satisfies given QoS requirements. In general, QoS requirements consists of a number of constraints. The selection process requires global optimization and can be formalized as a mixed integer linear programming problem which cannot be solved in polynomial time. However, since the number of submitted workflows is large and the QoS is highly dynamic, the fast selection of composite Web Services is particularly important. In this paper, we present a QoS broker-based framework for Web services execution in autonomic grid environments. The main goal of the framework is to support the broker in selecting Web services based on the required QoS. To achieve this goal, we propose a novel approach: since successive composed Web services requests can have the same task to Web service assignment, we address the Multiple Instance WSC (MI-WSC) problem optimizing simultaneously the set of requests which will be submitted to the system in the successive time interval instead of independently computing a solution for each incoming request. Experimental results show that the proposed algorithm has better performance with respect to existing techniques. Moreover, the qualities of the selected composite Web services are not significantly different from the optimal ones.*

6. Gero Decker, Oliver Kopp, Frank Leymann, Mathias Weske. *BPEL4Chor: Extending BPEL for Modeling Choreographies.* ICWS, pages 296–303, 2007.

   Abstract: *The Business Process Execution Language (BPEL) is a language to orchestrate web services into a single business process. In a choreography view, several processes are interconnected and their interaction behavior is described from a global perspective. This paper shows how BPEL can be extended for defining choreographies. The proposed extensions (BPEL4Chor) distinguish between three aspects: (i) participant behavior descriptions, i.e. control flow dependencies in each participant, (ii) the participant topology, i.e. the existing participants and their interconnection using message links and (iii) participant groundings, i.e. concrete configurations for data formats and port types. As BPEL itself is used unchanged, the extensions facilitate a seamless integration between service choreographies and orchestrations. The suitability of the extensions is validated by assessing their support for the Service Interaction Patterns.*

7. Ester Giallonardo, Eugenio Zimeo. *More Semantics in QoS Matching.* SOCA, pages 163–171, 2007.

   Abstract: *The evolution of the Web towards a global computing environment is promoting new research efforts aimed at the formal characterization of Web Services QoS. Reasoning on QoS is a key to improve matching process during the discovery of desired services and a step towards the transformation of applications in collections of loosely coupled services virtually connected by semantic similarities. The paper presents the on QoS ontology, an openly available OWL ontology for QoS, and evaluates it in a QoS-aware matching environment. The ontology can be used to express functions of QoS metrics useful to improve the recall tied to the matching of a template request with target Web Services. To this end, the ontology* *introduces the concept of derivation in the matching process. This gives the possibility of matching a QoS template with published Web Services by deriving different QoS parameters when a one-to-one matching fails. The proposed matching algorithm utilizes a reasoner that exploits the ontology to avoid apparent mismatches. An experimental evaluation shows that exploiting QoS knowledge significantly improves matching recall without deteriorating precision.*

8. Dimka Karastoyanova, Branimir Wetzstein, Tammo van Lessen, Daniel Wutke, Jörg Nitzsche, Frank Leymann. *Semantic Service Bus: Architecture and Implementation of a Next Generation Middleware.* ICDE Workshops, pages 347–354, 2007.

   Abstract: *In this paper we present a middleware for the service oriented architecture, called the Semantic Service Bus. It is an advanced middleware possessing enhanced features, as compared to the conventional service buses. It is distinguished by the fact that it uses semantic description of service capabilities, and requirements towards services to enable more elaborate service discovery, selection, routing, composition and data mediation. The contributions of the paper are the conceptual architecture of the Semantic Service Bus and a prototypical implementation supporting different semantic Web service technologies (OWL-S and WSMO) and conventional Web services. Since mission critical application scenarios (for SOA) involve complex orchestrations of services, we have chosen to utilize semantically annotated service orchestrations as the applications to employ this middleware.*

9. Kyriakos Kritikos, Dimitris Plexousakis. *Requirements for QoS-based Web Service Description and Discovery.* COMPSAC (2), pages 467–472, 2007.

   Abstract: *The goal of Service Oriented Architectures is to enable the creation of business applications through the automatic discovery and composition of independently developed and deployed (Web) services. Automatic discovery of Web Services (WSs) can be achieved by incorporating semantics into a richer WS description model (WSDM) and the use of Semantic Web (SW) technologies in the WS matchmaking and selection models. A sufficiently rich WSDM should encompass not only functional but also non-functional aspects like Quality of Service (QoS). QoS is a set of performance attributes that has a substantial impact on WS requesters' expectations. Thus, it can be used as a discriminating factor of functionally-equivalent WSs. The focus of this paper is twofold: to analyze the requirements of a semantically rich QoS-based WSDM and to provide SW and constrainedbased mechanisms for enriching syntactic QoS-based WS Discovery (WSDi) algorithms. In addition, a roadmap of extending WS standard techniques for realizing semantic, functional and QoS-based WSDi is presented.*

10. Kyriakos Kritikos, Dimitris Plexousakis. *Semantic QoS-based Web Service Discovery Algorithms.* Fifth European Conference on Web Services, pages 181–190, IEEE Computer Society, 2007.

    Abstract: *The success of the Web Service (WS) paradigm has led to a proliferation of available WSs, which are advertised in WS registries. While sophisticated semantic WS discovery algorithms are operating on these registries to return matchmaking results with high precision and recall, many functionally-equivalent WSs are returned. The solution to the above problem comes in terms of semantic QoS-based description and discovery of WSs. We have already presented a rich and extensible ontology language for QoS-based WS description called OWL-Q. We have also proposed a semantic QoS metric matching algorithm. Based on this algorithm, we have extended a CSP-based approach for QoS-based WS discovery. In this paper, we firstly analyze the evolution of OWL-Q and its extension with SWRL rules, we propose a modification to the metric matching algorithm and we show the way the metric alignment process takes place. Then we propose two novel semantic QoS-based WS Discovery algorithms that return matches even for over-constrained QoS-based WS requests. The first one deals with unary constraints while the second one is more generic. Finally, implementation aspects of our QoS-based WS discovery system are discussed.*

11. Anca Muscholl, Igor Walukiewicz. *A Lower Bound on Web Services Composition.* FoSSaCS, pages 274–286, 2007.

    Abstract: *A web service is modeled here as a finite state machine. A composition problem for web services is to decide if a given web service can be constructed from a given set of web services; where the construction is understood as a simulation of the specification by a fully asynchronous product of the given services. We* *show an EXPTIME-lower bound for this problem, thus matching the known upper bound. Our result also applies to richer models of web services, such as the Roman model.*

12. Elisabetta Di Nitto, Massimiliano Di Penta, Alessio Gambi, Gianluca Ripa, Maria Luisa Villani. *Negotiation of Service Level Agreements: An Architecture and a Search-Based Approach.* ICSOC, pages 295–306, 2007.

    Abstract: *Software systems built by composing existing services are more and more capturing the interest of researchers and practitioners. The envisaged long term scenario is that services, offered by some competing providers, are chosen by some consumers and used for their own purpose, possibly, in conjunction with other services. In the case the consumer is not anymore satisfied by the performance of some service, he can try to replace it with some other service. This implies the creation of a global market of services and poses new requirements concerning validation of exploited services, security of transactions engaged with services, trustworthiness, creation and negotiation of Service Level Agreements with these services. In this paper we focus on the last aspect and present our approach for negotiation of Service Level Agreements. Our architecture supports the actuation of various negotiation processes and offers a search-based algorithm to assist the negotiating parts in the achievement of an agreement.*

13. Qing Zhu, Shan Wang, Guorong Li, Guangqiang Liu, Xiaoyong Du. *QoS-Based Services Selecting and Optimizing Algorithms on Grid.* APWeb/WAIM Workshops, pages 156–167, 2007.

    Abstract: *QoS-Based Services Selecting and Optimizing Composition between the peers play an increasingly important role to ensure interoperability on Grid environment. However, the prohibitive cost of selecting, matching, mapping and composing algorithm has now become a key bottleneck hindering the deployment of a wide variety of Grid services. In this paper, we present QoS-Based Services Selecting and Optimizing Composition on Grid. First, it checks requesters' semantic in order to form candidate service graph. Second, it designs service selecting and mapping algorithms for optimizing the model. Third, it creates an executed plan of optimum composition on Grid. We conducted experiments to simulate and evaluate our approach.*

14. A. Rozinat, W.M.P. van der Aalst. *Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models.* Business Process Management Workshops, Lecture Notes in Computer Science, Vol. 3812, pages 163–176, February 2006.

    Abstract: *Most information systems log events (e.g., transaction logs, audit trails) to audit and monitor the processes they support. At the same time, many of these processes have been explicitly modeled. For example, SAP R/3 logs events in transaction logs and there are EPCs (Event-driven Process Chains) describing the so-called reference models. These reference models describe how the system should be used. The coexistence of event logs and process models raises an interesting question: "Does the event log conform to the process model and vice versa?". This paper demonstrates that there is not a simple answer to this question. To tackle the problem, we distinguish two dimensions of conformance: fitness (the event log may be the result of the process modeled) and appropriateness (the model is a likely candidate from a structural and behavioral point of view). Different metrics have been defined and a Conformance Checker has been implemented within the ProM Framework.*

15. Wil M. P. van der Aalst, Marlon Dumas, Chun Ouyang, Anne Rozinat, H. M. W. Verbeek. *Choreography Conformance Checking: An Approach based on BPEL and Petri Nets.* The Role of Business Processes in Service Oriented Architectures, 2006.

    Abstract: *Recently, languages such as BPEL and CDL have been proposed to describe the way services can interact from a behavioral perspective. 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 paper investigates a possible usage of behavioral service descriptions, namely as input for conformance checking. Conformance checking is the act of verifying whether one or more parties stick to the agreed-upon behavior by observing the actual behavior, e.g., the exchange of messages between all parties. This paper shows that it is possible to translate BPEL business protocols to Petri nets and to relate SOAP messages to transitions in the Petri net. As a result, Petri net-based conformance checking techniques can be used to quantify fitness (whether the observed behavior is possible in the business* *protocol) and appropriateness (whether the observed behavior "overfits" or "underfits" the business protocol). Moreover, non-conformance can be visualized to pinpoint deviations. The approach has been implemented in the context of the ProM n framework.*

16. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, Marco Montali, Sergio Storari, Paolo Torroni. *Computational Logic for Run-Time Verification of Web Services Choreographies: Exploiting the OCS-SI Tool.* WS-FM, pages 58–72, 2006.

    Abstract: *In this work, we investigate the feasibility of using a framework based on computational logic, and mainly defined in the context of Multi-Agent Systems for Global Computing (SOCS UE Project), for modeling choreographies of Web Services with respect to the conversational aspect. One of the fundamental motivations of using computational logic, beside its declarative and highly expressive nature, is given by its operational counterpart, that can provide a proof-theoretic framework able to verify the consistency of services designed in a cooperative and incremental manner. In particular, in this paper we show that suitable "Social Integrity Constraints", introduced in the SOCS social model, can be used for specifying global protocols at the choreography level. In this way, we can use a suitable tool, derived from the proof-procedure defined in the context of the SOCS project, to check at run-time whether a set of existing services behave in a conformant manner w.r.t. the defined choreography.*

17. Fabio Barbon, Paolo Traverso, Marco Pistore, Michele Trainotti. *Run-Time Monitoring of Instances and Classes of Web Service Compositions.* ICWS, pages 63–71, 2006.

    Abstract: *The run-time monitoring of web service compositions has been widely acknowledged as a significant and challenging problem. In this paper, we propose a novel solution to the problem of monitoring web services implemented in BPEL. We devise an architecture that clearly separates the business logic of a web service from its monitoring functionality. The architecture supports both "instance monitors" that deal with the execution of a single instance of BPEL process, as well as "class monitors" that report aggregated information about all the instances of a BPEL process. We also define a language for the specification of instance and class monitors. The language allows for specifying boolean, statistic, and time-related properties. Finally, we devise a technique for the automatic translation of all these kinds of monitors to Java programs.*

18. Yan-ping Chen, Zeng-zhi Li, Qin-xue Jin, Chuang Wang. *Study on QoS Driven Web Services Composition.* Frontiers of WWW Research and Development - APWeb 2006, Lecture Notes on Computer Science, Vol. 3841, pages 702–707, Springer Verlag, 2006.

    Abstract: *Providing composed Web Services based on the QoS requirements of clients is still an urgent problem to be solved. In this paper, we try to solve this problem. Firstly, we enhanced the current WSDL to describe the QoS of services, and then gave a way to choose the proper pre-exist services based on their QoS.*

19. Mohan Baruwal Chhetri, Jian Lin, SukKeong Goh, Jian Ying Zhang, Ryszard Kowalczyk, Jun Yan. *A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation ofWeb Service Composition.* ASWEC, pages 90–99, 2006.

    Abstract: *Recent progress in the field of Web services has made it possible to integrate inter-organizational and heterogeneous services on the Web at runtime. If a user request cannot be satisfied by a single Web service, it is (or should be) possible to combine existing services in order to fulfill the request. However, there are several challenging issues that need to be addressed before this can be realized in the true sense. One of them is the ability to ensure end-to-end QoS of a Web service composition. There is a need for a SLA negotiation system which can ensure the autonomous QoS negotiation of Web service compositions irrespective of the application domain. In this paper we propose agent-based coordinated-negotiation architecture to ensure collective functionality, end-to-end QoS and the stateful coordination of complex services. We describe a prototype implementation to demonstrate how this architecture can be used in different application domains. We have also demonstrated how the negotiation system on the service provider's side can be implemented both as an agent based negotiation system and as a Web service based negotiation system.*

20. Gero Decker, Johannes Maria Zaha, Marlon Dumas. *Execution Semantics for Service Choreographies.* WS-FM, pages 163–177, 2006.

Abstract: *A service choreography is a model of interactions in which a set of services engage to achieve a goal, seen from the perspective of an ideal observer that records all messages exchanged between these services. Choreographies have been put forward as a starting point for building service-oriented systems since they provide a global picture of the system's behavior. In previous work we presented a language for service choreography modeling targeting the early phases of the development lifecycle. This paper provides an execution semantics for this language in terms of a mapping to PI-calculus. This formal semantics provides a basis for analyzing choreographies. The paper reports on experiences using the semantics to detect unreachable interactions.*

21. Howard Foster, Sebastian Uchitel, Jeff Magee, Jeff Kramer. *LTSA-WS: a tool for model-based verification of web service compositions and choreography.* ICSE '06: Proceedings of the 28th international conference on Software engineering, pages 771–774, ACM, 2006.
    Abstract: *In this paper we describe a tool for a model-based approach to verifying compositions of web service implementations. The tool supports verification of properties created from design specifications and implementation models to confirm expected results from the viewpoints of both the designer and implementer. Scenarios are modeled in UML, in the form of Message Sequence Charts (MSCs), and then compiled into the Finite State Process (FSP) process algebra to concisely model the required behavior. BPEL4WS implementations are mechanically translated to FSP to allow an equivalence trace verification process to be performed. By providing early design verification and validation, the implementation, testing and deployment of web service compositions can be eased through the understanding of the behavior exhibited by the composition. The approach is implemented as a plug-in for the Eclipse development environment providing cooperating tools for specification, formal modeling, verification and validation of the composition process.*

22. Kyriakos Kritikos, Dimitris Plexousakis. *Semantic QoS Metric Matching.* ECOWS, pages 265–274, 2006.
    Abstract: *As the Web Service paradigm gains popularity for its promise to transform the way business is conducted, the number of deployed Web Services grows with a fast rate. While sophisticated semantic discovery mechanisms have been invented to overcome the UDDI's syntactic discovery solution in order to provide more recallable and precise results, the amount of functionally equivalent Web Services returned is still large. The solution to this problem is the description of the QoS non-functional aspect of Web Services. QoS encompasses the performance of Web Services and can be used as a discriminator factor for refining Web Service advertisement result lists. However, most scientific efforts presented so far are purely syntactic and are not capturing all aspects of QoS-based Web Service description leading to imprecise syntactic discovery mechanisms. This paper presents a novel, rich and extensible ontology-based approach for describing QoS of Web Services that complements OWL-S. It is shown that, by using this approach and by introducing the concept of semantic QoS metric matching, QoS-based syntactic matchmaking and selection algorithms are transformed to semantic ones leading to better results.*

23. Marina Mongiello, Daniela Castelluccia. *Modelling and verification of BPEL business processes.* Fourth Workshop on Model-Based Development of Computer-Based Systems, 2006.
    Abstract: *A business process is a complex Web service with functions provided by different Web services, which are already existing in Web and are dynamically integrated for granting a more complex business task. For this reason, business processes have become more and more diffuse in B2B and B2C domains, so that the importance of their activities asks for a high-level of reliability. Methods and tools for supporting automatic system verification and validation could be useful. Among the techniques of automatic verification, we choose model checking method, because we applied it efficiently for verification of a single Web service and in this paper we extend the area of application also in business processes. Descriptions of the behavior of a business process are coded using a standard language, BPEL4WS, that has broadly spread because it is able to describe a business process as both an executable process and an abstract process. Therefore, we model a BPEL description of a generic business process with a formal model and we formalize correctness properties about the reliability of the business process design. Also, we build a framework that performs automatic verification of formal models of business processes through NuSMV model checker. If there is a violation of correctness specifications, NuSMV provides counter-examples, so we can locate errors and effect right changes for correcting business process design.*

24. Nicole Oldham, Kunal Verma, Amit Sheth, Farshad Hakimpour. *Semantic WS-agreement partner selection.* WWW '06: Proceedings of the 15th international conference on World Wide Web, pages 697–706, ACM, 2006.

    Abstract: *In a dynamic service oriented environment it is desirable for service consumers and providers to offer and obtain guarantees regarding their capabilities and requirements. WS-Agreement defines a language and protocol for establishing agreements between two parties. The agreements are complex and expressive to the extent that the manual matching of these agreements would be expensive both in time and resources. It is essential to develop a method for matching agreements automatically. This work presents the framework and implementation of an innovative tool for the matching providers and consumers based on WS-Agreements. The approach utilizes Semantic Web technologies to achieve rich and accurate matches. A key feature is the novel and flexible approach for achieving user personalized matches.*

25. Mohsen Rouached, Olivier Perrin, Claude Godart. *Towards Formal Verification of Web Service Composition.* Business Process Management, pages 257–273, 2006.

    Abstract: *Web services composition is an emerging paradigm for enabling application integration within and across organizational boundaries. Current Web services composition proposals, such as BPML, WS-BPEL, WSCI, and OWL-S, provide solutions for describing the control and data flows in Web service composition. However, such proposals remain at the descriptive level, without providing any kind of mechanisms or tool support for analysis and verification. Therefore, there is a growing interest for the verification techniques which enable designers to test and repair design errors even before actual running of the service, or allow designers to detect erroneous properties and formally verify whether the service process design does have certain desired properties. In this paper, we propose to verify Web services composition using an event driven approach. We assume Web services that are coordinated by a composition process expressed in WSBPEL and we use Event Calculus to specify the properties and requirements to be monitored.*

26. Yuan Yuan, Zhongjie Li, Wei Sun. *A Graph-Search Based Approach to BPEL4WS Test Generation.* ICSEA, 14 pages, 2006.

    Abstract: *Business Process Execution Language for Web Services (BPEL4WS) is a kind of concurrent programming languages with several special features that raise special challenges for verification and testing. This paper proposes a graph-search based approach to BPEL test case generation, which effectively deals with BPEL concurrency semantics. This approach defines an extension of CFG (Control Flow Graph) - BPEL Flow Graph (BFG) - to represent a BPEL program in a graphical model. Then concurrent test paths can be generated by traversing the BFG model, and test data for each path can be generated using a constraint solving method. Finally test paths and data are combined into complete test cases.*

27. Johannes Maria Zaha, Alistair P. Barros, Marlon Dumas, Arthur H. M. ter Hofstede. *Let's Dance: A Language for Service Behavior Modeling.* OTM Conferences (1), pages 145–162, 2006.

    Abstract: *In Service-Oriented Architectures (SOAs), software systems are decomposed into independent units, namely services, that interact with one another through message exchanges. To promote reuse and evolvability, these interactions are explicitly described right from the early phases of the development lifecycle. Up to now, emphasis has been placed on capturing structural aspects of service interactions. Gradually though, the description of behavioral dependencies between service interactions is gaining increasing attention as a means to push forward the SOA vision. This paper deals with the description of these behavioral dependencies during the analysis and design phases. The paper outlines a set of requirements that a language for modeling service interactions at this level should fulfill, and proposes a language whose design is driven by these requirements.*

28. Huang, H., Tsai, W.-T., Paul, R.. *Automated model checking and testing for composite Web services.* pages 300–307, May 2005.

    Abstract: *Web services form a new distributed computing paradigm. Collaborative verification and validation are important when Web services from different vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web services by model checking the process model of OWL-S (Web ontology language for Web services) and to validate them by the test cases automatically* *generated in the model checking process. We extend the BLAST, a model checker that handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

29. Rocco De Nicola, Gianluigi Ferrari, Ugo Montanari, Rosario Pugliese, Emilio Tuosto. *A Basic Calculus for Modelling Service Level Agreements.* International Conference on Coordination Models and Languages, LNCS, Vol. 3454, pages 33–48, Springer Verlag, April 2005.

    Abstract: *The definition of suitable abstractions and models for identifying, understanding and managing Quality of Service (QoS) constraints is a challenging issue of the Service Oriented Computing paradigm. In this paper we introduce a process calculus where QoS attributes are first class objects. We identify a minimal set of primitives that allow capturing in an abstract way the ability to control and coordinate services in presence of QoS constraints.*

30. Bernd-Holger Schlingloff, Axel Martens, Karsten Schmidt. *Modeling and Model Checking Web Services.* Vol. 126, pages 3–26, Elsevier, March 2005.

    Abstract: *We give an overview on web services and the web service technology stack. We then show how to build Petri net models of web services formulated in the specification language BPEL4WS. We define an abstract correctness criterion for these models and study the automated verification according to this criterion. Finally, we relate correctness of web service models to the model checking problem for alternating temporal logics.*

31. Luciano Baresi, Sam Guinea. *Towards Dynamic Monitoring of WS-BPEL Processes.* ICSOC, pages 269–282, 2005.

    Abstract: *The intrinsic flexibility and dynamism of service-centric applications preclude their pre-release validation and demand for suitable probes to monitor their behavior at run-time. Probes must be suitably activated and deactivated according to the context in which the application is executed, but also according to the confidence we get on its quality. The paper supports the idea that significant data may come from very different sources and probes must be able to accommodate all of them. The paper presents: (1) an approach to specify monitoring directives, called monitoring rules, and weave them dynamically into the process they belong to; (2) a proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time; (3) a user-oriented language to integrate data acquisition and analysis into monitoring rules.*

32. Alistair P. Barros, Marlon Dumas, Arthur H. M. ter Hofstede. *Service Interaction Patterns.* Business Process Management, pages 302–318, 2005.

    Abstract: *With increased sophistication and standardization of modeling languages and execution platforms supporting business process management (BPM) across traditional boundaries, has come the need for consolidated insights into their exploitation from a business perspective. Key technology developments in BPM bear this out, with several web services-related initiatives investing significant effort in the collection of compelling use cases to heighten the exploitation of BPM in multi-party collaborative environments. In this setting, we present a collection of patterns of service interactions which allow emerging web services functionality, especially that pertaining to choreography and orchestration, to be benchmarked against abstracted forms of representative scenarios. Beyond bilateral interactions, these patterns cover multilateral, competing, atomic and causally related interactions. Issues related to the implementation of these patterns using established and emerging web services standards, most notably BPEL, are discussed.*

33. Aysu Betin-Can, Tevfik Bultan, Xiang Fu. *Design for verification for asynchronously communicating Web services.* WWW, pages 750–759, 2005.

    Abstract: *We present a design for verification approach to developing reliable web services. We focus on composite web services which consist of asynchronously communicating peers. Our goal is to automatically verify properties of interactions among such peers. We propose a design pattern that eases the development of such web services and enables a modular, assume-guarantee style verification strategy. In the proposed design pattern, each peer is associated with a behavioral interface description which specifies how that peer* *will interact with other peers. Using these peer interfaces we automatically generate BPEL specifications to publish for interoperability. Assuming that the participating peers behave according to their interfaces, we verify safety and liveness properties about the global behavior of the composite web service during behavior verification. During interface verification, we check that each peer implementation conforms to its interface. Using the modularity in the proposed design pattern, we are able to perform the interface verification of each peer and the behavior verification as separate steps. Our experiments show that, using this modular approach, one can automatically and efficiently verify web service implementations.*

34. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani. *An approach for QoS-aware service composition based on genetic algorithms.* GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, pages 1069–1075, ACM, 2005.

    Abstract: *Web services are rapidly changing the landscape of software engineering. One of the most interesting challenges introduced by web services is represented by Quality Of Service (QoS)–aware composition and late–binding. This allows to bind, at run–time, a service–oriented system with a set of services that, among those providing the required features, meet some non–functional constraints, and optimize criteria such as the overall cost or response time. In other words, QoS–aware composition can be modeled as an optimization problem.We propose to adopt Genetic Algorithms to this aim. Genetic Algorithms, while being slower than integer programming, represent a more scalable choice, and are more suitable to handle generic QoS attributes. The paper describes our approach and its applicability, advantages and weaknesses, discussing results of some numerical simulations.*

35. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani. *QoS-Aware Replanning of Composite Web Services.* ICWS '05: Proceedings of the IEEE International Conference on Web Services, pages 121–129, IEEE Computer Society, 2005.

    Abstract: *Run-time service discovery and late-binding constitute some ofthe most challenging issues of service-oriented software engineering.For late-binding to be effective in the case of composite services,a QoS-aware composition mechanism is needed. This meansdetermining the set of services that, once composed, not only willperform the required functionality, but also will best contribute toachieve the level of QoS promised in Service Level Agreements (SLAs). However, QoS-aware composition relies on estimated QoS values and workflow execution paths previously obtained using a monitoring mechanism. At run-time, the actual QoS valuesmay deviate from the estimations, or the execution path may not bethe one foreseen. These changes could increase the risk of breakingSLAs and obtaining a poor QoS. Such a risk could be avoided byreplanning the service bindings of the workflow slice still to be executed. This paper proposes an approach to trigger and perform compositeservice replanning during execution. An evaluation has been performedsimulating execution and replanning on a set of composite serviceworkflows.*

36. Samuele Carpineti, Cosimo Laneve, Paolo Milazzo. *BoPi - A Distributed Machine for Experimenting Web Services Technologies.* ACSD, pages 202–211, 2005.

    Abstract: *BoPi is a programming language with a runtime support that allows the distribution and the execution of programs over the network. The language is a process calculus with XML values and datatypes, and with a pattern matching mechanism for deconstructing values. The compiler gives a typesafe bytecode in the form of an XML document, that may be deployed on the network. What comes out is a simple, statically typed, and formally defined core BPEL language with a basic query mechanism supplied by patterns.*

37. Rocco De Nicola, Daniele Gorla, Rosario Pugliese. *Basic Observables for a Calculus for Global Computing.* ICALP, pages 1226–1238, 2005.

    Abstract: *We introduce a foundational language for modelling applications over global computers whose interconnection structure can be explicitly manipulated. Together with process distribution, mobility, remote operations and asynchronous communication through distributed data spaces, the language provides constructs for explicitly modelling inter-node connections and for dynamically establishing and removing them. For the proposed language, we define natural notions of extensional observations and study their closure under operational reductions and/or language contexts to obtain barbed congruence and may testing equivalence. For such equivalences, we provide alternative characterizations in terms of a labelled bisimulation and a trace equivalence that can be used for actual proofs.*

38. Casey K. Fung, Patrick C. K. Hung, Guijun Wang, Richard C. Linger, Gwendolyn H. Walton. *A Study of Service Composition with QoS Management.* ICWS, pages 717–724, 2005.
Abstract: *Quality of service (QoS) management in compositions of services requires careful consideration of QoS characteristics of the services and effective QoS management in their execution. A Web service is a software system that supports interoperable application-to-application interaction over the Internet. Web services are based on a set of XML standards such as simple object access protocol (SOAP). The interactions of SOAP messages between Web services form the theoretical model of SOAP message exchange patterns (MEP). Web Services Business Process Execution Language (WSBPEL) defines an interoperable integration model that facilitates automated process integration in intra- and inter-corporate environments. A service-level agreement (SLA) is a formal contract between a Web services requestor and provider guaranteeing quantifiable issues at defined levels only through mutual concessions. Based on a prior research work on message detail record (MDR), this paper further proposes a SOAP message tracking model for supporting QoS end-to-end management in the context of WSBPEL and SLA. This paper motivates the study of QoS management in a Web service composition framework with the evolution of a distributed toolkit in an industrial setting.*

39. Dan Hirsch, Emilio Tuosto. *SHReQ: Coordinating Application Level QoS.* SEFM, pages 425–434, 2005.
Abstract: *We present SHReQ, a formal framework for specifying systems that handle abstract high-level QoS requirements which are becoming more and more important for service oriented computing. SHReQ combines Synchronised Hyperedge Replacement (SHR) with constraint-semirings. SHR is a (hyper)graph rewriting mechanism for modelling evolution of systems. The novelty of the approach relies on the synchronisation mechanism which is based on constraint-semirings, algebraic structures that provide both the mathematics for multi-criteria QoS and the synchronisation policies underlying the SHR mechanism.*

40. Hai Huang, Wei-Tek Tsai, Raymond Paul, Yinong Chen. *Automated Model Checking and Testing for Composite Web Services..* 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2005), pages 300–307, IEEE Computer Society, 2005.
Abstract: *Web Services form a new distributed computing paradigm. Collaborative verification and validation are important when Web Services from differen vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web Services by model checking the proces model of OWL-S (Web Ontology Language for Web Services) and to validate them by the test cases automatically generated in the model checking process. We extend the BLAST, a model checker tha handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

41. Michael C. Jaeger, Gero Mühl, Sebastian Golze. *QoS-Aware Composition of Web Services: A Look at Selection Algorithms.* ICWS, pages 807–808, 2005.
Abstract: *When a composition of Web services is designed, available services are put together to form a defined flow of executions. In a discovery process, a trader proposes available Web services as potential candidates. In a succeeding selection, for each task a trader chooses one candidate to form the optimal composition due to selection criteria. This paper discusses how the selection can consider different quality-of-service (QoS) categories to determine the most suitable candidates for the composition. If more than one category is used for optimisation, a multi-dimensional optimisation problem arises. This mentions similarities to similar combinatorial problems. Then, possible solutions are proposed and their performance is evaluated.*

42. Dimka Karastoyanova, Alejandro Houspanossian, Mariano Cilia, Frank Leymann, Alejandro P. Buchmann. *Extending BPEL for Run Time Adaptability.* EDOC, pages 15–26, 2005.
Abstract: *The existing Web Service Flow (WS-flow) technologies enable both static and dynamic binding of participating Web services (WSs) on the process model level. Adaptability on per-instance basis is not sufficiently supported and therefore must be addressed to improve process flexibility upon changes in the*

environment. Ad-hoc process instance changes can be enabled by swapping participating WS instances, by modifying port Types of the partners to be invoked, and by changing process logic. In this work we address the problem of dynamic binding of WSs to WS-flow instances at run time, i.e. the ability to exchange a WS instance participating in a WS-flow instance with an alternative one. The problem is additionally complicated by the fact that the execution of a process depends on its deployment. We describe the ”find and bind” mechanism, and we show its representation as a BPEL extension. We discuss the benefits that could be gained and the disadvantages it brings in. The mechanism extends and improves the existing process technologies. It facilitates a precisely controlled policy-based selection of WSs at run time and also provides for process instance repair, while maintaining simplicity. We also discuss a prototypical implementation of the presented functionality.

43. Khaled Mahbub, George Spanoudakis. *Run-time Monitoring of Requirements for Systems Composed of Web-Services: Initial Implementation and Evaluation Experience.* ICWS, pages 257–265, 2005.

    Abstract: *This paper describes a framework supporting the runtime monitoring of requirements for systems implemented as compositions of Web-services specified in BPEL. The requirements that can be monitored are specified in event calculus. The paper presents an overview of the framework and describes the architecture and implementation of a tool that we have developed to operationalise it. It also presents the results of a preliminary experimental evaluation of the framework.*

44. Mike P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann, Bernd J. Krämer. *05462 Service-Oriented Computing: A Research Roadmap.* Service Oriented Computing, 2005.

    Abstract: *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.*

45. Florian Rosenberg, Schahram Dustdar. *Business Rules Integration in BPEL - A Service-Oriented Approach.* CEC, pages 476–479, 2005.

    Abstract: *Business rules change quite often. These changes cannot be handled efficiently by representing business rules embedded in the source code of the business logic. Efficient handling of rules that govern ones business is one factor for success. That is where business rules engines play an important role. The service-oriented computing paradigm is becoming more and more popular. Services offered by different providers, are composed to new services by using Web service composition languages such as BPEL. Such process-based composition languages lack the ability to use business rules managed by different business rules engines in the composition process. In this paper, we propose an approach on how to use and integrate business rules in a service-oriented way into BPEL.*

46. Wei-Tek Tsai, Yinong Chen, Raymond A. Paul. *Specification-Based Verification and Validation of Web Services and Service-Oriented Operating Systems.* WORDS, pages 139–147, 2005.

    Abstract: *Service-Oriented Architecture (SOA) and Web Services (WS) have received significant attention* *recently. Even though WS are based on open standards and support software interoperability, but the trustworthy issues of WS has actually limited the growth of WS applications as organizations do not trust those WS developed by other vendors and at the same time they do not have access to the source code. This paper addressed this issue by proposing several solutions including specification-based verification and validation, collaborative testing, and group testing. The key concept is that it is possible to provide a comprehensive evaluation of WS even if their source code is not available.*

47. Foster, H., Uchitel, S., Magee, J., Kramer, J.. *Compatibility verification for Web service choreography.* pages 738–741, July 2004.

    Abstract: *In this paper we discuss a model-based approach to verifying process interactions for coordinated Web service compositions. The approach uses finite state machine representations of Web service orchestrations and assigns semantics to the distributed process interactions. The move towards implementing Web service compositions by multiple interested parties as a form of distributed system architecture motivates the need for supporting compatibility verification of activities and transactions in all the processes. The described approach is supported by a suite of cooperating tools for specification, formal modeling and providing verification results from orchestrated Web service interactions.*

48. Rohit Aggarwal, Kunal Verma, John A. Miller, William Milnor. *Constraint Driven Web Service Composition in METEOR-S.* IEEE SCC, pages 23–30, 2004.

    Abstract: *Creating Web processes using Web service technology gives us the opportunity for selecting new services which best suit our need at the moment. Doing this automatically would require us to quantify our criteria for selection. In addition, there are challenging issues of correctness and optimality. We present a Constraint Driven Web Service Composition tool in METEOR-S, which allows the process designers to bind Web Services to an abstract process, based on business and process constraints and generate an executable process. Our approach is to reduce much of the service composition problem to a constraint satisfaction problem. It uses a multi-phase approach for constraint analysis. This work was done as part of the METEOR-S framework, which aims to support the complete lifecycle of semantic Web processes.*

49. Jesus Arias-Fisteus, Luis Sanchez Fernandez, Carlos Delgado Kloos. *Formal Verification of BPEL4WS Business Collaborations.* E-Commerce and Web Technologies, 5th International Conference, EC-Web 2004, Proceedings, Lecture Notes in Computer Science, Vol. 3182, pages 76–85, Springer Verlag, 2004.

    Abstract: *Web services are a very appropriate communication mechanism to perform distributed business processes among several organisations. These processes should be reliable, because a failure in them can cause high economic losses. To increase their reliability at design time, we have developed VERBUS, a framework for the formal verification of business processes. VERBUS can automatically translate business process definitions to specifications verifiable in several available tools. It is based on a modular and extensible architecture: new process definition languages and verification tools can be added easily to the framework. The prototype of VERBUS presented in this work can verify BPEL4WS process specifications, by translating them to Promela. The Promela specifications are verified with the well known model checker Spin. In this paper we describe the general architecture of VERBUS and how BPEL4WS specifications are translated and verified. The explanation is completed by describing what types of properties can be verified and providing an overview of the implementation.*

50. Tevfik Bultan, Xiang Fu, Jianwen Su. *Tools for Automated Verification of Web Services.* ATVA, pages 8–10, 2004.

51. Vikas Deora, Jianhua Shao, Gareth Shercliff, Patrick J. Stockreisser, W. A. Gray, N. J. Fiddian. *Incorporating QoS Specifications in Service Discovery.* WISE Workshops, pages 252–263, 2004.

    Abstract: *In this paper, we extend the current approaches to service discovery in a service oriented computing environment, such as Web Services and Grid, by allowing service providers and consumers to express their promises and requirements for quality of service (QoS). More specifically, we allow service providers to advertise their services in an extended DAML-S that supports quality specifications, and we allow service consumers to request services by stating required quality levels. We propose a model here for incorporating*

QoS specifications and requirements in service discovery, and describe how matchmaking between advertised and requested services based on functional as well as quality requirements is supported in our model.

52. Roozbeh Farahbod, Uwe Glässer, Mona Vajihollahi. *Specification and Validation of the Business Process Execution Language for Web Services.* Abstract State Machines, pages 78–94, 2004.

    Abstract: *We formally define an abstract executable semantics for the Business Process Execution Language for Web Services in terms of a distributed ASM. The goal of this work is to support the design and standardization of the language. There is a need for formalism. It will allow us to not only reason about the current specification and related issues, but also uncover issues that would otherwise go unnoticed. Empirical deduction is not sufficient. - Issue 42, OASIS WSBPEL TC. The language definition assumes an infrastructure for running Web services on some asynchronous communication architecture. A business process is built on top of a collection of Web services performing continuous interactions with the outside world by sending and receiving messages over a communication network. The underlying execution model is characterized by its concurrent and reactive behavior making it particularly difficult to predict dynamic system properties with a sufficient degree of detail and precision under all circumstances.*

53. Xiang Fu, Tevfik Bultan, Jianwen Su. *Analysis of Interacting BPEL Web Services.* WWW'04: Proceedings of the 13th International Conference on World Wide Web, pages 621–630, ACM Press, 2004.

    Abstract: *This paper presents a set of tools and techniques for analyzing interactions of composite web services which are specified in BPEL and communicate through asynchronous XML messages. We model the interactions of composite web services as conversations, the global sequence of messages ex- changed by the web services. As opposed to earlier work, our tool-set handles rich data manipulation via XPath expressions. This allows us to verify designs at a more detailed level and check properties about message content. We present a framework where BPEL specifications of web ser- vices are translated to an intermediate representation, followed by the translation of the intermediate representation to a verification language. As an intermediate representation we use guarded automata augmented with unbounded queues for incoming messages, where the guards are expressed as XPath expressions. As the target verification language we use Promela, input language of the model checker SPIN. Since SPIN model checker is a finite-state verification tool we can only achieve partial verification by fixing the sizes of the input queues in the translation. We propose the concept of synchronizability to address this problem. We show that if a composite web service is synchronizable, then its conversation set remains same when asynchronous communication is replaced with synchronous communication. We give a set of sufficient conditions that guarantee synchronizability and that can be checked statically. Based on our synchronizability results, we show that a large class of composite web services with unbounded input queues can be completely verified using a finite state model checker such as SPIN.*

54. Xiang Fu, Tevfik Bultan, Jianwen Su. *Realizability of Conversation Protocols With Message Contents.* ICWS, 2004.

    Abstract: *A conversation protocol is a top-down specification framework which specifies desired global behaviors of a web service composition. In our earlier work [A formalism for specification and verification of reactive electronic services] we studied the problem of realizability, i.e., given a conversation protocol, can a web service composition be synthesized to generate behaviors as specified by the protocol. Several sufficient realizability conditions were proposed in [A formalism for specification and verification of reactive electronic services] to ensure realizability. Conversation protocols studied in [A formalism for specification and verification of reactive electronic services], however, are essentially abstract control flows without data semantics. This paper extends the work in [A formalism for specification and verification of reactive electronic services] and achieves more accurate analysis by considering data semantics. To overcome the state-space explosion caused by the data content, we propose a symbolic analysis technique for each realizability condition. In addition, we show that the analysis of the autonomy condition can be done using an iterative refinement approach.*

55. Xiang Fu, Tevfik Bultan, Jianwen Su. *Model checking XML manipulating software.* ISSTA, pages 252–262, 2004.

Abstract: *The use of XML as the de facto data exchange standard has allowed integration of heterogeneous web based software systems regardless of implementation platforms and programming languages. On the other hand, the rich tree-structured data representation, and the expressive XML query languages (such as XPath) make formal specification and verification of software systems that manipulate XML data a challenge. In this paper, we present our initial efforts in automated verification of XML data manipulation operations using the SPIN model checker. We present algorithms for translating (bounded) XML data and XPath expressions to Promela, the input language of SPIN. The techniques presented in this paper constitute the basis of our Web Service Analysis Tool (WSAT) which verifies LTL properties of composite web services.*

56. Gang Huang, Meng Wang, Liya Ma, Ling Lan, Tiancheng Liu, Hong Mei. *Towards Architecture Model based Deployment for Dynamic Grid Services.* IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04), 2004.

    Abstract: *The deployment of grid services should make the services, including those to be deployed and those already deployed, operate with desired functionalities and qualities. The critical challenge in the deployment is that many technical and nontechnical factors have to be taken into account, such as performance, reliability, utilization, operating cost, incomes, and so on. Since the factors change continuously, some deployed services may have to be redeployed for guaranteeing their functionalities and qualities. This position paper presents an approach to the deployment and redeployment of grid services based on software architecture models. In this approach, all services in a grid consist in a software architecture, which represents the services, their relationships and other factors in a global, understandable and easy-to-use way. To demonstrate the approach, a visual tool for deploying services onto a set of popular grid infrastructures, including J2EE application servers and BPEL engines, with the help of software architectures is developed.*

57. Michael C. Jaeger, Gregor Rojec-Goldmann, Gero Mühl. *QoS Aggregation for Web Service Composition using Workflow Patterns.* EDOC, pages 149–159, 2004.

    Abstract: *Contributions in the field of Web services have identified that (a) finding matches between semantic descriptions of advertised and requested services and (b) nonfunctional characteristics - the quality of service (QoS) - are the most crucial criteria for composition of Web services. A mechanism is introduced that determines the QoS of a Web service composition by aggregating the QoS dimensions of the individual services. This allows to verify whether a set of services selected for composition satisfies the QoS requirements for the whole composition. The aggregation performed builds upon abstract composition patterns, which represent basic structural elements of a composition, like sequence, loop, or parallel execution. This work focusses on workflow management environments. We define composition patterns that are derived from Van der Aalst's et al. comprehensive collection of workflow patterns. The resulting aggregation schema supports the same structural elements as found in workflows. Furthermore, the aggregation of several QoS dimensions is discussed.*

58. Jinghai Rao, Xiaomeng Su. *A Survey of Automated Web Service Composition Methods.* SWSWPC, pages 43–54, 2004.

    Abstract: *In todays Web, Web services are created and updated on the fly. Its already beyond the human ability to analysis them and generate the composition plan manually. A number of approaches have been proposed to tackle that problem. Most of them are inspired by the researches in cross-enterprise workflow and AI planning. This paper gives an overview of recent research efforts of automatic Web service composition both from the workflow and AI planning research community.*

59. Gwen Salaün, Lucas Bordeaux, Marco Schaerf. *Describing and Reasoning on Web Services using Process Algebra.* ICWS, 2004.

    Abstract: *We argue that essential facets of web services, and especially those useful to understand their interaction, can be described using process-algebraic notations. Web service description and execution languages such as BPEL are essentially process description languages; they are based on primitives for behaviour description and message exchange which can also be found in more abstract process algebras. One legitimate question is therefore whether the formal approach and the sophisticated tools introduced for process algebra can be used to improve the effectiveness and the reliability of web service development. Our investigations suggest a positive answer, and we claim that process algebras provide a very complete and satisfactory assistance to the whole process of web service development. We show on a case study that readily available* *tools based on process algebra are effective at verifying that web services conform their requirements and respect properties. We advocate their use both at the design stage and for reverse engineering issues. More prospectively, we discuss how they can be helpful to tackle choreography issues.*

60. James Skene, D. Davide Lamanna, Wolfgang Emmerich. *Precise Service Level Agreements.* ICSE, pages 179–188, 2004.

    Abstract: *SLAng is an XML language for defining service level agreements, the part of a contract between the client and provider of an Internet service that describes the quality attributes that the service is required to possess. We define the semantics of SLAng precisely by modelling the syntax of the language in UML, then relating the language model to a model that describes the structure and behaviour of services. The presence of SLAng elements imposes behavioural constraints on service elements, and the precise definition of these constraints using OCL constitutes the semantic description of the language. We use the semantics to define a notion of SLA compatibility, and an extension to UML that enables the modelling of service situations as a precursor to analysis, implementation and provisioning activities.*

61. Andreas Wombacher, Peter Fankhauser, Erich J. Neuhold. *Transforming BPEL into Annotated Deterministic Finite State Automata for Service Discovery.* ICWS, pages 316–323, 2004.

    Abstract: *Web services advocate loosely coupled systems, although current loosely coupled applications are limited to stateless services. The reason for this limitation is the lack of a method supporting matchmaking of state dependent services exemplarily specified in BPEL. In particular, the sender's requirement that the receiver must support all possible messages sent at a certain state are not captured by models currently used for service discovery. Annotated deterministic finite state automata provide this expressiveness. In this paper the transformation of a local process specification given in BPEL to annotated deterministic finite state automata is presented.*

62. Tao Yu, Kwei-Jay Lin. *Service Selection Algorithms for Web Services with End-to-End QoS Constraints.* CEC, pages 129–136, 2004.

    Abstract: *Web services are new forms of Internet software that can be universally deployed and invoked using standard protocol. Services from different providers can be integrated to provide composite services. In this paper, we study the end-to-end QoS issues of composite service by utilizing a QoS broker that is responsible for coordinating the individual service component to meet the quality constraint. We design the service selection algorithms used by QoS brokers to meet end-to-end QoS constraints. The objective of the algorithms is to maximize the user-defined utility while meeting the end-to-end delay constraint. We model the problem as the Multiple Choice Knapsack Problem (MCKP) and provide efficient solutions. The algorithms are tested for their performance.*

63. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Mathias Weske. *Business Process Management: A Survey .* International Conference on Business Process Management (BPM), Lecture Notes in Computer Science, Vol. 2678, Springer Verlag, 2003.

    Abstract: *Business Process Management (BPM) includes methods, techniques, and tools to support the design, enactment, management, and analysis of operational business processes. It can be considered as an extension of classical Workflow Management (WFM) systems and approaches. Although the practical relevance of BPM is undisputed, a clear definition of BPM and related acronyms such as BAM, BPA, and STP are missing. Moreover, a clear scientific foundation is missing. In this paper, we try to demystify the acronyms in this domain, describe the state-of-the-art technology, and argue that BPM could benefit from formal methods/languages (cf. Petri nets, process algebras, etc.).*

64. Vikas Deora, Jianhua Shao, W. A. Gray, N. J. Fiddian. *A Quality of Service Management Framework Based on User Expectations.* ICSOC, pages 104–114, 2003.

    Abstract: *The ability to gauge the quality of a service is critical if we are to achieve the service oriented computing paradigm. Many techniques have been proposed and most of them attempt to calculate the quality of a service by collecting quality ratings from the users of the service, then combining them in one way or another. We argue that collecting quality ratings alone from the users is not sufficient for deriving a reliable*

or accurate quality measure for a service. This is because different users often have different expectations on the quality of a service and their ratings tend to be closely related to their expectations, i.e. how their expectations are met. In this paper, we propose a quality of service management framework based on user expectations. That is, we collect expectations as well as ratings from the users of a service, then calculate the quality of the service only at the time a request for the service is made and only using the ratings that have similar expectations. We give examples to show that our approach can result in a more accurate and meaningful measure for quality of service.

65. Howard Foster, Sebastián Uchitel, Jeff Magee, Jeff Kramer. *Model-based Verification of Web Service Compositions..* 18th IEEE International Conference on Automated Software Engineering (ASE 2003) , pages 152–163, IEEE Computer Society, 2003.
Abstract: *In this paper we discuss a model-based approach to verifying web service compositions for web service implementations. The approach supports verification against specification models and assigns semantics to the behavior of implementation models so as to confirm expected results for both the designer and implementer. Specifications of the design are modeled in UML, in the form of Message Sequence Charts (MSCs), and mechanically compiled into the Finite State Process notation (FSP) to concisely describe and reason about the concurrent programs. Implementations are mechanically translated to FSP to allow a trace equivalence verification process to be performed. By providing early design verification, the implementation, testing and deployment of web service compositions can be eased through the understanding of the differences, limitations and undesirable traces allowed by the composition. The approach is supported by a suite of cooperating tools for specification, formal modeling and trace animation of the composition workflow.*

66. Rania Khalaf, Frank Leymann. *On Web Services Aggregation.* TES, pages 1–13, 2003.
Abstract: *The Web services framework is enabling applications from different providers to be offered as services that can be used and composed in a loosely-coupled manner. Subsequently, the aggregation of services to form composite applications and maximize reuse is key. While choreography has received the most attention, services often need to be aggregated in a much less constrained manner. As a number of different mechanisms emerge to create these aggregations, their relation to each other and to prior work is useful when deciding how to create an aggregation, as well as in extending the models themselves and proposing new ones. In this paper, we discuss Web services aggregation by presenting a first-step classification based on the approaches taken by the different proposed aggregation techniques. Finally, a number of models are presented that are created from combinations of the above.*

67. Octavio Martín-Díaz, Antonio Ruiz Cortés, Amador Durán, David Benavides, Miguel Toro. *Automating the Procurement of Web Services.* ICSOC, pages 91–103, 2003.
Abstract: *As government agencies and business become more dependent on web services, software solutions to automate their procurement gain importance. Current approaches for automating the procurement of web services suffer from an important drawback: neither uncertainty measures nor non-linear, and complex relations among parameters can be used by providers to specify quality-of-service in offers. In this paper, we look deeply into the roots of this drawback and present a proposal which overcomes it. The key point to achieve this improvement has been using the constraint programming as a formal basis, since it endows the model with a very powerful expressiveness. A XML-based implementation is presented along with some experimental results and comparisons with other approaches.*

68. Vladimir Tosic, Bernard Pagurek, Kruti Patel. *WSOL - A Language for the Formal Specification of Classes of Service for Web Services.* ICWS, pages 375–381, 2003.
Abstract: *We are developing a language, Web Service Offerings Language (WSOL), for the formal specification of various constraints, management statements, and classes of service for Web Services. WSOL is an XML (Extensible Markup Language) notation compatible with WSDL (Web Services Description Language). A service offering in WSOL is a formal description of one class of service of a Web Service. It contains formal representation of various constraints: functional (pre-, post-, and future- conditions), Quality of Service (QoS, a.k.a. non-functional, extra- functional), and access rights. It also contains management statements, such as statements about prices, monetary penalties, and management responsibilities. One Web Service can be associated with multiple service offerings. For easier specification of similar service offerings, WSOL*

enables specification of constraint groups (CGs) and constraint group templates (CGTs). We have also developed a format for representation of dynamic relationships between service offerings. WSOL service offerings are simple contracts and SLAs (Service Level Agreements) between Web Services. Describing a Web Service in WSOL, in addition to WSDL, enables monitoring, metering, and management of Web Services. The Web Service, its consumer, or one or more designated third parties (usually SOAP message intermediaries) can meter QoS metrics and evaluate constraints in WSOL service offerings. Further, manipulation of service offerings can be used for dynamic adaptation and management of Web Service compositions. In addition, WSOL supports selection of a more appropriate Web Service and service offering for particular circumstances. The main distinctive characteristics of WSOL, compared to recent related works, are its expressive power, features that reduce run-time overhead, and orientation towards management applications.

69. Massimo Mecella, Francesco Parisi Presicce an Barbara Pernici. *Modeling E -service Orchestration through Petri Nets .* Technologies for E-Services, Lecture Notes in Computer Science, Vol. 2444, pages 109–134, Springer Verlag, July 2002.

    Abstract: *B2B interaction requires new forms of coordination between participating organizations. Indeed, the main requirement is that the autonomy of each participating partner is preserved during the interaction, guaranteeing at the same time that the overall goals of the common process are reached. Mechanisms for regulating distributed workflow evolution when the workflow is composed of the invocation of E-services of different organizations are needed. The E-service orchestration model proposed in this paper provides a mechanism for supporting control of process evolution in terms both of control and data flows, and for distributing and assigning process responsibilities.*

70. Srini Narayanan, Sheila A. McIlraith. *Simulation, verification and automated composition of web services.* WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 77–88, ACM, 2002.

    Abstract: *Web services – Web-accessible programs and devices - are a key application area for the Semantic Web. With the proliferation of Web services and the evolution towards the Semantic Web comes the opportunity to automate various Web services tasks. Our objective is to enable markup and automated reasoning technology to describe, simulate, compose, test, and verify compositions of Web services. We take as our starting point the DAML-S DAML+OIL ontology for describing the capabilities of Web services. We define the semantics for a relevant subset of DAML-S in terms of a first-order logical language. With the semantics in hand, we encode our service descriptions in a Petri Net formalism and provide decision procedures for Web service simulation, verification and composition. We also provide an analysis of the complexity of these tasks under different restrictions to the DAML-S composite services we can describe. Finally, we present an implementation of our analysis techniques. This implementation takes as input a DAML-S description of a Web service, automatically generates a Petri Net and performs the desired analysis. Such a tool has broad applicability both as a back end to existing manual Web service composition tools, and as a stand-alone tool for Web service developers.*

71. Vladimir Tosic, Kruti Patel, Bernard Pagurek. *WSOL - Web Service Offerings Language.* CAiSE '02/ WES '02: Revised Papers from the International Workshop on Web Services, E-Business, and the Semantic Web, pages 57–67, Springer-Verlag, 2002.

    Abstract: *WSOL (Web Service Offerings Language) is an XML (Extensible Markup Language) notation compatible with the WSDL (Web Services Description Language) standard. While WSDL is used for describing operations provided by Web Services, WSOL enables formal specification of multiple classes of service for one Web Service. A service offering is a formal representation of one class of service for a Web Service. As classes of service for Web Services are determined by combinations of various constraints, WSOL enables formal specification of functional constraints, some QoS (a.k.a., non-functional) constraints, simple access rights (for differentiation of service), price, and relationships with other service offerings of the same Web Service. Describing a Web Service in WSOL, in addition to WSDL, enables selection of a more appropriate Web Service and service offering for particular circumstances. Further, it supports dynamic adaptation and management of Web Service compositions using manipulation of service offerings.*

72. Wil M. P. van der Aalst, Alistair P. Barros, Arthur H. M. ter Hofstede, Bartek Kiepuszewski. *Advanced Workflow Patterns.* CoopIS, pages 18–29, 2000.

    Abstract: *Conventional workflow functionality like task sequencing, split parallelism, join synchronization and iteration have proven effective for business process automation and have widespread support in current workflow products. However, newer requirements for workflows are enconntered in practice, opening grave uncertainties about the extensions for current languages. Different concepts, although outwardly appearing to be more or less the same, are based on different paradigms, have fundamentally different semantics and different levels of applicability -more specialized for modeling or more generalized for workflow engine posit. By way of developmental insight of new requirements, we define workflow patterns which are described imperatively but independently of current workflow languages. These patterns provide the basis for an in-depth comparison of 12 workflow management systems. As such, the work reported in this paper can be seen as the academic response to evaluations made by prestigious consulting companies. Typically, these evaluations hardly consider the workflow modeling language and routing capabilities and focus more on the purely technical and commercial aspects.*

73. Wil M. P. van der Aalst. *Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques.* Business Process Management, pages 161–183, 2000.

    Abstract: *Workflow management systems facilitate the everyday operation of business processes by taking care of the logistic control of work. In contrast to traditional information systems, they attempt to support frequent changes of the workflows at hand. Therefore, the need for analysis methods to verify the correctness of workflows is becoming more prominent. In this chapter we present a method based on Petri nets. This analysis method exploits the structure of the Petri net to find potential errors in the design of the workflow. Moreover, the analysis method allows for the compositional verification of workflows.*

74. Laurent Fribourg, Marcos Veloso Peixoto. *Concurrent Constraint Automata.* International Logic Programming Symposium, 1993.

    Abstract: *We address the problem of the specification and the proof of properties of concurrent systems which manipulate an unbounded number of data. We propose an approach based on an extended notion of automata, called "Concurrent Constraint Automata (CCA)". A CCA is an automaton with constraints and synchronous communication. By automata with constraints, we mean a state machine whose actions and states contain parameters that take their value in the set of natural numbers. (...)*

Books and Monographs:

1. Mariya Koshkina. *Verification of Business Processes for Web Services.* Ms. Thesis, York University - Department of Computer Science, October 2003.

   Abstract: *The Business Process Execution Language for Web Services (BPEL4WS or simply BPEL) is a recently developed language, which is used to specify interactions between web services. Among its features it allows specification of concurrent behavior. Erroneous specification can lead to such problems as deadlock. In our research we focus on the concurrency mechanism in BPEL. Our main goal is to analyze processes in order to detect possible deadlocks. To achieve this we introduce a process algebra called the BPE-calculus. It is a small language which captures all the BPEL features relevant to the analysis. This process algebra is modelled using a labeled transition system. An existing verification tool called the Concurrency Workbench is customized to use our BPE-calculus. This tool allows us to verify many properties of BPE-calculus processes specified in a logic called the mu-calculus, including deadlock freedom.*

2. J. Cardoso. *Quality of Service and Semantic Composition of Workflows.* Ph.D. Thesis, University of Georgia, 2002.

   Abstract: *Workflow management systems (WfMSs) have been used to support a variety of business processes. As organizations adopt new working models, such as e-commerce, new challenges arise for workflow systems. These challenges include support for the adequate management of quality of service (QoS) and the development of new solutions to facilitate the composition of workflow applications involving Web services. The good management of QoS directly impacts the success of organizations participating in e- commerce activities by better fulfilling customer expectations and achieving customer satisfaction. To enable adequate*

QoS management, research is required to develop mechanisms that specify, compute, monitor, and control the QoS of the products or services to be delivered. The composition of workflows to model e-service applications differs from the design of traditional workflows due to the number of Web services available during the composition process and to their heterogeneity. Two main problems need to be solved: how to efficiently discover Web services and how to facilitate their interoperability. To enhance WfMSs with QoS management, we have developed a QoS model that allows for the description of nonfunctional aspects of workflow components, from a quality of service perspective. To automatically compute the overall QoS of a workflow, we have developed a mathematical model and implemented an algorithm (SWR algorithm). Our QoS model and mathematical model have been validated with the deployment and execution of a set of production workflows in the area of genetics. The analysis of the collected data proves that our models provide a suitable framework for estimating, predicting, and analyzing the QoS of production workflows. To support, facilitate, and assist the composition of workflows involving Web services, we present a solution based on ontologies. We have developed an algorithm that workflow systems and discovery mechanisms can use to find Web services with desired interfaces and operational metrics, and to assist designers in resolving heterogeneity issues among Web services. Our approach provides an important solution to enhance Web service discovery and interoperability.

Articles in Books and Other Collections:

1. Natasha Sharygina, Daniel Kröning. *Model Checking with Abstraction for Web Services.* Test and Analysis of Web Services, pages 121–145, 2007.
   Abstract: *Web services are highly distributed programs and, thus, are prone to concurrency-related errors. Model checking is a powerful technique to identify flaws in concurrent systems. However, the existing model checkers have only very limited support for the programming languages and communication mechanisms used by typical implementations of web services. This chapter presents a formalization of communication semantics geared for web services, and an automated way to extract formal models from programs implementing web services for automatic formal analysis. The formal models are analyzed by means of a symbolic model checker that implements automatic abstraction refinement. Our implementation takes one or more PHP5 programs as input, and is able to verify joint properties of these programs running concurrently.*

Publications in Refereed Workshops:

1. S. Nakajima. *Model-Checking Behavioral Specification of BPEL Applications.* Proceedings of the International Workshop on Web Languages and Formal Methods, WLFM 2005, 2005.
   Abstract: *To provide a framework to compose lots of specialised services flexibly, BPEL is proposed to describe Web service flows. Since the Web service flow description is basically a distributed collaboration, writing correct programs in BPEL is not easy. Verifying BPEL program prior to its execution is essential. This paper proposes a method to extract the behavioral specification from a BPEL appliation program and to analyze it by using the SPIN model checker. With the adequate abstraction method and support for DPE, the method can analyze all the four example cases in the BPEL standard document.*

2. M. Tian, A. Gramm, T. Naumowicz, H. Ritter, J. Schiller. *A Concept for QoS Integration in Web Services.* Web Information Systems Engineering Workshops, pages 149–155, December 2003.
   Abstract: *With the growing popularity of Web services, a general QoS support for Web services plays an important role for the success of this emerging technology. Unfortunately, current Web service environments do not offer comprehensive QoS support. In this paper, we present an approach that does not only enable the QoS integration in Web services, but also the selection of appropriate services based on QoS requirements regarding server and network performance. Furthermore, we present how application requirements regarding communication QoS are mapped onto the underlying QoS aware network at runtime, as well as how users can obtain real-time information about server performance in order to monitor the accomplishment of assured services, giving the user an instant QoS feedback.*

3. S. Nakajima. *Model-Checking Verification for Reliable Web Service.* OOPSLA Workshop on Object-Oriented Web Services, 2002.

   Abstract: *Model-checking is a promising technique for the verification and validation of software systems. Web service, an emerging technology in the Internet, is an autonomous server that may offer an individual service. It sometimes requires to combine more than one to meet our requirements. WSFL(Web Services Flow Language) is proposed as a language to provide means to describe Web service aggregation. We are interested in how much the software model-checking technique can be used as a basis for raising reliability of Web service, Web service flow descriptions in particular. Our experience shows that faulty flow descriptions can be identified with the proposed method. The method is also very helpful in studying an alternative semantics of the WSFL in regard to the handling of dataflows.*

4. Sabata, B., Chatterjee, S., Davis, M., Sydir, J.J., Lawrence, T.F.. *Taxonomy for QoS specifications.* pages 100–107, 1997.

   Abstract: *It is becoming increasingly commonplace for multiple applications with different quality of service (QoS) requirements to share the resources of a distributed system. Within this environment, the resource management algorithms must take into account the QoS desired by applications and the ability of the system resources to provide it. We present a taxonomy for specifying QoS for the different components of a distributed system, from the applications down to the resources. We specify QoS as a combination of metrics and policies. QoS metrics are used to specify performance parameters, security requirements and the relative importance of the work in the system. We define three types of QoS performance parameters: timeliness, precision, and accuracy. QoS policies capture application-specific policies that govern how an application is treated by the resource manager. Examples of such policies are management policies and the levels of service. We explore each of these components of the QoS taxonomy in detail*

Technical Reports and Manuals:

1. Diane Jordan, John Evdemon, Alexandre Alves, Assaf Arkin, Sid Askary, Charlton Barreto, Ben Bloch, Francisco Curbera, Mark Ford, Yaron Goland, Alejandro Guzar, Neelakantan Kartha, Canyang Kevin Liu, Rania Khalaf, Dieter Knig, Mike Marin, Vinkesh Mehta, Satish Thatte, Danny van der Rijn, Prasad Yendluri, Alex Yiu . *Web Services Business Process Execution Language Version 2.0.* IBM, Microsoft, BEA, Intalio, Individual, Adobe Systems, Systinet, Active Endpoints, JBoss, Sterling Commerce, SAP, Deloitte, TIBCO Software, webMethods, Oracle, 2007.

   Abstract: *This document defines a language for specifying business process behavior based on Web Services. This language is called Web Services Business Process Execution Language (abbreviated to WS-BPEL in the rest of this document). Processes in WS-BPEL export and import functionality by using Web Service interfaces exclusively. Business processes can be described in two ways. Executable business processes model actual behavior of a participant in a business interaction. Abstract business processes are partially specified processes that are not intended to be executed. An Abstract Process may hide some of the required concrete operational details. Abstract Processes serve a descriptive role, with more than one possible use case, including observable behavior and process template. WS-BPEL is meant to be used to model the behavior of both Executable and Abstract Processes. WS-BPEL provides a language for the specification of Executable and Abstract business processes. By doing so, it extends the Web Services interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to- business spaces.*

2. Siddharth Bajaj, Don Box, Dave Chappell, Francisco Curbera, Glen Daniels, Phillip Hallam-Baker, Maryann Hondo, Chris Kaler, Dave Langworthy, Anthony Nadalin, Nataraj Nagaratnam, Hemma Prafullchandra, Claus von Riegen, Daniel Roth, Jeffrey Schlimmer, Chris Sharp, John Shewchuk, Asir Vedamuthu, Ümit Yal cınalp, David Orchard . *Web Services Policy Framework (WS- Policy).* IBM, March 2006.

   Abstract: *The Web Services Policy Framework (WS-Policy) provides a general purpose model and corresponding syntax to describe the policies of a Web Service. WS-Policy defines a base set of constructs that can* *be used and extended by other Web services specifications to describe a broad range of service requirements and capabilities.*

3. Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, Ming Xu. *Web Services Agreement Specification (WS-Agreement).* Grid Resource Allocation Agreement Protocol, September 2005.

   Abstract: *This document describes Web Services Agreement Specification (WS-Agreement), a Web Services protocol for establishing agreement between two parties, such as between a service provider and consumer, using an extensible XML language for specifying the nature of the agreement, and agreement templates to facilitate discovery of compatible agreement parties. The specification consists of three parts which may be used in a composable manner: a schema for specifying an agreement, a schema for specifying an agreement template, and a set of port types and operations for managing agreement life-cycle, including creation, expiration, and monitoring of agreement states.*

4. The OASIS Group. *Quality Model for Web Services.* The Oasis Group, September 2005.

   Abstract: *The purpose of this document is to provide a model for Web services quality management and quality factors in the process of developing and using Web services. We define the consistent and systematic conceptual model of Web services quality, which may be used by intimate associates, i.e. stakeholders, developers, service providers, and customers of Web services.*

5. The OMG Group. *UML^{TM} Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms.* Num. ptc/2005-05-02, The OMG Group, May 2005.

6. I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich . *The Open Grid Services Architecture, Version 1.0.* ANL, IBM, Fujitsu, NeSC, CCLRC-RAL, UVa, Hitachi, Intel, HP, 2005.

   Abstract: *Successful realization of the Open Grid Services Architecture (OGSA) vision of a broadly applicable and adopted framework for distributed system integration, virtualization, and management requires the definition of a core set of interfaces, behaviors, resource models, and bindings. This document, produced by the OGSA working group within the Global Grid Forum (GGF), provides a first version of this OGSA definition. The document focuses on requirements and the scope of important capabilities required to support Grid systems and applications in both e-science and e-business. The capabilities described are Execution Management, Data, Resource Management, Security, Self-Management, and Information. The description of the capabilities is at a high-level and includes, to some extent, the interrelationships between the capabilities.*

7. The Oasis Group. *Summary of Quality Model for Web Services.* The Oasis Group, 2005.

8. Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, Ivana Trickovic, Sanjiva Weerawarana. *Business Process Execution Language for Web Services, version 1.1.* BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems, 2003.

   Abstract: *This document defines a notation for specifying business process behavior based on Web Services. This notation is called Business Process Execution Language for Web Services (abbreviated to BPEL4WS in the rest of this document). Processes in BPEL4WS export and import functionality by using Web Service interfaces exclusively. Business processes can be described in two ways. Executable business processes model actual behavior of a participant in a business interaction. Business protocols, in contrast, use process descriptions that specify the mutually visible message exchange behavior of each of the parties involved in the protocol, without revealing their internal behavior. The process descriptions for business protocols are called abstract processes. BPEL4WS is meant to be used to model the behavior of both executable and abstract processes. BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions. BPEL4WS defines an interoperable integration model that should facilitate*

the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.

9. Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, Richard Franck . *Web Service Level Agreement (WSLA) Language Specification.* IBM Corporation, 2003.

   Abstract: *This document describes the specification language for service level agreements for Web Services, the Web Service Level Agreement (WSLA) language. WSLAs are agreements between a service provider and a customer and as such define the obligations of the parties involved. Primarily, this is the obligation of a service provider to perform a service according to agreed-upon guarantees for IT-level service parameters (such as availability, response time and throughput) for Web Services.*

   An SLA also specifies the measures to be taken in case of deviation and failure to meet the asserted service guarantees, for example, a notification of the service customer. The assertions of the service provider are based on a detailed definition of service parameters including the algorithms – how basic metrics should be measured in systems and how they are aggregated into composite metrics and SLA parameters. In addition, a WSLA can express the operations of monitoring and managing the service. This may include third parties (such as Management Service Providers) that contribute to the measurement of metrics, supervision of guarantees or even the management of deviations of service guarantees. These multi-party constellations necessitate the definition of the interactions among the parties supervising the WSLA.

   However, a WSLA only covers the agreed common view of a service between the parties involved. To actually act as a participant in a WSLA, parties have various degrees of freedom to define an implementation policy for a service and its supervision. Typically, the obligations of a WSLA must be translated into system-level configuration information, which can be proprietary to each party involved.

10. Akhil Sahai, Anna Durante, Vijay Machiraju. *Towards Automated SLA Management for Web Services.* Num. HPL-2001-310, HP Laboratories, July 2002.

    Abstract: *In order to automate SLA management it is essential to specify SLAs in precise and unambiguous manner as well as keep the specification flexible. While precision will help automate the process of monitoring and metric collection, flexibility will enable extending it to unforeseen service level agreement specifications.*

11. Anbazhagan Mani, Arun Nagarajan. *Understanding quality of service for Web services.* IBM, 2002.

    Abstract: *With the widespread proliferation of Web services, quality of service (QoS) will become a significant factor in distinguishing the success of service providers. QoS determines the service usability and utility, both of which influence the popularity of the service. In this article, we look at the various Web service QoS requirements, bottlenecks affecting performance of Web services, approaches of providing service quality, transactional services, and a simple method of measuring response time of your Web services using the service proxy.*

12. Svend Frolund, Jari Koistinen. *QML: A Language for Quality of Service Specification.* Num. HPL-98-10, 63 pages, HP Laboratories, 1998.

    Abstract: *To be competitive, future software system must provide not only the correct functionality, but also an adequate level of quality of service (QoS). By QoS, we refer to non-functional properties, such as reliability, performance, timing, and security. To provide an adequate level of QoS, software systems need to include capabilities such as QoS negotiation, monitoring, and adaptation. These capabilities all require the expected and the provided QoS levels to be explicitly specified. QoS can be specified statically at the time of implementation, design, or dynamically at deployment or runtime. To facilitate QoS specification, we present a general Quality of service Modeling Language (QML) for defining multi-category QoS specifications for components in distributed object systems. QML is designed to support QoS in general, encompassing QoS categories such as reliability, performance, security, and timing. QoS specification in QML facilitate the static decomposition of a software system into components with precisely specified QoS boundaries. They also facilitate dynamic QoS functions, such as negotiations, monitoring, and adaptation. QML is designed for a good fit with object-oriented distributed architectures and concepts such as interfaces and inheritance. It also allows specification at a fine-grained level for operations, operation arguments, and attributes. QML enables user- defined QoS categories, and allows specifications within those categories to be associated with* *component interface definitions. In addition, checks can be made dynamically to determine whether one QML specification satisfies another. This mechanism allows us to dynamically match QoS requirements and offers*

6 Publications in **Automaton-Based Techniques**

Articles in Refereed Journals:

1. Christel Baier, Marjan Sirjani, Farhad Arbab, Jan J. M. M. Rutten. *Modeling component connectors in Reo by constraint automata.* Sci. Comput. Program., Vol. 61, Num. 2, pages 75–113, 2006.

   Abstract: *In this paper we introduce constraint automata and propose them as an operational model for Reo, an exogenous coordination language for compositional construction of component connectors based on a calculus of channels. By providing composition operators for constraint automata and defining notions of equivalence and refinement relations for them, this paper covers the foundations for building tools to address concerns such as the automated construction of the automaton for a given component connector, equivalence checking or containment checking of the behavior of two given connectors, and verification of coordination mechanisms.*

2. George A. Papadopoulos, Farhad Arbab. *Coordination Models and Languages.* Advances in Computers, Vol. 46, pages 330–401, 1998.

   Abstract: *A new class of models, formalisms and mechanisms has recently evolved for describing concurrent and distributed computations based on the concept of "coordination". The purpose of a coordination model and associated language is to provide a means of integrating a number of possibly heterogeneous components together, by interfacing with each component in such a way that the collective set forms a single application that can execute on and take advantage of parallel and distributed systems. In this chapter we initially define and present in sufficient detail the fundamental concepts of what constitutes a coordination model or language. We then go on to classify these models and languages as either "data-driven" or "control-driven" (also called "process-" or "task-oriented"). Next, the main existing coordination models and languages are described in sufficient detail to let the reader appreciate their features and put them into perspective with respect to each other. The chapter ends with a discussion comparing the various models and some conclusions.*

Articles in Refereed Conferences:

1. Aysu Betin-Can, Tevfik Bultan, Xiang Fu. *Design for verification for asynchronously communicating Web services.* WWW, pages 750–759, 2005.

   Abstract: *We present a design for verification approach to developing reliable web services. We focus on composite web services which consist of asynchronously communicating peers. Our goal is to automatically verify properties of interactions among such peers. We propose a design pattern that eases the development of such web services and enables a modular, assume-guarantee style verification strategy. In the proposed design pattern, each peer is associated with a behavioral interface description which specifies how that peer will interact with other peers. Using these peer interfaces we automatically generate BPEL specifications to publish for interoperability. Assuming that the participating peers behave according to their interfaces, we verify safety and liveness properties about the global behavior of the composite web service during behavior verification. During interface verification, we check that each peer implementation conforms to its interface. Using the modularity in the proposed design pattern, we are able to perform the interface verification of each peer and the behavior verification as separate steps. Our experiments show that, using this modular approach, one can automatically and efficiently verify web service implementations.*

2. Foster, H., Uchitel, S., Magee, J., Kramer, J.. *Compatibility verification for Web service choreography.* pages 738–741, July 2004.

   Abstract: *In this paper we discuss a model-based approach to verifying process interactions for coordinated Web service compositions. The approach uses finite state machine representations of Web service orchestrations and assigns semantics to the distributed process interactions. The move towards implementing Web service compositions by multiple interested parties as a form of distributed system architecture motivates* *the need for supporting compatibility verification of activities and transactions in all the processes. The described approach is supported by a suite of cooperating tools for specification, formal modeling and providing verification results from orchestrated Web service interactions.*

3. Tevfik Bultan, Xiang Fu, Jianwen Su. *Tools for Automated Verification of Web Services.* ATVA, pages 8–10, 2004.

4. Xiang Fu, Tevfik Bultan, Jianwen Su. *Analysis of Interacting BPEL Web Services.* WWW'04: Proceedings of the 13th International Conference on World Wide Web, pages 621–630, ACM Press, 2004.

   Abstract: *This paper presents a set of tools and techniques for analyzing interactions of composite web services which are specified in BPEL and communicate through asynchronous XML messages. We model the interactions of composite web services as conversations, the global sequence of messages ex- changed by the web services. As opposed to earlier work, our tool-set handles rich data manipulation via XPath expressions. This allows us to verify designs at a more detailed level and check properties about message content. We present a framework where BPEL specifications of web ser- vices are translated to an intermediate representation, followed by the translation of the intermediate representation to a verification language. As an intermediate representation we use guarded automata augmented with unbounded queues for incoming messages, where the guards are expressed as XPath expressions. As the target verification language we use Promela, input language of the model checker SPIN. Since SPIN model checker is a finite-state verification tool we can only achieve partial verification by fixing the sizes of the input queues in the translation. We propose the concept of synchronizability to address this problem. We show that if a composite web service is synchronizable, then its conversation set remains same when asynchronous communication is replaced with synchronous communication. We give a set of sufficient conditions that guarantee synchronizability and that can be checked statically. Based on our synchronizability results, we show that a large class of composite web services with unbounded input queues can be completely verified using a finite state model checker such as SPIN.*

5. Xiang Fu, Tevfik Bultan, Jianwen Su. *Realizability of Conversation Protocols With Message Contents.* ICWS, 2004.

   Abstract: *A conversation protocol is a top-down specification framework which specifies desired global behaviors of a web service composition. In our earlier work [A formalism for specification and verification of reactive electronic services] we studied the problem of realizability, i.e., given a conversation protocol, can a web service composition be synthesized to generate behaviors as specified by the protocol. Several sufficient realizability conditions were proposed in [A formalism for specification and verification of reactive electronic services] to ensure realizability. Conversation protocols studied in [A formalism for specification and verification of reactive electronic services], however, are essentially abstract control flows without data semantics. This paper extends the work in [A formalism for specification and verification of reactive electronic services] and achieves more accurate analysis by considering data semantics. To overcome the state-space explosion caused by the data content, we propose a symbolic analysis technique for each realizability condition. In addition, we show that the analysis of the autonomy condition can be done using an iterative refinement approach.*

6. Xiang Fu, Tevfik Bultan, Jianwen Su. *Model checking XML manipulating software.* ISSTA, pages 252–262, 2004.

   Abstract: *The use of XML as the de facto data exchange standard has allowed integration of heterogeneous web based software systems regardless of implementation platforms and programming languages. On the other hand, the rich tree-structured data representation, and the expressive XML query languages (such as XPath) make formal specification and verification of software systems that manipulate XML data a challenge. In this paper, we present our initial efforts in automated verification of XML data manipulation operations using the SPIN model checker. We present algorithms for translating (bounded) XML data and XPath expressions to Promela, the input language of SPIN. The techniques presented in this paper constitute the basis of our Web Service Analysis Tool (WSAT) which verifies LTL properties of composite web services.*

7. Laurent Fribourg, Marcos Veloso Peixoto. *Concurrent Constraint Automata.* International Logic Programming Symposium, 1993.

   Abstract: *We address the problem of the specification and the proof of properties of concurrent systems which manipulate an unbounded number of data. We propose an approach based on an extended notion of automata, called "Concurrent Constraint Automata (CCA)". A CCA is an automaton with constraints and synchronous communication. By automata with constraints, we mean a state machine whose actions and states contain parameters that take their value in the set of natural numbers. (...)*

Publications in Refereed Workshops:

1. S. Nakajima. *Model-Checking Verification for Reliable Web Service.* OOPSLA Workshop on Object-Oriented Web Services, 2002.

   Abstract: *Model-checking is a promising technique for the verification and validation of software systems. Web service, an emerging technology in the Internet, is an autonomous server that may offer an individual service. It sometimes requires to combine more than one to meet our requirements. WSFL(Web Services Flow Language) is proposed as a language to provide means to describe Web service aggregation. We are interested in how much the software model-checking technique can be used as a basis for raising reliability of Web service, Web service flow descriptions in particular. Our experience shows that faulty flow descriptions can be identified with the proposed method. The method is also very helpful in studying an alternative semantics of the WSFL in regard to the handling of dataflows.*

7 Publications in **Bussiness Process Modeling**

Articles in Refereed Journals:

1. Chun Ouyanga, Eric Verbeekb, Wil M.P. van der Aalst, Stephan Breutel, Marlon Dumas, Arthur H.M. ter Hofstede. *Formal semantics and analysis of control flow in WS-BPEL.* Science of Computer Programming, Vol. 67, Num. 2-3, pages 162–198, April 2007.
   Abstract: *Web service composition refers to the creation of new (Web) services by combining functionalities provided by existing ones. A number of domain-specific languages for service composition have been proposed, with consensus being formed around a process-oriented language known as WS-BPEL (or BPEL). The kernel of BPEL consists of simple communication primitives that may be combined using control-flow constructs expressing sequence, branching, parallelism, synchronization, etc. We present a comprehensive and rigorously defined mapping of BPEL constructs onto Petri net structures, and use this for the analysis of various dynamic properties related to unreachable activities, conflicting messages, garbage collection, conformance checking, and deadlocks and lifelocks in interaction processes. We use a mapping onto Petri nets because this allows us to use existing theoretical results and analysis tools. Unlike approaches based on finite state machines, we do not need to construct the state space, and can use structural analysis (e.g., transition invariants) instead. We have implemented a tool that translates BPEL processes into Petri nets and then applies Petri-net-based analysis techniques. This tool has been tested on different examples, and has been used to answer a variety of questions.*

2. Roberto Lucchi, Manuel Mazzara. *A pi-calculus based semantics for WS-BPEL .* Journal of Logic and Algebraic Programming, Vol. 70, Num. 1, pages 96–118, January 2007.
   Abstract: *Recently, the term Web services orchestration has been introduced to address some issues related to Web services composition, that is the way of defining a complex service out of simpler ones. Several proposals for describing orchestration for business processes have been presented in the last years and many of these languages make use of concepts as long-running transactions and compensations for coping with error handling. WS-BPEL 2.0, the most credited candidate for becoming a standard, provides three different mechanisms allowing to cope with abnormal situations: exception, event and compensation handling. This complexity makes it difficult to formally define the framework, thus limiting the formal reasoning about the designed applications. In this paper we advocate that three different mechanisms for error handling are not necessary and we formalize a novel orchestration language based on the idea of event notification as the unique error handling mechanism. To this end, we formally define the three BPEL mechanisms in terms of our calculus. It is possible to take advantages of this formal description in two ways. Firstly, this language represents by itself a proposal of simplification for WS-BPEL 2.0 including an unambiguous specification. Secondly, an implementor of an actual WS-BPEL 2.0 orchestration engine could implement simply this single mechanism providing all the remaining ones by compilation. With this attempt we intend to give a concrete contribute towards the improvement of the quality of the BPEL specification, the applicability of BPEL itself and the implementation of real orchestration engines. Finally, as a case study we consider some of the hundreds of open issues met by the WS-BPEL designers and we propose a solution making use of the experience gained developing our algebra.*

3. George A. Papadopoulos, Farhad Arbab. *Coordination Models and Languages.* Advances in Computers, Vol. 46, pages 330–401, 1998.
   Abstract: *A new class of models, formalisms and mechanisms has recently evolved for describing concurrent and distributed computations based on the concept of "coordination". The purpose of a coordination model and associated language is to provide a means of integrating a number of possibly heterogeneous components together, by interfacing with each component in such a way that the collective set forms a single application that can execute on and take advantage of parallel and distributed systems. In this chapter we initially define and present in sufficient detail the fundamental concepts of what constitutes a coordination model or language. We then go on to classify these models and languages as either "data-driven" or "control-driven" (also called "process-" or "task-oriented"). Next, the main existing coordination models and languages are described in sufficient detail to let the reader appreciate their features and put them into perspective*

with respect to each other. The chapter ends with a discussion comparing the various models and some conclusions.

Articles in Refereed Conferences:

1. A. Rozinat, W.M.P. van der Aalst. *Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models.* Business Process Management Workshops, Lecture Notes in Computer Science, Vol. 3812, pages 163–176, February 2006.

   Abstract: *Most information systems log events (e.g., transaction logs, audit trails) to audit and monitor the processes they support. At the same time, many of these processes have been explicitly modeled. For example, SAP R/3 logs events in transaction logs and there are EPCs (Event-driven Process Chains) describing the so-called reference models. These reference models describe how the system should be used. The coexistence of event logs and process models raises an interesting question: "Does the event log conform to the process model and vice versa?". This paper demonstrates that there is not a simple answer to this question. To tackle the problem, we distinguish two dimensions of conformance: fitness (the event log may be the result of the process modeled) and appropriateness (the model is a likely candidate from a structural and behavioral point of view). Different metrics have been defined and a Conformance Checker has been implemented within the ProM Framework.*

2. Wil M. P. van der Aalst, Marlon Dumas, Chun Ouyang, Anne Rozinat, H. M. W. Verbeek. *Choreography Conformance Checking: An Approach based on BPEL and Petri Nets.* The Role of Business Processes in Service Oriented Architectures, 2006.

   Abstract: *Recently, languages such as BPEL and CDL have been proposed to describe the way services can interact from a behavioral perspective. 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 paper investigates a possible usage of behavioral service descriptions, namely as input for conformance checking. Conformance checking is the act of verifying whether one or more parties stick to the agreed-upon behavior by observing the actual behavior, e.g., the exchange of messages between all parties. This paper shows that it is possible to translate BPEL business protocols to Petri nets and to relate SOAP messages to transitions in the Petri net. As a result, Petri net-based conformance checking techniques can be used to quantify fitness (whether the observed behavior is possible in the business protocol) and appropriateness (whether the observed behavior "overfits" or "underfits" the business protocol). Moreover, non-conformance can be visualized to pinpoint deviations. The approach has been implemented in the context of the ProM n framework.*

3. Marina Mongiello, Daniela Castelluccia. *Modelling and verification of BPEL business processes.* Fourth Workshop on Model-Based Development of Computer-Based Systems, 2006.

   Abstract: *A business process is a complex Web service with functions provided by different Web services, which are already existing in Web and are dynamically integrated for granting a more complex business task. For this reason, business processes have become more and more diffuse in B2B and B2C domains, so that the importance of their activities asks for a high-level of reliability. Methods and tools for supporting automatic system verification and validation could be useful. Among the techniques of automatic verification, we choose model checking method, because we applied it efficiently for verification of a single Web service and in this paper we extend the area of application also in business processes. Descriptions of the behavior of a business process are coded using a standard language, BPEL4WS, that has broadly spread because it is able to describe a business process as both an executable process and an abstract process. Therefore, we model a BPEL description of a generic business process with a formal model and we formalize correctness properties about the reliability of the business process design. Also, we build a framework that performs automatic verification of formal models of business processes through NuSMV model checker. If there is a violation of correctness specifications, NuSMV provides counter-examples, so we can locate errors and effect right changes for correcting business process design.*

4. Florian Rosenberg, Schahram Dustdar. *Business Rules Integration in BPEL - A Service-Oriented Approach.* CEC, pages 476–479, 2005.

Abstract: *Business rules change quite often. These changes cannot be handled efficiently by representing business rules embedded in the source code of the business logic. Efficient handling of rules that govern ones business is one factor for success. That is where business rules engines play an important role. The service-oriented computing paradigm is becoming more and more popular. Services offered by different providers, are composed to new services by using Web service composition languages such as BPEL. Such process-based composition languages lack the ability to use business rules managed by different business rules engines in the composition process. In this paper, we propose an approach on how to use and integrate business rules in a service-oriented way into BPEL.*

5. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Mathias Weske. *Business Process Management: A Survey .* International Conference on Business Process Management (BPM), Lecture Notes in Computer Science, Vol. 2678, Springer Verlag, 2003.

   Abstract: *Business Process Management (BPM) includes methods, techniques, and tools to support the design, enactment, management, and analysis of operational business processes. It can be considered as an extension of classical Workflow Management (WFM) systems and approaches. Although the practical relevance of BPM is undisputed, a clear definition of BPM and related acronyms such as BAM, BPA, and STP are missing. Moreover, a clear scientific foundation is missing. In this paper, we try to demystify the acronyms in this domain, describe the state-of-the-art technology, and argue that BPM could benefit from formal methods/languages (cf. Petri nets, process algebras, etc.).*

8 Publications in **Bussiness Process Execution Language**

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

2. Bianculli, D., Ghezzi, C., Spoletini, P.. *A Model Checking Approach to Verify BPEL4WS Workflows.* Service-Oriented Computing and Applications, 2007. SOCA '07. IEEE International Conference on, To Appear, June 2007.

   Abstract: *The increasing diffusion of service oriented computing in critical business transactions demands reliability and correctness of the workflow logic representing web service orchestrations. We present an approach for the formal verification of workflow-based compositions of web services, described in BPEL4WS. Workflow processes can be verified in isolation, assuming that the external services invoked are known only through their interface. It is also possible to verify that the actual composition of two or more processes behaves correctly. We can verify deadlock freedom, properties expressed as data-bound assertions written in WS-CoL, a specification language for web services, and LTL temporal properties. Our approach is based on the software model checker Bogor, whose language supports the modeling of all BPEL4WS constructs. We provide an empirical evaluation of our approach and we compare the results with other BPEL4WS model checking tools.*

3. Chun Ouyanga, Eric Verbeekb, Wil M.P. van der Aalst, Stephan Breutel, Marlon Dumas, Arthur H.M. ter Hofstede. *Formal semantics and analysis of control flow in WS-BPEL.* Science of Computer Programming, Vol. 67, Num. 2-3, pages 162–198, April 2007.

   Abstract: *Web service composition refers to the creation of new (Web) services by combining functionalities provided by existing ones. A number of domain-specific languages for service composition have been proposed, with consensus being formed around a process-oriented language known as WS-BPEL (or BPEL). The kernel of BPEL consists of simple communication primitives that may be combined using control-flow constructs expressing sequence, branching, parallelism, synchronization, etc. We present a comprehensive and rigorously defined mapping of BPEL constructs onto Petri net structures, and use this for the analysis of various dynamic properties related to unreachable activities, conflicting messages, garbage collection, conformance checking, and deadlocks and lifelocks in interaction processes. We use a mapping onto Petri nets because this allows us to use existing theoretical results and analysis tools. Unlike approaches based on finite state machines, we do not need to construct the state space, and can use structural analysis (e.g., transition invariants) instead. We have implemented a tool that translates BPEL processes into Petri nets and then applies Petri-net-based analysis techniques. This tool has been tested on different examples, and has been used to answer a variety of questions.*

4. J. Cardoso. *Complexity analysis of BPEL Web processes.* Software Process: Improvement and Practice, Vol. 12, Num. 1, pages 35–49, Whiley-Interscience, 2007.

   Abstract: *Several organizations have already realized the potential of using WS-BEPL, the Process Execution Language for Web Services, to model the behavior of Web services in business processes. WS-BPEL provides a model for describing simple or complex interactions between business partners. In some cases, WS-BPEL process designs can be highly complex, due, for example, to the vast number of Web services carried out in global markets. High complexity in a process has several undesirable drawbacks; it may result in poor understandability, more errors, defects, and exceptions, leading to processes requiring more time to be developed, tested, and maintained. Therefore, excessive complexity should be avoided. Processes that are highly complex tend be less flexible, since it is more complicated to make changes to the process. The major goal of this article is to present two metrics to analyze the control-flow complexity (CFC) of WS-BPEL Web processes. The metrics are to be used at design-time to evaluate the complexity of a process design before implementation actually takes place.*

5. James Pasley. *How BPEL and SOA Are Changing Web Services Development.* IEEE Internet Computing, Vol. 9, Num. 3, pages 60–67, 2005.

   Abstract: *As the use of Web services grows, organizations are increasingly choosing the Business Process Execution Language for modeling business processes within the Web services architecture. In addition to orchestrating organizations' Web services, BPEL's strengths include asynchronous message handling, reliability, and recovery. By developing Web services with BPEL in mind, organizations can implement aspects of the service-oriented architecture that might previously have been difficult to achieve.*

6. Xiulan Yu, Long Zhang, Ying Li, Ying Chen. *WSCE: a flexible Web Service Composition Environment.* Web Services, 2004. Proceedings. IEEE International Conference on, To Appear, July 2004.

   Abstract: *In this paper, we propose the concepts of virtual partner and inspector into the Web services composition. Virtual partner, as an IT level concept, is a Web service (pseudo Web service) using the same interface with the actual partner but different binding message. A virtual partner can be invoked directly by a business process described by BPEL, so that the BPEL programmer can test both application's functionality and non functionality performance early in the development cycle to avoid any problems in the final runtime, or test the selection of their partners in business level design. The IT virtual partners provide developers with a range of the techniques which let them explore every aspect of their program. Inspector is proposed when using the third-party process engine. An inspector itself is also a Web service. The programmer can register any required output information in it. The IT virtual partner and the inspector concepts have been integrated in our WSCE, a flexible Web Services Composition Environment for a business process. WSCE is a prototype of autonomic modeling and simulation environment. With the help of a third-party BPEL engine, it provides programmer with concepts and tools to facilitate business process programming.*

7. Marco Pistore, Marco Roveri, Paolo Busetta. *Requirements-Driven Verification of Web Services.* Electr. Notes Theor. Comput. Sci., Vol. 105, pages 95–108, 2004.

   Abstract: *We propose a requirements-driven approach to the design and verification of Web services. The proposed methodology starts from a requirements model, which defines a business domain at a "strategic" level, describing the participating actors, their mutual dependencies, goals, requirements, and expectations. This business requirements model is then refined into a business process model. In this refinement, definitions of the processes carried out by the actors of the domain are added to the model in the form of BPEL4WS code. We show how to exploit model checking techniques for the verification of the specification, both at the requirements and at the process level. At the requirements level, model checking is used to validate the specification against a set of queries specified by the designer; at the process level, it is used to verify if the BPEL4WS processes satisfy the constraints described in the requirements model.*

8. W. van der Aalst. *Don't go with the flow: Web services composition standards exposed.* IEEE Intelligent Systems, To Appear, Jan/Feb 2003.

   Abstract: *The recently released Business Process Execution Language for Web Services (BPEL4WS) is said to combine the best of other standards for web services composition such as WSFL from IBM and*

XLANG of Microsoft. BPEL4WS allows for a mixture of block structured and graph structured process models thus making the language expressive at the price of being complex. Although BPEL4WS is not such a bad proposal by itself, it is remarkable how much attention this standard receives while the more fundamental issues and problems such as semantics, expressiveness, and adequacy do not get the attention they deserve. Having a standard is a very good idea. However, there are too many of them and most of them die before becoming mature. A simple indicator of this development is the increasing length of acronyms: PDL, XPDL, BPSS, EDOC, BPML, WSDL, WSCI, ebXML, and BPEL4WS are just some of the acronyms referring to various standards in the domain. Another problem is that these languages typically have no clearly defined semantics. The only way to overcome these problems is to critically evaluate the so-called standards for web services composition, i.e., Don't go with the flow!

Articles in Refereed Conferences:

1. Moreno Marzolla, Raffaela Mirandola. *Performance Prediction of Web Service Workflows.* Software Architectures, Components, and Applications, Lecture Notes in Computer Science, Vol. 4880, pages 127–144, January 2008.

   Abstract: *Web Services play an important role in the SOA paradigm, as they allow services to be selected on-the-fly to build applications out of existing components. In this scenario, the BPEL notation can be used as an orchestration language which allows the user to describe interactions with Web Services in a standard way. The performance of a BPEL workflow is a very important factor for deciding which components must be selected, or to choose whether a given sequence of interactions can provide the requested quality of service. Due to its very dynamic nature, workflow performance evaluation can not be accomplished using traditional, heavy-weight techniques. In this paper we present a multi-view approach for the performance prediction of service-based applications encompassing both users and service provider(s) perspectives. As a first step towards the realization of this integrated framework we present an efficient approach for performance assessment of Web Service workflows described using the BPEL notation. Starting from annotated BPEL and WSDL specifications, we derive performance bounds on response time and throughput. In such a way users are able to assess the efficiency of a BPEL workflow, while service provider(s) can perform sizing studies or estimate performance gains of alternative upgrades to existing systems. To bring this approach to fruition we developed a prototype tool called bpel2qnbound, using which we analyze a simple case study.*

2. Sheng Chen, Liang Bao, Ping Chen. *OptBPEL: A Tool for Performance Optimization of BPEL Process.* Software Composition, pages 141–148, 2008.

   Abstract: *The Business Process Execution Language (BPEL) is now a de facto standard for specifying and executing business process for web service composition and orchestration. As more and more web services are composed using BPEL, tuning these compositions and gain better performance becomes increasingly important. This paper presents our approach for optimizing the BPEL process and introduces OptBPEL, a tool for performance optimization of BPEL process. The approach starts from the optimization of synchronization structure concerning link in BPEL. After that, some concurrency analysis techniques are applied to obtain further performance improvement. Finally, we give some experiments and prove the efficiency of these optimization algorithms used in OptBPEL.*

3. Gero Decker, Oliver Kopp, Frank Leymann, Mathias Weske. *BPEL4Chor: Extending BPEL for Modeling Choreographies.* ICWS, pages 296–303, 2007.

   Abstract: *The Business Process Execution Language (BPEL) is a language to orchestrate web services into a single business process. In a choreography view, several processes are interconnected and their interaction behavior is described from a global perspective. This paper shows how BPEL can be extended for defining choreographies. The proposed extensions (BPEL4Chor) distinguish between three aspects: (i) participant behavior descriptions, i.e. control flow dependencies in each participant, (ii) the participant topology, i.e. the existing participants and their interconnection using message links and (iii) participant groundings, i.e. concrete configurations for data formats and port types. As BPEL itself is used unchanged, the extensions facilitate a seamless integration between service choreographies and orchestrations. The suitability of the extensions is validated by assessing their support for the Service Interaction Patterns.*

4. Dmytro Rud, Andreas Schmietendorf, Reiner Dumke. *Performance Modeling of WS-BPEL-Based Web Service Compositions.* Services Computing Workshops, pages 140–147, September 2006.

   Abstract: *This paper addresses quality of service aspects of Web service orchestrations created using WS-BPEL from the standpoint of a Web service integrator. A mathematical model based on operations research techniques and formal semantics of WS-BPEL is proposed to estimate and forecast the influence of the execution of orchestrated processes on utilization and throughput of individual involved nodes and of the whole system. This model is applied to the optimization of service levels agreement process between the involved parties*

5. Wil M. P. van der Aalst, Marlon Dumas, Chun Ouyang, Anne Rozinat, H. M. W. Verbeek. *Choreography Conformance Checking: An Approach based on BPEL and Petri Nets.* The Role of Business Processes in Service Oriented Architectures, 2006.

   Abstract: *Recently, languages such as BPEL and CDL have been proposed to describe the way services can interact from a behavioral perspective. 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 paper investigates a possible usage of behavioral service descriptions, namely as input for conformance checking. Conformance checking is the act of verifying whether one or more parties stick to the agreed-upon behavior by observing the actual behavior, e.g., the exchange of messages between all parties. This paper shows that it is possible to translate BPEL business protocols to Petri nets and to relate SOAP messages to transitions in the Petri net. As a result, Petri net-based conformance checking techniques can be used to quantify fitness (whether the observed behavior is possible in the business protocol) and appropriateness (whether the observed behavior "overfits" or "underfits" the business protocol). Moreover, non-conformance can be visualized to pinpoint deviations. The approach has been implemented in the context of the ProM n framework.*

6. Fabio Barbon, Paolo Traverso, Marco Pistore, Michele Trainotti. *Run-Time Monitoring of Instances and Classes of Web Service Compositions.* ICWS, pages 63–71, 2006.

   Abstract: *The run-time monitoring of web service compositions has been widely acknowledged as a significant and challenging problem. In this paper, we propose a novel solution to the problem of monitoring web services implemented in BPEL. We devise an architecture that clearly separates the business logic of a web service from its monitoring functionality. The architecture supports both "instance monitors" that deal with the execution of a single instance of BPEL process, as well as "class monitors" that report aggregated information about all the instances of a BPEL process. We also define a language for the specification of instance and class monitors. The language allows for specifying boolean, statistic, and time-related properties. Finally, we devise a technique for the automatic translation of all these kinds of monitors to Java programs.*

7. Marina Mongiello, Daniela Castelluccia. *Modelling and verification of BPEL business processes.* Fourth Workshop on Model-Based Development of Computer-Based Systems, 2006.

   Abstract: *A business process is a complex Web service with functions provided by different Web services, which are already existing in Web and are dynamically integrated for granting a more complex business task. For this reason, business processes have become more and more diffuse in B2B and B2C domains, so that the importance of their activities asks for a high-level of reliability. Methods and tools for supporting automatic system verification and validation could be useful. Among the techniques of automatic verification, we choose model checking method, because we applied it efficiently for verification of a single Web service and in this paper we extend the area of application also in business processes. Descriptions of the behavior of a business process are coded using a standard language, BPEL4WS, that has broadly spread because it is able to describe a business process as both an executable process and an abstract process. Therefore, we model a BPEL description of a generic business process with a formal model and we formalize correctness properties about the reliability of the business process design. Also, we build a framework that performs automatic verification of formal models of business processes through NuSMV model checker. If there is a violation of correctness specifications, NuSMV provides counter-examples, so we can locate errors and effect right changes for correcting business process design.*

8. Yuan Yuan, Zhongjie Li, Wei Sun. *A Graph-Search Based Approach to BPEL4WS Test Generation.* ICSEA, 14 pages, 2006.

   Abstract: *Business Process Execution Language for Web Services (BPEL4WS) is a kind of concurrent programming languages with several special features that raise special challenges for verification and testing. This paper proposes a graph-search based approach to BPEL test case generation, which effectively deals with BPEL concurrency semantics. This approach defines an extension of CFG (Control Flow Graph) - BPEL Flow Graph (BFG) - to represent a BPEL program in a graphical model. Then concurrent test paths can be generated by traversing the BFG model, and test data for each path can be generated using a constraint solving method. Finally test paths and data are combined into complete test cases.*

9. Bernd-Holger Schlingloff, Axel Martens, Karsten Schmidt. *Modeling and Model Checking Web Services.* Vol. 126, pages 3–26, Elsevier, March 2005.

   Abstract: *We give an overview on web services and the web service technology stack. We then show how to build Petri net models of web services formulated in the specification language BPEL4WS. We define an abstract correctness criterion for these models and study the automated verification according to this criterion. Finally, we relate correctness of web service models to the model checking problem for alternating temporal logics.*

10. Jorge Cardoso. *About the Data-Flow Complexity of Web Processes.* 6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility, pages 67–74, 2005.

    Abstract: *Abstract. Organizations are increasingly faced with the challenge of managing e-commerce and e-business applications involving Web services and Web processes. In some cases, Web processes design can be highly complex, due, for example, to the vast number of services carried out in global markets. High complexity in a process has several undesirable drawbacks, it may result in bad understandability, more errors, defects, and exceptions leading processes to need more time to develop, test, and maintain. Therefore, excessive complexity should be avoided. Flexibility and complexity are guiding principles in the design of business processes. In most business processes, flexibility and complexity are related inversely. Processes with a high complexity tend be less flexible, since it is more complicated to make changes to the process. In our previous work we have defined a metric to measure the control-flow complexity of Web processes. The major goal of this paper is to study the issues and establish the requirements for the development of a metric to analyze the data-flow complexity of Web processes.*

11. Samuele Carpineti, Cosimo Laneve, Paolo Milazzo. *BoPi - A Distributed Machine for Experimenting Web Services Technologies.* ACSD, pages 202–211, 2005.

    Abstract: *BoPi is a programming language with a runtime support that allows the distribution and the execution of programs over the network. The language is a process calculus with XML values and datatypes, and with a pattern matching mechanism for deconstructing values. The compiler gives a typesafe bytecode in the form of an XML document, that may be deployed on the network. What comes out is a simple, statically typed, and formally defined core BPEL language with a basic query mechanism supplied by patterns.*

12. Dimka Karastoyanova, Alejandro Houspanossian, Mariano Cilia, Frank Leymann, Alejandro P. Buchmann. *Extending BPEL for Run Time Adaptability.* EDOC, pages 15–26, 2005.

    Abstract: *The existing Web Service Flow (WS-flow) technologies enable both static and dynamic binding of participating Web services (WSs) on the process model level. Adaptability on per-instance basis is not sufficiently supported and therefore must be addressed to improve process flexibility upon changes in the environment. Ad-hoc process instance changes can be enabled by swapping participating WS instances, by modifying port Types of the partners to be invoked, and by changing process logic. In this work we address the problem of dynamic binding of WSs to WS-flow instances at run time, i.e. the ability to exchange a WS instance participating in a WS-flow instance with an alternative one. The problem is additionally complicated by the fact that the execution of a process depends on its deployment. We describe the "find and bind" mechanism, and we show its representation as a BPEL extension. We discuss the benefits that could be gained and the disadvantages it brings in. The mechanism extends and improves the existing process technologies. It facilitates a precisely controlled policy-based selection of WSs at run time and also provides* *for process instance repair, while maintaining simplicity. We also discuss a prototypical implementation of the presented functionality.*

13. Florian Rosenberg, Schahram Dustdar. *Business Rules Integration in BPEL - A Service-Oriented Approach.* CEC, pages 476–479, 2005.

    Abstract: *Business rules change quite often. These changes cannot be handled efficiently by representing business rules embedded in the source code of the business logic. Efficient handling of rules that govern ones business is one factor for success. That is where business rules engines play an important role. The service-oriented computing paradigm is becoming more and more popular. Services offered by different providers, are composed to new services by using Web service composition languages such as BPEL. Such process-based composition languages lack the ability to use business rules managed by different business rules engines in the composition process. In this paper, we propose an approach on how to use and integrate business rules in a service-oriented way into BPEL.*

14. Jesus Arias-Fisteus, Luis Sanchez Fernandez, Carlos Delgado Kloos. *Formal Verification of BPEL4WS Business Collaborations.* E-Commerce and Web Technologies, 5th International Conference, EC-Web 2004, Proceedings, Lecture Notes in Computer Science, Vol. 3182, pages 76–85, Springer Verlag, 2004.

    Abstract: *Web services are a very appropriate communication mechanism to perform distributed business processes among several organisations. These processes should be reliable, because a failure in them can cause high economic losses. To increase their reliability at design time, we have developed VERBUS, a framework for the formal verification of business processes. VERBUS can automatically translate business process definitions to specifications verifiable in several available tools. It is based on a modular and extensible architecture: new process definition languages and verification tools can be added easily to the framework. The prototype of VERBUS presented in this work can verify BPEL4WS process specifications, by translating them to Promela. The Promela specifications are verified with the well known model checker Spin. In this paper we describe the general architecture of VERBUS and how BPEL4WS specifications are translated and verified. The explanation is completed by describing what types of properties can be verified and providing an overview of the implementation.*

15. Roozbeh Farahbod, Uwe Glässer, Mona Vajihollahi. *Specification and Validation of the Business Process Execution Language for Web Services.* Abstract State Machines, pages 78–94, 2004.

    Abstract: *We formally define an abstract executable semantics for the Business Process Execution Language for Web Services in terms of a distributed ASM. The goal of this work is to support the design and standardization of the language. There is a need for formalism. It will allow us to not only reason about the current specification and related issues, but also uncover issues that would otherwise go unnoticed. Empirical deduction is not sufficient. - Issue 42, OASIS WSBPEL TC. The language definition assumes an infrastructure for running Web services on some asynchronous communication architecture. A business process is built on top of a collection of Web services performing continuous interactions with the outside world by sending and receiving messages over a communication network. The underlying execution model is characterized by its concurrent and reactive behavior making it particularly difficult to predict dynamic system properties with a sufficient degree of detail and precision under all circumstances.*

16. Xiang Fu, Tevfik Bultan, Jianwen Su. *Analysis of Interacting BPEL Web Services.* WWW'04: Proceedings of the 13th International Conference on World Wide Web, pages 621–630, ACM Press, 2004.

    Abstract: *This paper presents a set of tools and techniques for analyzing interactions of composite web services which are specified in BPEL and communicate through asynchronous XML messages. We model the interactions of composite web services as conversations, the global sequence of messages ex- changed by the web services. As opposed to earlier work, our tool-set handles rich data manipulation via XPath expressions. This allows us to verify designs at a more detailed level and check properties about message content. We present a framework where BPEL specifications of web ser- vices are translated to an intermediate representation, followed by the translation of the intermediate representation to a verification language. As* *an intermediate representation we use guarded automata augmented with unbounded queues for incoming messages, where the guards are expressed as XPath expressions. As the target verification language we use Promela, input language of the model checker SPIN. Since SPIN model checker is a finite-state verification tool we can only achieve partial verification by fixing the sizes of the input queues in the translation. We propose the concept of synchronizability to address this problem. We show that if a composite web service is synchronizable, then its conversation set remains same when asynchronous communication is replaced with synchronous communication. We give a set of sufficient conditions that guarantee synchronizability and that can be checked statically. Based on our synchronizability results, we show that a large class of composite web services with unbounded input queues can be completely verified using a finite state model checker such as SPIN.*

17. Andreas Wombacher, Peter Fankhauser, Erich J. Neuhold. *Transforming BPEL into Annotated Deterministic Finite State Automata for Service Discovery.* ICWS, pages 316–323, 2004.
Abstract: *Web services advocate loosely coupled systems, although current loosely coupled applications are limited to stateless services. The reason for this limitation is the lack of a method supporting matchmaking of state dependent services exemplarily specified in BPEL. In particular, the sender's requirement that the receiver must support all possible messages sent at a certain state are not captured by models currently used for service discovery. Annotated deterministic finite state automata provide this expressiveness. In this paper the transformation of a local process specification given in BPEL to annotated deterministic finite state automata is presented.*

Books and Monographs:

1. Mariya Koshkina. *Verification of Business Processes for Web Services.* Ms. Thesis, York University - Department of Computer Science, October 2003.
Abstract: *The Business Process Execution Language for Web Services (BPEL4WS or simply BPEL) is a recently developed language, which is used to specify interactions between web services. Among its features it allows specification of concurrent behavior. Erroneous specification can lead to such problems as deadlock. In our research we focus on the concurrency mechanism in BPEL. Our main goal is to analyze processes in order to detect possible deadlocks. To achieve this we introduce a process algebra called the BPE-calculus. It is a small language which captures all the BPEL features relevant to the analysis. This process algebra is modelled using a labeled transition system. An existing verification tool called the Concurrency Workbench is customized to use our BPE-calculus. This tool allows us to verify many properties of BPE-calculus processes specified in a logic called the mu-calculus, including deadlock freedom.*

Publications in Refereed Workshops:

1. S. Nakajima. *Model-Checking Behavioral Specification of BPEL Applications.* Proceedings of the International Workshop on Web Languages and Formal Methods, WLFM 2005, 2005.
Abstract: *To provide a framework to compose lots of specialised services flexibly, BPEL is proposed to describe Web service flows. Since the Web service flow description is basically a distributed collaboration, writing correct programs in BPEL is not easy. Verifying BPEL program prior to its execution is essential. This paper proposes a method to extract the behavioral specification from a BPEL appliation program and to analyze it by using the SPIN model checker. With the adequate abstraction method and support for DPE, the method can analyze all the four example cases in the BPEL standard document.*

Technical Reports and Manuals:

1. Oliver Kopp, Rania Khalaf, Frank Leymann. *Reaching Definitions Analysis Respecting Dead Path Elimination Semantics in BPEL Processes.* Num. 2007/04, Institut für Architektur von Anwendungssystemen, November 2007.

Abstract: *The Business Process Execution Language for Web Services (BPEL) is a workflow language geared towards Service Oriented Computing. BPEL provides basic workflow capabilities, such as the ability to impose control on a set of activities via explicit links, as well as advanced features such as recovery and event handling. The area of workflow often requires understanding the data dependencies as well as the control dependencies between activities. This aids in business process design as well as in analysis and rengineering. BPEL, however, uses shared variables to model data. Activities read and write data to these variables, i.e. there is no explicit data link construct. In order to draw out the flow of data between the activities of a BPEL process, one must therefore perform data-flow analysis on that process. To address this problem, we present an algorithm that statically determines such data dependencies. ”Statically” means, that we do not determine the dependencies per process instance at runtime, but on the process model without creating any instances. Mainstream data-flow analysis techniques are presented in [ALSU06, Muc97, NNH04]. However, these techniques cannot be directly applied to a BPEL process, since BPEL supports both parallelism and dead path elimination (DPE, [CDG+ 03, OAS07, CKLW03, BK05, LR00]). DPE is a technique used in BPEL to propagate activity disablement down paths that can no longer be executed. The algorithm in this paper consists of reaching definitions analysis dealing with DPE, enabling it to reduce the number of data dependencies when compared to approaches not dealing with DPE.*

2. Diane Jordan, John Evdemon, Alexandre Alves, Assaf Arkin, Sid Askary, Charlton Barreto, Ben Bloch, Francisco Curbera, Mark Ford, Yaron Goland, Alejandro Guzar, Neelakantan Kartha, Canyang Kevin Liu, Rania Khalaf, Dieter Knig, Mike Marin, Vinkesh Mehta, Satish Thatte, Danny van der Rijn, Prasad Yendluri, Alex Yiu . *Web Services Business Process Execution Language Version 2.0.* IBM, Microsoft, BEA, Intalio, Individual, Adobe Systems, Systinet, Active Endpoints, JBoss, Sterling Commerce, SAP, Deloitte, TIBCO Software, webMethods, Oracle, 2007.

   Abstract: *This document defines a language for specifying business process behavior based on Web Services. This language is called Web Services Business Process Execution Language (abbreviated to WS-BPEL in the rest of this document). Processes in WS-BPEL export and import functionality by using Web Service interfaces exclusively. Business processes can be described in two ways. Executable business processes model actual behavior of a participant in a business interaction. Abstract business processes are partially specified processes that are not intended to be executed. An Abstract Process may hide some of the required concrete operational details. Abstract Processes serve a descriptive role, with more than one possible use case, including observable behavior and process template. WS-BPEL is meant to be used to model the behavior of both Executable and Abstract Processes. WS-BPEL provides a language for the specification of Executable and Abstract business processes. By doing so, it extends the Web Services interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to- business spaces.*

3. Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, Ivana Trickovic, Sanjiva Weerawarana. *Business Process Execution Language for Web Services, version 1.1.* BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems, 2003.

   Abstract: *This document defines a notation for specifying business process behavior based on Web Services. This notation is called Business Process Execution Language for Web Services (abbreviated to BPEL4WS in the rest of this document). Processes in BPEL4WS export and import functionality by using Web Service interfaces exclusively. Business processes can be described in two ways. Executable business processes model actual behavior of a participant in a business interaction. Business protocols, in contrast, use process descriptions that specify the mutually visible message exchange behavior of each of the parties involved in the protocol, without revealing their internal behavior. The process descriptions for business protocols are called abstract processes. BPEL4WS is meant to be used to model the behavior of both executable and abstract processes. BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions. BPEL4WS defines an interoperable integration model that should facilitate*

the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.

9 Publications in **Coordination Languages**

Articles in Refereed Journals:

1. Fabrizio Montesi, Claudio Guidi, Roberto Lucchi, Gianluigi Zavattaro. *JOLIE: a Java Orchestration Language Interpreter Engine.* Electr. Notes Theor. Comput. Sci., Vol. 181, pages 19–33, 2007.

   Abstract: *Service oriented computing is an emerging paradigm for programming distributed applications based on services. Services are simple software elements that supply their functionalities by exhibiting their interfaces and that can be invoked by exploiting simple communication primitives. The emerging mechanism exploited in service oriented computing for composing services -in order to provide more complex functionalities- is by means of orchestrators. An orchestrator is able to invoke and coordinate other services by exploiting typical workflow patterns such as parallel composition, sequencing and choices. Examples of orchestration languages are XLANG [IBM, ”XLANG: Web Services for Business Process Design,” http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm] and WS-BPEL [OASIS, ”Web Services Business Process Execution Language Version 2.0, Working Draft,” http://www.oasis-open.org/committees/download.php/10347/wsbpe specification-draft-120204.htm]. In this paper we present JOLIE, an interpreter and engine for orchestration programs. The main novelties of JOLIE are that it provides an easy to use development environment (because it supports a more programmer friendly C/Java-like syntax instead of an XML-based syntax) and it is based on a solid mathematical underlying model (developed in previous works of the authors [N. Busi, R. Gorrieri, C. Guidi, R. Lucchi and G. Zavattaro, Towards a formal framework for Choreography, in: Proc. of 3rd International Workshop on Distributed and Mobile Collaboration (DMC 2005) (2005), N. Busi, R. Gorrieri, C. Guidi, R. Lucchi and G. Zavattaro, Choreography and orchestration conformance for system design, in: Proc. of 8th International Conference on Coordination Models and Languages (COORDINATION'06), LNCS to appear, 2006, C. Guidi and R. Lucchi, Mobility mechanisms in service oriented computing, in: Proc. of 8th International Conference on on Formal Methods for Open Object-Based Distributed Systems (FMOODS'06), LNCS to appear, 2006]).*

2. George A. Papadopoulos, Farhad Arbab. *Coordination Models and Languages.* Advances in Computers, Vol. 46, pages 330–401, 1998.

   Abstract: *A new class of models, formalisms and mechanisms has recently evolved for describing concurrent and distributed computations based on the concept of ”coordination”. The purpose of a coordination model and associated language is to provide a means of integrating a number of possibly heterogeneous components together, by interfacing with each component in such a way that the collective set forms a single application that can execute on and take advantage of parallel and distributed systems. In this chapter we initially define and present in sufficient detail the fundamental concepts of what constitutes a coordination model or language. We then go on to classify these models and languages as either ”data-driven” or ”control-driven” (also called ”process-” or ”task-oriented”). Next, the main existing coordination models and languages are described in sufficient detail to let the reader appreciate their features and put them into perspective with respect to each other. The chapter ends with a discussion comparing the various models and some conclusions.*

Articles in Refereed Conferences:

1. Johannes Maria Zaha, Alistair P. Barros, Marlon Dumas, Arthur H. M. ter Hofstede. *Let's Dance: A Language for Service Behavior Modeling.* OTM Conferences (1), pages 145–162, 2006.

   Abstract: *In Service-Oriented Architectures (SOAs), software systems are decomposed into independent units, namely services, that interact with one another through message exchanges. To promote reuse and evolvability, these interactions are explicitly described right from the early phases of the development lifecycle. Up to now, emphasis has been placed on capturing structural aspects of service interactions. Gradually though, the description of behavioral dependencies between service interactions is gaining increasing attention as a means to push forward the SOA vision. This paper deals with the description of these behavioral* *dependencies during the analysis and design phases. The paper outlines a set of requirements that a language for modeling service interactions at this level should fulfill, and proposes a language whose design is driven by these requirements.*

2. Alistair P. Barros, Marlon Dumas, Arthur H. M. ter Hofstede. *Service Interaction Patterns.* Business Process Management, pages 302–318, 2005.

   Abstract: *With increased sophistication and standardization of modeling languages and execution platforms supporting business process management (BPM) across traditional boundaries, has come the need for consolidated insights into their exploitation from a business perspective. Key technology developments in BPM bear this out, with several web services-related initiatives investing significant effort in the collection of compelling use cases to heighten the exploitation of BPM in multi-party collaborative environments. In this setting, we present a collection of patterns of service interactions which allow emerging web services functionality, especially that pertaining to choreography and orchestration, to be benchmarked against abstracted forms of representative scenarios. Beyond bilateral interactions, these patterns cover multilateral, competing, atomic and causally related interactions. Issues related to the implementation of these patterns using established and emerging web services standards, most notably BPEL, are discussed.*

3. Samuele Carpineti, Cosimo Laneve, Paolo Milazzo. *BoPi - A Distributed Machine for Experimenting Web Services Technologies.* ACSD, pages 202–211, 2005.

   Abstract: *BoPi is a programming language with a runtime support that allows the distribution and the execution of programs over the network. The language is a process calculus with XML values and datatypes, and with a pattern matching mechanism for deconstructing values. The compiler gives a typesafe bytecode in the form of an XML document, that may be deployed on the network. What comes out is a simple, statically typed, and formally defined core BPEL language with a basic query mechanism supplied by patterns.*

10 Publications in **Formal Methods**

Articles in Refereed Journals:

1. Chun Ouyanga, Eric Verbeekb, Wil M.P. van der Aalst, Stephan Breutel, Marlon Dumas, Arthur H.M. ter Hofstede. *Formal semantics and analysis of control flow in WS-BPEL.* Science of Computer Programming, Vol. 67, Num. 2-3, pages 162–198, April 2007.

   Abstract: *Web service composition refers to the creation of new (Web) services by combining functionalities provided by existing ones. A number of domain-specific languages for service composition have been proposed, with consensus being formed around a process-oriented language known as WS-BPEL (or BPEL). The kernel of BPEL consists of simple communication primitives that may be combined using control-flow constructs expressing sequence, branching, parallelism, synchronization, etc. We present a comprehensive and rigorously defined mapping of BPEL constructs onto Petri net structures, and use this for the analysis of various dynamic properties related to unreachable activities, conflicting messages, garbage collection, conformance checking, and deadlocks and lifelocks in interaction processes. We use a mapping onto Petri nets because this allows us to use existing theoretical results and analysis tools. Unlike approaches based on finite state machines, we do not need to construct the state space, and can use structural analysis (e.g., transition invariants) instead. We have implemented a tool that translates BPEL processes into Petri nets and then applies Petri-net-based analysis techniques. This tool has been tested on different examples, and has been used to answer a variety of questions.*

2. Roberto Lucchi, Manuel Mazzara. *A pi-calculus based semantics for WS-BPEL .* Journal of Logic and Algebraic Programming, Vol. 70, Num. 1, pages 96–118, January 2007.

   Abstract: *Recently, the term Web services orchestration has been introduced to address some issues related to Web services composition, that is the way of defining a complex service out of simpler ones. Several proposals for describing orchestration for business processes have been presented in the last years and many of these languages make use of concepts as long-running transactions and compensations for coping with error handling. WS-BPEL 2.0, the most credited candidate for becoming a standard, provides three different mechanisms allowing to cope with abnormal situations: exception, event and compensation handling. This complexity makes it difficult to formally define the framework, thus limiting the formal reasoning about the designed applications. In this paper we advocate that three different mechanisms for error handling are not necessary and we formalize a novel orchestration language based on the idea of event notification as the unique error handling mechanism. To this end, we formally define the three BPEL mechanisms in terms of our calculus. It is possible to take advantages of this formal description in two ways. Firstly, this language represents by itself a proposal of simplification for WS-BPEL 2.0 including an unambiguous specification. Secondly, an implementor of an actual WS-BPEL 2.0 orchestration engine could implement simply this single mechanism providing all the remaining ones by compilation. With this attempt we intend to give a concrete contribute towards the improvement of the quality of the BPEL specification, the applicability of BPEL itself and the implementation of real orchestration engines. Finally, as a case study we consider some of the hundreds of open issues met by the WS-BPEL designers and we propose a solution making use of the experience gained developing our algebra.*

3. Christel Baier, Marjan Sirjani, Farhad Arbab, Jan J. M. M. Rutten. *Modeling component connectors in Reo by constraint automata.* Sci. Comput. Program., Vol. 61, Num. 2, pages 75–113, 2006.

   Abstract: *In this paper we introduce constraint automata and propose them as an operational model for Reo, an exogenous coordination language for compositional construction of component connectors based on a calculus of channels. By providing composition operators for constraint automata and defining notions of equivalence and refinement relations for them, this paper covers the foundations for building tools to address concerns such as the automated construction of the automaton for a given component connector, equivalence checking or containment checking of the behavior of two given connectors, and verification of coordination mechanisms.*

4. Mariya Koshkina, Franck van Breugel. *Modelling and verifying web service orchestration by means of the concurrency workbench.* SIGSOFT Softw. Eng. Notes, Vol. 29, Num. 5, pages 1–10, ACM, 2004.

   Abstract: *Verification techniques like model checking, preorder checking and equivalence checking are shown to be relevant to web service orchestration. The Concurrency Workbench of the New Century (CWB) is a verification tool that supports these verification techniques. By means of the Process Algebra Compiler (PAC), the CWB is modified to support the BPE-calculus. The BPE-calculus is a small language, based on BPEL4WS, to express web service orchestration. Both the syntax and the semantics of the BPE-calculus are formally defined. These are subsequently used as input for the PAC. As output, the PAC produces modules that are incorporated into the CWB so that it supports the BPE-calculus and, hence, provides a verification tool for web service orchestration.*

5. Marco Pistore, Marco Roveri, Paolo Busetta. *Requirements-Driven Verification of Web Services.* Electr. Notes Theor. Comput. Sci., Vol. 105, pages 95–108, 2004.

   Abstract: *We propose a requirements-driven approach to the design and verification of Web services. The proposed methodology starts from a requirements model, which defines a business domain at a "strategic" level, describing the participating actors, their mutual dependencies, goals, requirements, and expectations. This business requirements model is then refined into a business process model. In this refinement, definitions of the processes carried out by the actors of the domain are added to the model in the form of BPEL4WS code. We show how to exploit model checking techniques for the verification of the specification, both at the requirements and at the process level. At the requirements level, model checking is used to validate the specification against a set of queries specified by the designer; at the process level, it is used to verify if the BPEL4WS processes satisfy the constraints described in the requirements model.*

6. Gian Luigi Ferrari, Stefania Gnesi, Ugo Montanari, Marco Pistore. *A model-checking verification environment for mobile processes.* ACM Trans. Softw. Eng. Methodol., Vol. 12, Num. 4, pages 440–473, 2003.

   Abstract: *This article presents a semantic-based environment for reasoning about the behavior of mobile systems. The verification environment, called HAL, exploits a novel automata-like model that allows finite-state verification of systems specified in the π-calculus. The HAL system is able to interface with several efficient toolkits (e.g. model-checkers) to determine whether or not certain properties hold for a given specification. We report experimental results on some case studies.*

Articles in Refereed Conferences:

1. A. Rozinat, W.M.P. van der Aalst. *Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models.* Business Process Management Workshops, Lecture Notes in Computer Science, Vol. 3812, pages 163–176, February 2006.

   Abstract: *Most information systems log events (e.g., transaction logs, audit trails) to audit and monitor the processes they support. At the same time, many of these processes have been explicitly modeled. For example, SAP R/3 logs events in transaction logs and there are EPCs (Event-driven Process Chains) describing the so-called reference models. These reference models describe how the system should be used. The coexistence of event logs and process models raises an interesting question: "Does the event log conform to the process model and vice versa?". This paper demonstrates that there is not a simple answer to this question. To tackle the problem, we distinguish two dimensions of conformance: fitness (the event log may be the result of the process modeled) and appropriateness (the model is a likely candidate from a structural and behavioral point of view). Different metrics have been defined and a Conformance Checker has been implemented within the ProM Framework.*

2. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, Marco Montali, Sergio Storari, Paolo Torroni. *Computational Logic for Run-Time Verification of Web Services Choreographies: Exploiting the OCS-SI Tool.* WS-FM, pages 58–72, 2006.

   Abstract: *In this work, we investigate the feasibility of using a framework based on computational logic, and mainly defined in the context of Multi-Agent Systems for Global Computing (SOCS UE Project), for* *modeling choreographies of Web Services with respect to the conversational aspect. One of the fundamental motivations of using computational logic, beside its declarative and highly expressive nature, is given by its operational counterpart, that can provide a proof-theoretic framework able to verify the consistency of services designed in a cooperative and incremental manner. In particular, in this paper we show that suitable "Social Integrity Constraints", introduced in the SOCS social model, can be used for specifying global protocols at the choreography level. In this way, we can use a suitable tool, derived from the proof-procedure defined in the context of the SOCS project, to check at run-time whether a set of existing services behave in a conformant manner w.r.t. the defined choreography.*

3. Patricia Bouyer, Pierre-Alain Reynier, Serge Haddad. *Extended Timed Automata and Time Petri Nets.* ACSD, pages 91–100, 2006.

   Abstract: *Timed Automata (TA) and Time Petri Nets (TPN) are two well-established formal models for real-time systems. Recently, a linear transformation of TA to TPNs preserving reachability properties and timed languages has been proposed, which does however not extend to larger classes of TA which would allow diagonal constraints or more general resets of clocks. Though these features do not add expressiveness, they yield exponentially more concise models. In this work, we propose two translations: one from extended TA to TPNs whose size is either linear or quadratic in the size of the original TA, depending on the features which are allowed; another one from a parallel composition of TA to TPNs, which is also linear. As a consequence, we get that TPNs are exponentially more concise than TA.*

4. Gero Decker, Johannes Maria Zaha, Marlon Dumas. *Execution Semantics for Service Choreographies.* WS-FM, pages 163–177, 2006.

   Abstract: *A service choreography is a model of interactions in which a set of services engage to achieve a goal, seen from the perspective of an ideal observer that records all messages exchanged between these services. Choreographies have been put forward as a starting point for building service-oriented systems since they provide a global picture of the system's behavior. In previous work we presented a language for service choreography modeling targeting the early phases of the development lifecycle. This paper provides an execution semantics for this language in terms of a mapping to PI-calculus. This formal semantics provides a basis for analyzing choreographies. The paper reports on experiences using the semantics to detect unreachable interactions.*

5. Howard Foster, Sebastian Uchitel, Jeff Magee, Jeff Kramer. *LTSA-WS: a tool for model-based verification of web service compositions and choreography.* ICSE '06: Proceedings of the 28th international conference on Software engineering, pages 771–774, ACM, 2006.

   Abstract: *In this paper we describe a tool for a model-based approach to verifying compositions of web service implementations. The tool supports verification of properties created from design specifications and implementation models to confirm expected results from the viewpoints of both the designer and implementer. Scenarios are modeled in UML, in the form of Message Sequence Charts (MSCs), and then compiled into the Finite State Process (FSP) process algebra to concisely model the required behavior. BPEL4WS implementations are mechanically translated to FSP to allow an equivalence trace verification process to be performed. By providing early design verification and validation, the implementation, testing and deployment of web service compositions can be eased through the understanding of the behavior exhibited by the composition. The approach is implemented as a plug-in for the Eclipse development environment providing cooperating tools for specification, formal modeling, verification and validation of the composition process.*

6. Raman Kazhamiakin, Paritosh K. Pandya, Marco Pistore. *Representation, Verification, and Computation of Timed Properties in Web Service Compositions.* ICWS, pages 497–504, 2006.

   Abstract: *In this paper we address the problem of qualitative and quantitative analysis of timing aspects of Web service compositions defined as a set of BPEL4WS processes. We introduce a formalism, called Web Service Timed State Transition Systems (WSTTS), to capture the timed behavior of the composite web services. We also exploit an interval temporal logic to express complex timed assumptions and requirements on the system's behavior. Building on top of this formalization, we provide techniques and tools for modelchecking BPEL4WS compositions against time-related requirements. We also present a symbolic algorithm that* *can be used to compute duration bounds of behavioral intervals that satisfy such requirements. We perform a preliminary experimental evaluation of our approach and tools with the help of an e-Government case study.*

7. Mohsen Rouached, Olivier Perrin, Claude Godart. *Towards Formal Verification of Web Service Composition.* Business Process Management, pages 257–273, 2006.

   Abstract: *Web services composition is an emerging paradigm for enabling application integration within and across organizational boundaries. Current Web services composition proposals, such as BPML, WS-BPEL, WSCI, and OWL-S, provide solutions for describing the control and data flows in Web service composition. However, such proposals remain at the descriptive level, without providing any kind of mechanisms or tool support for analysis and verification. Therefore, there is a growing interest for the verification techniques which enable designers to test and repair design errors even before actual running of the service, or allow designers to detect erroneous properties and formally verify whether the service process design does have certain desired properties. In this paper, we propose to verify Web services composition using an event driven approach. We assume Web services that are coordinated by a composition process expressed in WSBPEL and we use Event Calculus to specify the properties and requirements to be monitored.*

8. Huang, H., Tsai, W.-T., Paul, R.. *Automated model checking and testing for composite Web services.* pages 300–307, May 2005.

   Abstract: *Web services form a new distributed computing paradigm. Collaborative verification and validation are important when Web services from different vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web services by model checking the process model of OWL-S (Web ontology language for Web services) and to validate them by the test cases automatically generated in the model checking process. We extend the BLAST, a model checker that handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

9. Rocco De Nicola, Gianluigi Ferrari, Ugo Montanari, Rosario Pugliese, Emilio Tuosto. *A Basic Calculus for Modelling Service Level Agreements.* International Conference on Coordination Models and Languages, LNCS, Vol. 3454, pages 33–48, Springer Verlag, April 2005.

   Abstract: *The definition of suitable abstractions and models for identifying, understanding and managing Quality of Service (QoS) constraints is a challenging issue of the Service Oriented Computing paradigm. In this paper we introduce a process calculus where QoS attributes are first class objects. We identify a minimal set of primitives that allow capturing in an abstract way the ability to control and coordinate services in presence of QoS constraints.*

10. Bernd-Holger Schlingloff, Axel Martens, Karsten Schmidt. *Modeling and Model Checking Web Services.* Vol. 126, pages 3–26, Elsevier, March 2005.

    Abstract: *We give an overview on web services and the web service technology stack. We then show how to build Petri net models of web services formulated in the specification language BPEL4WS. We define an abstract correctness criterion for these models and study the automated verification according to this criterion. Finally, we relate correctness of web service models to the model checking problem for alternating temporal logics.*

11. Rocco De Nicola, Daniele Gorla, Rosario Pugliese. *Basic Observables for a Calculus for Global Computing.* ICALP, pages 1226–1238, 2005.

    Abstract: *We introduce a foundational language for modelling applications over global computers whose interconnection structure can be explicitly manipulated. Together with process distribution, mobility, remote operations and asynchronous communication through distributed data spaces, the language provides constructs for explicitly modelling inter-node connections and for dynamically establishing and removing them. For the proposed language, we define natural notions of extensional observations and study their closure* *under operational reductions and/or language contexts to obtain barbed congruence and may testing equivalence. For such equivalences, we provide alternative characterizations in terms of a labelled bisimulation and a trace equivalence that can be used for actual proofs.*

12. Dan Hirsch, Emilio Tuosto. *SHReQ: Coordinating Application Level QoS.* SEFM, pages 425–434, 2005.

    Abstract: *We present SHReQ, a formal framework for specifying systems that handle abstract high-level QoS requirements which are becoming more and more important for service oriented computing. SHReQ combines Synchronised Hyperedge Replacement (SHR) with constraint-semirings. SHR is a (hyper)graph rewriting mechanism for modelling evolution of systems. The novelty of the approach relies on the synchronisation mechanism which is based on constraint-semirings, algebraic structures that provide both the mathematics for multi-criteria QoS and the synchronisation policies underlying the SHR mechanism.*

13. Khaled Mahbub, George Spanoudakis. *Run-time Monitoring of Requirements for Systems Composed of Web-Services: Initial Implementation and Evaluation Experience.* ICWS, pages 257–265, 2005.

    Abstract: *This paper describes a framework supporting the runtime monitoring of requirements for systems implemented as compositions of Web-services specified in BPEL. The requirements that can be monitored are specified in event calculus. The paper presents an overview of the framework and describes the architecture and implementation of a tool that we have developed to operationalise it. It also presents the results of a preliminary experimental evaluation of the framework.*

14. Wei-Tek Tsai, Yinong Chen, Raymond A. Paul. *Specification-Based Verification and Validation of Web Services and Service-Oriented Operating Systems.* WORDS, pages 139–147, 2005.

    Abstract: *Service-Oriented Architecture (SOA) and Web Services (WS) have received significant attention recently. Even though WS are based on open standards and support software interoperability, but the trustworthy issues of WS has actually limited the growth of WS applications as organizations do not trust those WS developed by other vendors and at the same time they do not have access to the source code. This paper addressed this issue by proposing several solutions including specification-based verification and validation, collaborative testing, and group testing. The key concept is that it is possible to provide a comprehensive evaluation of WS even if their source code is not available.*

15. Foster, H., Uchitel, S., Magee, J., Kramer, J.. *Compatibility verification for Web service choreography.* pages 738–741, July 2004.

    Abstract: *In this paper we discuss a model-based approach to verifying process interactions for coordinated Web service compositions. The approach uses finite state machine representations of Web service orchestrations and assigns semantics to the distributed process interactions. The move towards implementing Web service compositions by multiple interested parties as a form of distributed system architecture motivates the need for supporting compatibility verification of activities and transactions in all the processes. The described approach is supported by a suite of cooperating tools for specification, formal modeling and providing verification results from orchestrated Web service interactions.*

16. Roozbeh Farahbod, Uwe Glässer, Mona Vajihollahi. *Specification and Validation of the Business Process Execution Language for Web Services.* Abstract State Machines, pages 78–94, 2004.

    Abstract: *We formally define an abstract executable semantics for the Business Process Execution Language for Web Services in terms of a distributed ASM. The goal of this work is to support the design and standardization of the language. There is a need for formalism. It will allow us to not only reason about the current specification and related issues, but also uncover issues that would otherwise go unnoticed. Empirical deduction is not sufficient. - Issue 42, OASIS WSBPEL TC. The language definition assumes an infrastructure for running Web services on some asynchronous communication architecture. A business process is built on top of a collection of Web services performing continuous interactions with the outside world by sending and receiving messages over a communication network. The underlying execution model*

is characterized by its concurrent and reactive behavior making it particularly difficult to predict dynamic system properties with a sufficient degree of detail and precision under all circumstances.

17. Gwen Salaün, Lucas Bordeaux, Marco Schaerf. *Describing and Reasoning on Web Services using Process Algebra.* ICWS, 2004.

    Abstract: *We argue that essential facets of web services, and especially those useful to understand their interaction, can be described using process-algebraic notations. Web service description and execution languages such as BPEL are essentially process description languages; they are based on primitives for behaviour description and message exchange which can also be found in more abstract process algebras. One legitimate question is therefore whether the formal approach and the sophisticated tools introduced for process algebra can be used to improve the effectiveness and the reliability of web service development. Our investigations suggest a positive answer, and we claim that process algebras provide a very complete and satisfactory assistance to the whole process of web service development. We show on a case study that readily available tools based on process algebra are effective at verifying that web services conform their requirements and respect properties. We advocate their use both at the design stage and for reverse engineering issues. More prospectively, we discuss how they can be helpful to tackle choreography issues.*

18. Andreas Wombacher, Peter Fankhauser, Erich J. Neuhold. *Transforming BPEL into Annotated Deterministic Finite State Automata for Service Discovery.* ICWS, pages 316–323, 2004.

    Abstract: *Web services advocate loosely coupled systems, although current loosely coupled applications are limited to stateless services. The reason for this limitation is the lack of a method supporting matchmaking of state dependent services exemplarily specified in BPEL. In particular, the sender's requirement that the receiver must support all possible messages sent at a certain state are not captured by models currently used for service discovery. Annotated deterministic finite state automata provide this expressiveness. In this paper the transformation of a local process specification given in BPEL to annotated deterministic finite state automata is presented.*

19. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Mathias Weske. *Business Process Management: A Survey .* International Conference on Business Process Management (BPM), Lecture Notes in Computer Science, Vol. 2678, Springer Verlag, 2003.

    Abstract: *Business Process Management (BPM) includes methods, techniques, and tools to support the design, enactment, management, and analysis of operational business processes. It can be considered as an extension of classical Workflow Management (WFM) systems and approaches. Although the practical relevance of BPM is undisputed, a clear definition of BPM and related acronyms such as BAM, BPA, and STP are missing. Moreover, a clear scientific foundation is missing. In this paper, we try to demystify the acronyms in this domain, describe the state-of-the-art technology, and argue that BPM could benefit from formal methods/languages (cf. Petri nets, process algebras, etc.).*

20. Massimo Mecella, Francesco Parisi Presicce an Barbara Pernici. *Modeling E -service Orchestration through Petri Nets .* Technologies for E-Services, Lecture Notes in Computer Science, Vol. 2444, pages 109–134, Springer Verlag, July 2002.

    Abstract: *B2B interaction requires new forms of coordination between participating organizations. Indeed, the main requirement is that the autonomy of each participating partner is preserved during the interaction, guaranteeing at the same time that the overall goals of the common process are reached. Mechanisms for regulating distributed workflow evolution when the workflow is composed of the invocation of E-services of different organizations are needed. The E-service orchestration model proposed in this paper provides a mechanism for supporting control of process evolution in terms both of control and data flows, and for distributing and assigning process responsibilities.*

21. Srini Narayanan, Sheila A. McIlraith. *Simulation, verification and automated composition of web services.* WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 77–88, ACM, 2002.

    Abstract: *Web services – Web-accessible programs and devices - are a key application area for the Semantic Web. With the proliferation of Web services and the evolution towards the Semantic Web comes*

the opportunity to automate various Web services tasks. Our objective is to enable markup and automated reasoning technology to describe, simulate, compose, test, and verify compositions of Web services. We take as our starting point the DAML-S DAML+OIL ontology for describing the capabilities of Web services. We define the semantics for a relevant subset of DAML-S in terms of a first-order logical language. With the semantics in hand, we encode our service descriptions in a Petri Net formalism and provide decision procedures for Web service simulation, verification and composition. We also provide an analysis of the complexity of these tasks under different restrictions to the DAML-S composite services we can describe. Finally, we present an implementation of our analysis techniques. This implementation takes as input a DAML-S description of a Web service, automatically generates a Petri Net and performs the desired analysis. Such a tool has broad applicability both as a back end to existing manual Web service composition tools, and as a stand-alone tool for Web service developers.

22. Wil M. P. van der Aalst. *Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques.* Business Process Management, pages 161–183, 2000.
    Abstract: *Workflow management systems facilitate the everyday operation of business processes by taking care of the logistic control of work. In contrast to traditional information systems, they attempt to support frequent changes of the workflows at hand. Therefore, the need for analysis methods to verify the correctness of workflows is becoming more prominent. In this chapter we present a method based on Petri nets. This analysis method exploits the structure of the Petri net to find potential errors in the design of the workflow. Moreover, the analysis method allows for the compositional verification of workflows.*

23. Laurent Fribourg, Marcos Veloso Peixoto. *Concurrent Constraint Automata.* International Logic Programming Symposium, 1993.
    Abstract: *We address the problem of the specification and the proof of properties of concurrent systems which manipulate an unbounded number of data. We propose an approach based on an extended notion of automata, called "Concurrent Constraint Automata (CCA)". A CCA is an automaton with constraints and synchronous communication. By automata with constraints, we mean a state machine whose actions and states contain parameters that take their value in the set of natural numbers. (...)*

Books and Monographs:

1. Mariya Koshkina. *Verification of Business Processes for Web Services.* Ms. Thesis, York University - Department of Computer Science, October 2003.
   Abstract: *The Business Process Execution Language for Web Services (BPEL4WS or simply BPEL) is a recently developed language, which is used to specify interactions between web services. Among its features it allows specification of concurrent behavior. Erroneous specification can lead to such problems as deadlock. In our research we focus on the concurrency mechanism in BPEL. Our main goal is to analyze processes in order to detect possible deadlocks. To achieve this we introduce a process algebra called the BPE-calculus. It is a small language which captures all the BPEL features relevant to the analysis. This process algebra is modelled using a labeled transition system. An existing verification tool called the Concurrency Workbench is customized to use our BPE-calculus. This tool allows us to verify many properties of BPE-calculus processes specified in a logic called the mu-calculus, including deadlock freedom.*

Articles in Books and Other Collections:

1. Natasha Sharygina, Daniel Kröning. *Model Checking with Abstraction for Web Services.* Test and Analysis of Web Services, pages 121–145, 2007.
   Abstract: *Web services are highly distributed programs and, thus, are prone to concurrency-related errors. Model checking is a powerful technique to identify flaws in concurrent systems. However, the existing model checkers have only very limited support for the programming languages and communication mechanisms used by typical implementations of web services. This chapter presents a formalization of communication semantics geared for web services, and an automated way to extract formal models from programs implementing web services for automatic formal analysis. The formal models are analyzed by means of a symbolic*

model checker that implements automatic abstraction refinement. Our implementation takes one or more PHP5 programs as input, and is able to verify joint properties of these programs running concurrently.

Publications in Refereed Workshops:

1. S. Nakajima. *Model-Checking Verification for Reliable Web Service.* OOPSLA Workshop on Object-Oriented Web Services, 2002.

   Abstract: *Model-checking is a promising technique for the verification and validation of software systems. Web service, an emerging technology in the Internet, is an autonomous server that may offer an individual service. It sometimes requires to combine more than one to meet our requirements. WSFL(Web Services Flow Language) is proposed as a language to provide means to describe Web service aggregation. We are interested in how much the software model-checking technique can be used as a basis for raising reliability of Web service, Web service flow descriptions in particular. Our experience shows that faulty flow descriptions can be identified with the proposed method. The method is also very helpful in studying an alternative semantics of the WSFL in regard to the handling of dataflows.*

11 Publications in **Grids**

Articles in Refereed Journals:

1. Andrew W. Cooke, Alasdair J. G. Gray, Werner Nutt, James Magowan, Manfred Oevers, Paul Taylor, Roney Cordenonsi, Rob Byrom, Linda Cornwall, Abdeslem Djaoui, Laurence Field, Steve Fisher, Steve Hicks, Jason Leake, Robin Middleton, Antony J. Wilson, Xiaomei Zhu, Norbert Podhorszki, Brian A. Coghlan, Stuart Kenny, David O'Callaghan, John Ryan. *The Relational Grid Monitoring Architecture: Mediating Information about the Grid.* J. Grid Comput., Vol. 2, Num. 4, pages 323–339, 2004.

   Abstract: *We have developed and implemented the Relational Grid Monitoring Architecture (R-GMA) as part of the DataGrid project, to provide a flexible information and monitoring service for use by other middleware components and applications. R-GMA presents users with a virtual database and mediates queries posed at this database: users pose queries against a global schema and R-GMA takes responsibility for locating relevant sources and returning an answer. R-GMAs architecture and mechanisms are general and can be used wherever there is a need for publishing and querying information in a distributed environment. We discuss the requirements, design and implementation of R-GMA as deployed on the DataGrid testbed. We also describe some of the ways in which R-GMA is being used.*

Articles in Refereed Conferences:

1. Jonatha Anselmi, Danilo Ardagna, Paolo Cremonesi. *A QoS-based selection approach of autonomic grid services.* SOCP '07: Proceedings of the 2007 workshop on Service-oriented computing performance: aspects, issues, and approaches, pages 1–8, ACM, 2007.

   Abstract: *The Web service composition (WSC) is the process of building an instance of an abstract workflow by combining appropriate Web services that satisfies given QoS requirements. In general, QoS requirements consists of a number of constraints. The selection process requires global optimization and can be formalized as a mixed integer linear programming problem which cannot be solved in polynomial time. However, since the number of submitted workflows is large and the QoS is highly dynamic, the fast selection of composite Web Services is particularly important. In this paper, we present a QoS broker-based framework for Web services execution in autonomic grid environments. The main goal of the framework is to support the broker in selecting Web services based on the required QoS. To achieve this goal, we propose a novel approach: since successive composed Web services requests can have the same task to Web service assignment, we address the Multiple Instance WSC (MI-WSC) problem optimizing simultaneously the set of requests which will be submitted to the system in the successive time interval instead of independently computing a solution for each incoming request. Experimental results show that the proposed algorithm has better performance with respect to existing techniques. Moreover, the qualities of the selected composite Web services are not significantly different from the optimal ones.*

2. Qing Zhu, Shan Wang, Guorong Li, Guangqiang Liu, Xiaoyong Du. *QoS-Based Services Selecting and Optimizing Algorithms on Grid.* APWeb/WAIM Workshops, pages 156–167, 2007.

   Abstract: *QoS-Based Services Selecting and Optimizing Composition between the peers play an increasingly important role to ensure interoperability on Grid environment. However, the prohibitive cost of selecting, matching, mapping and composing algorithm has now become a key bottleneck hindering the deployment of a wide variety of Grid services. In this paper, we present QoS-Based Services Selecting and Optimizing Composition on Grid. First, it checks requesters' semantic in order to form candidate service graph. Second, it designs service selecting and mapping algorithms for optimizing the model. Third, it creates an executed plan of optimum composition on Grid. We conducted experiments to simulate and evaluate our approach.*

3. Gang Huang, Meng Wang, Liya Ma, Ling Lan, Tiancheng Liu, Hong Mei. *Towards Architecture Model based Deployment for Dynamic Grid Services.* IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04), 2004.

Abstract: *The deployment of grid services should make the services, including those to be deployed and those already deployed, operate with desired functionalities and qualities. The critical challenge in the deployment is that many technical and nontechnical factors have to be taken into account, such as performance, reliability, utilization, operating cost, incomes, and so on. Since the factors change continuously, some deployed services may have to be redeployed for guaranteeing their functionalities and qualities. This position paper presents an approach to the deployment and redeployment of grid services based on software architecture models. In this approach, all services in a grid consist in a software architecture, which represents the services, their relationships and other factors in a global, understandable and easy-to-use way. To demonstrate the approach, a visual tool for deploying services onto a set of popular grid infrastructures, including J2EE application servers and BPEL engines, with the help of software architectures is developed.*

Technical Reports and Manuals:

1. I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich . *The Open Grid Services Architecture, Version 1.0.* ANL, IBM, Fujitsu, NeSC, CCLRC-RAL, UVa, Hitachi, Intel, HP, 2005.

   Abstract: *Successful realization of the Open Grid Services Architecture (OGSA) vision of a broadly applicable and adopted framework for distributed system integration, virtualization, and management requires the definition of a core set of interfaces, behaviors, resource models, and bindings. This document, produced by the OGSA working group within the Global Grid Forum (GGF), provides a first version of this OGSA definition. The document focuses on requirements and the scope of important capabilities required to support Grid systems and applications in both e-science and e-business. The capabilities described are Execution Management, Data, Resource Management, Security, Self-Management, and Information. The description of the capabilities is at a high-level and includes, to some extent, the interrelationships between the capabilities.*

12 Publications in Monitoring

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

2. Andrew W. Cooke, Alasdair J. G. Gray, Werner Nutt, James Magowan, Manfred Oevers, Paul Taylor, Roney Cordenonsi, Rob Byrom, Linda Cornwall, Abdeslem Djaoui, Laurence Field, Steve Fisher, Steve Hicks, Jason Leake, Robin Middleton, Antony J. Wilson, Xiaomei Zhu, Norbert Podhorszki, Brian A. Coghlan, Stuart Kenny, David O'Callaghan, John Ryan. *The Relational Grid Monitoring Architecture: Mediating Information about the Grid.* J. Grid Comput., Vol. 2, Num. 4, pages 323–339, 2004.

   Abstract: *We have developed and implemented the Relational Grid Monitoring Architecture (R-GMA) as part of the DataGrid project, to provide a flexible information and monitoring service for use by other middleware components and applications. R-GMA presents users with a virtual database and mediates queries posed at this database: users pose queries against a global schema and R-GMA takes responsibility for locating relevant sources and returning an answer. R-GMAs architecture and mechanisms are general and can be used wherever there is a need for publishing and querying information in a distributed environment. We discuss the requirements, design and implementation of R-GMA as deployed on the DataGrid testbed. We also describe some of the ways in which R-GMA is being used.*

Articles in Refereed Conferences:

1. Fabio Barbon, Paolo Traverso, Marco Pistore, Michele Trainotti. *Run-Time Monitoring of Instances and Classes of Web Service Compositions.* ICWS, pages 63–71, 2006.

   Abstract: *The run-time monitoring of web service compositions has been widely acknowledged as a significant and challenging problem. In this paper, we propose a novel solution to the problem of monitoring web services implemented in BPEL. We devise an architecture that clearly separates the business logic of a web service from its monitoring functionality. The architecture supports both "instance monitors" that deal with the execution of a single instance of BPEL process, as well as "class monitors" that report aggregated information about all the instances of a BPEL process. We also define a language for the specification of instance and class monitors. The language allows for specifying boolean, statistic, and time-related properties. Finally, we devise a technique for the automatic translation of all these kinds of monitors to Java programs.*

2. Luciano Baresi, Sam Guinea. *Towards Dynamic Monitoring of WS-BPEL Processes.* ICSOC, pages 269–282, 2005.

Abstract: *The intrinsic flexibility and dynamism of service-centric applications preclude their pre-release validation and demand for suitable probes to monitor their behavior at run-time. Probes must be suitably activated and deactivated according to the context in which the application is executed, but also according to the confidence we get on its quality. The paper supports the idea that significant data may come from very different sources and probes must be able to accommodate all of them. The paper presents: (1) an approach to specify monitoring directives, called monitoring rules, and weave them dynamically into the process they belong to; (2) a proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time; (3) a user-oriented language to integrate data acquisition and analysis into monitoring rules.*

3. Khaled Mahbub, George Spanoudakis. *Run-time Monitoring of Requirements for Systems Composed of Web-Services: Initial Implementation and Evaluation Experience.* ICWS, pages 257–265, 2005.

   Abstract: *This paper describes a framework supporting the runtime monitoring of requirements for systems implemented as compositions of Web-services specified in BPEL. The requirements that can be monitored are specified in event calculus. The paper presents an overview of the framework and describes the architecture and implementation of a tool that we have developed to operationalise it. It also presents the results of a preliminary experimental evaluation of the framework.*

13 Publications in Negotiation

Articles in Refereed Journals:

1. Nicholas R. Jennings, Peyman Faratin, A. R. Lomuscio, Simon Parsons, Michael Wooldridge, Carles Sierra. *Automated Negotiation: Prospects Methods and Challenges.* Group Decision and Negotiation, To Appear, 2001.

Articles in Refereed Conferences:

1. Elisabetta Di Nitto, Massimiliano Di Penta, Alessio Gambi, Gianluca Ripa, Maria Luisa Villani. *Negotiation of Service Level Agreements: An Architecture and a Search-Based Approach.* ICSOC, pages 295–306, 2007.

   Abstract: *Software systems built by composing existing services are more and more capturing the interest of researchers and practitioners. The envisaged long term scenario is that services, offered by some competing providers, are chosen by some consumers and used for their own purpose, possibly, in conjunction with other services. In the case the consumer is not anymore satisfied by the performance of some service, he can try to replace it with some other service. This implies the creation of a global market of services and poses new requirements concerning validation of exploited services, security of transactions engaged with services, trustworthiness, creation and negotiation of Service Level Agreements with these services. In this paper we focus on the last aspect and present our approach for negotiation of Service Level Agreements. Our architecture supports the actuation of various negotiation processes and offers a search-based algorithm to assist the negotiating parts in the achievement of an agreement.*

2. Mohan Baruwal Chhetri, Jian Lin, SukKeong Goh, Jian Ying Zhang, Ryszard Kowalczyk, Jun Yan. *A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation ofWeb Service Composition.* ASWEC, pages 90–99, 2006.

   Abstract: *Recent progress in the field of Web services has made it possible to integrate inter-organizational and heterogeneous services on the Web at runtime. If a user request cannot be satisfied by a single Web service, it is (or should be) possible to combine existing services in order to fulfill the request. However, there are several challenging issues that need to be addressed before this can be realized in the true sense. One of them is the ability to ensure end-to-end QoS of a Web service composition. There is a need for a SLA negotiation system which can ensure the autonomous QoS negotiation of Web service compositions irrespective of the application domain. In this paper we propose agent-based coordinated-negotiation architecture to ensure collective functionality, end-to-end QoS and the stateful coordination of complex services. We describe a prototype implementation to demonstrate how this architecture can be used in different application domains. We have also demonstrated how the negotiation system on the service provider's side can be implemented both as an agent based negotiation system and as a Web service based negotiation system.*

3. Andreas Wombacher, Peter Fankhauser, Erich J. Neuhold. *Transforming BPEL into Annotated Deterministic Finite State Automata for Service Discovery.* ICWS, pages 316–323, 2004.

   Abstract: *Web services advocate loosely coupled systems, although current loosely coupled applications are limited to stateless services. The reason for this limitation is the lack of a method supporting matchmaking of state dependent services exemplarily specified in BPEL. In particular, the sender's requirement that the receiver must support all possible messages sent at a certain state are not captured by models currently used for service discovery. Annotated deterministic finite state automata provide this expressiveness. In this paper the transformation of a local process specification given in BPEL to annotated deterministic finite state automata is presented.*

4. Octavio Martín-Díaz, Antonio Ruiz Cortés, Amador Durán, David Benavides, Miguel Toro. *Automating the Procurement of Web Services.* ICSOC, pages 91–103, 2003.

   Abstract: *As government agencies and business become more dependent on web services, software solutions* *to automate their procurement gain importance. Current approaches for automating the procurement of web services suffer from an important drawback: neither uncertainty measures nor non-linear, and complex relations among parameters can be used by providers to specify quality-of-service in offers. In this paper, we look deeply into the roots of this drawback and present a proposal which overcomes it. The key point to achieve this improvement has been using the constraint programming as a formal basis, since it endows the model with a very powerful expressiveness. A XML-based implementation is presented along with some experimental results and comparisons with other approaches.*

14 Publications in **Quality of Service**

Articles in Refereed Journals:

1. D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, P. Plebani. *PAWS: A Framework for Executing Adaptive Web-Service Processes.* IEEE Software, To Appear, 2007.

   Abstract: *The processes with adaptive Web services framework couples design-time and runtime mechanisms to flexibly and adoptively execute managed Web-services-based business processes.*

2. Stefano Bistarelli, Ugo Montanari, Francesca Rossi, Francesco Santini. *Unicast and Multicast Qos Routing with Soft Constraint Logic Programming.* CoRR, Vol. abs/0704.1783, 2007.

   Abstract: *We present a formal model to represent and solve the unicast/multicast routing problem in networks with Quality of Service (QoS) requirements. To attain this, first we translate the network adapting it to a weighted graph (unicast) or and-or graph (multicast), where the weight on a connector corresponds to the multidimensional cost of sending a packet on the related network link: each component of the weights vector represents a different QoS metric value (e.g. bandwidth, cost, delay, packet loss). The second step consists in writing this graph as a program in Soft Constraint Logic Programming (SCLP): the engine of this framework is then able to find the best paths/trees by optimizing their costs and solving the constraints imposed on them (e.g. delay ¡ 40msec), thus finding a solution to QoS routing problems. Moreover, c-semiring structures are a convenient tool to model QoS metrics. At last, we provide an implementation of the framework over scale-free networks and we suggest how the performance can be improved.*

3. Giorgia Lodi, Fabio Panzieri, Davide Rossi, Elisa Turrini. *SLA-Driven Clustering of QoS-Aware Application Servers.* IEEE Trans. Software Eng., Vol. 33, Num. 3, pages 186–197, 2007.

   Abstract: *In this paper, we discuss the design, implementation, and experimental evaluation of a middleware architecture for enabling service level agreement (SLA)-driven clustering of QoS-aware application servers. Our middleware architecture supports application server technologies with dynamic resource management: application servers can dynamically change the amount of clustered resources assigned to hosted applications on-demand so as to meet application-level quality of service (QoS) requirements. These requirements can include timeliness, availability, and high throughput and are specified in SLAs. A prototype of our architecture has been implemented using the open-source J2EE application server JBoss. The evaluation of this prototype shows that our approach makes possible JBoss' resource usage optimization and allows JBoss to effectively meet the QoS requirements of the applications it hosts, i.e., to honor the SLAs of those applications.*

4. Tao Yu, Kwei-Jay Lin. *Service selection algorithms for Web services with end-to-end QoS constraints.* Inf. Syst. E-Business Management, Vol. 3, Num. 2, pages 103–126, 2005.

   Abstract: *Web services are new forms of Internet software that can be universally deployed and invoked using standard protocols. Services from different providers can be integrated into a composite service regardless of their locations, platforms, and/or execution speeds to implement complex business processes and transactions. In this paper, we study the end-to-end QoS issues of composite services by utilizing a QoS broker that is responsible for selecting and coordinating the individual service component. We design the service selection algorithms used by QoS brokers to construct the optimal composite service. The objective of the algorithms is to maximize the user-defined utility function value while meeting the end-to-end delay constraint. We propose two solution approaches to the service selection problem: the combinatorial approach, by modeling the problem as the Multiple Choice Knapsack Problem (MCKP), and the graph approach, by modeling the problem as the constrained shortest path problem in the graph theory. We study efficient solutions for each approach.*

5. Liangzhao Zeng, Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.. *QoS-aware middleware for Web services composition.* Software Engineering, IEEE Transactions on, Vol. 30, Num. 5, pages 311–327, May 2004.

   Abstract: *The paradigmatic shift from a Web of manual interactions to a Web of programmatic interactions* *driven by Web services is creating unprecedented opportunities for the formation of online business-to-business (B2B) collaborations. In particular, the creation of value-added services by composition of existing ones is gaining a significant momentum. Since many available Web services provide overlapping or identical functionality, albeit with different quality of service (QoS), a choice needs to be made to determine which services are to participate in a given composite service. This paper presents a middleware platform which addresses the issue of selecting Web services for the purpose of their composition in a way that maximizes user satisfaction expressed as utility functions over QoS attributes, while satisfying the constraints set by the user and by the structure of the composite service. Two selection approaches are described and compared: one based on local (task-level) selection of services and the other based on global allocation of tasks to services using integer programming.*

6. Xavier Franch, Juan Pablo Carvallo. *Using Quality Models in Software Package Selection.* IEEE Software, Vol. 20, Num. 1, pages 34–41, 2003.

   Abstract: *The growing importance of commercial off-the-shelf software packages requires adapting some software engineering practices, such as requirements elicitation and testing, to this emergent framework. Also, some specific new activities arise, among which selection of software packages plays a prominent role. All the methodologies that have been proposed recently for choosing software packages compare user requirements with the packages' capabilities. There are different types of requirements, such as managerial, political, and, of course, quality requirements. Quality requirements are often difficult to check. This is partly due to their nature, but there is another reason that can be mitigated, namely the lack of structured and widespread descriptions of package domains (that is, categories of software packages such as ERP systems, graphical or data structure libraries, and so on). This absence hampers the accurate description of software packages and the precise statement of quality requirements, and consequently overall package selection and confidence in the result of the process. Our methodology for building structured quality models helps solve this drawback.*

7. Shuping Ran. *A model for web services discovery with QoS.* SIGecom Exch., Vol. 4, Num. 1, pages 1–10, ACM, 2003.

   Abstract: *Web services technology has generated a lot interest, but its adoption rate has been slow. This paper discusses issues related to this slow take up and argues that quality of services is one of the contributing factors. The paper proposes a new Web services discovery model in which the functional and non-functional requirements (i.e. quality of services) are taken into account for the service discovery. The proposed model should give Web services consumers some confidence about the quality of service of the discovered Web services.*

8. Menasce, D.A.. *QoS Issues in Web Services.* Internet Computing, IEEE, Vol. 6, Num. 6, pages 72–75, November/December 2002.

   Abstract: *Quality of service (QoS) is a combination of several qualities or properties of a service, such as: availability is the percentage of time that a service is operating; security properties include the existence and type of authentication mechanisms the service offers, confidentiality and data integrity of messages exchanged, nonrepudiation of requests or messages, and resilience to denial-of-service attacks; response time is the time a service takes to respond to various types of requests; Response time is a function of load intensity, which can be measured in terms of arrival rates (such as requests per second) or number of concurrent requests. QoS takes into account not only the average response time, but also the percentile of the response time; and throughput is the rate at which a service can process requests. QoS measures can include the maximum throughput or a function that describes how throughput varies with load intensity. The QoS measure is observed by Web services users. These users are not human beings but programs that send requests for services to Web service providers. QoS issues in Web services have to be evaluated from the perspective of the providers of Web services and from the perspective of the users of these services.*

Articles in Refereed Conferences:

1. Kyriakos Kritikos, Dimitris Plexousakis. *Evaluation of QoS-Based Web Service Matchmaking Algorithms.* International Conference on Services Computing, 2008.

Abstract: *Web Service (WS) discovery is a prerequisite for achieving WS composition and orchestration. Although a lot of esearch has been conducted on the functional discovery of WSs, the proposed techniques fall short when faced with the foreseen increase in the number of (potentially functionally- equivalent) WSs. The above situation can be resolved with the addition of non-functional (Quality of Service (QoS)) discovery mechanisms to WS discovery engines. QoS-based WS matchmaking algorithms have been devised for this reason. However, they are either slow - as they are based on ontology reasoners - or produce inaccurate results. Inaccuracy is caused both by the syntactic matching of QoS concepts and by wrong matchmaking metrics. In this paper, we present two Constraint Programming (CP) QoS-based WS discovery algorithms for unary constrained WS specifications that produce accurate results with good performance. We also evaluate these algorithms on matchmaking time, precision and recall in different settings in order to demonstrate their efficiency and accuracy.*

2. Kyriakos Kritikos, Dimitris Plexousakis. *OWL-Q for Semantic QoS-based Web Service Description and Discovery.* First International Joint Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, November 2007.

   Abstract: *Semantic Web Services are emerging for their promise to produce a more accurate and precise Web Service discovery process. However, most of research approaches focus only on the functional part of semantic Web Service description. The above fact along with the proliferation of Web Services is highly probable to lead to a situation where Web Service registries will return many functionally-equivalent Web Service advertisements for each user request. This problem can be solved with the semantic description of QoS for Web Services. QoS is a set of non-functional properties encompassing performance and network-related characteristics of resources. So it can be used for distinguishing between functionally-equivalent Web Services. Current research approaches for QoS-based Web Service description are either syntactic or poor or non-extensible. To solve this problem, we have developed a rich and extensible ontological specification called OWL-Q for semantic QoS-based Web Service description. We analyze all OWL-Q parts and reason that rules should be added in order to support property inferencing and constraint enforcement. Finally, we line out our under-development semantic framework for QoS-based Web Service description and discovery.*

3. Cinzia Cappiello, Marco Comuzzi, Pierluigi Pleban. *On Automated Generation of Web Service Level Agreements .* Advanced Information Systems Engineering, Vol. 4495, Springer Verlag, June 2007.

   Abstract: *Before a service invocation takes place, an agreement between the service provider and the service user might be required. Such an agreement is the result of a negotiation process between the two parties and defines how the service invocation has to occur. Considering the Service Oriented Computing paradigm, the relationship among providers and users is extremely loose. Traditional agreements are likely to concern long term relationships and to be manually performed. In this paper, we propose a model to generate service level agreement on-the-fly. Just before the invocation commences, the quality of the service is negotiated in order to generate a service level agreement tied to that specific invocation. Such an approach relies on a quality model that supports both users requirements and providers capabilities definition.*

4. Jonatha Anselmi, Danilo Ardagna, Paolo Cremonesi. *A QoS-based selection approach of autonomic grid services.* SOCP '07: Proceedings of the 2007 workshop on Service-oriented computing performance: aspects, issues, and approaches, pages 1–8, ACM, 2007.

   Abstract: *The Web service composition (WSC) is the process of building an instance of an abstract workflow by combining appropriate Web services that satisfies given QoS requirements. In general, QoS requirements consists of a number of constraints. The selection process requires global optimization and can be formalized as a mixed integer linear programming problem which cannot be solved in polynomial time. However, since the number of submitted workflows is large and the QoS is highly dynamic, the fast selection of composite Web Services is particularly important. In this paper, we present a QoS broker-based framework for Web services execution in autonomic grid environments. The main goal of the framework is to support the broker in selecting Web services based on the required QoS. To achieve this goal, we propose a novel approach: since successive composed Web services requests can have the same task to Web service assignment, we address the Multiple Instance WSC (MI-WSC) problem optimizing simultaneously the set of requests which will be submitted to the system in the successive time interval instead of independently*

computing a solution for each incoming request. Experimental results show that the proposed algorithm has better performance with respect to existing techniques. Moreover, the qualities of the selected composite Web services are not significantly different from the optimal ones.

5. Ester Giallonardo, Eugenio Zimeo. *More Semantics in QoS Matching.* SOCA, pages 163–171, 2007.

Abstract: *The evolution of the Web towards a global computing environment is promoting new research efforts aimed at the formal characterization of Web Services QoS. Reasoning on QoS is a key to improve matching process during the discovery of desired services and a step towards the transformation of applications in collections of loosely coupled services virtually connected by semantic similarities. The paper presents the on QoS ontology, an openly available OWL ontology for QoS, and evaluates it in a QoS-aware matching environment. The ontology can be used to express functions of QoS metrics useful to improve the recall tied to the matching of a template request with target Web Services. To this end, the ontology introduces the concept of derivation in the matching process. This gives the possibility of matching a QoS template with published Web Services by deriving different QoS parameters when a one-to-one matching fails. The proposed matching algorithm utilizes a reasoner that exploits the ontology to avoid apparent mismatches. An experimental evaluation shows that exploiting QoS knowledge significantly improves matching recall without deteriorating precision.*

6. Kyriakos Kritikos, Dimitris Plexousakis. *Requirements for QoS-based Web Service Description and Discovery.* COMPSAC (2), pages 467–472, 2007.

Abstract: *The goal of Service Oriented Architectures is to enable the creation of business applications through the automatic discovery and composition of independently developed and deployed (Web) services. Automatic discovery of Web Services (WSs) can be achieved by incorporating semantics into a richer WS description model (WSDM) and the use of Semantic Web (SW) technologies in the WS matchmaking and selection models. A sufficiently rich WSDM should encompass not only functional but also non-functional aspects like Quality of Service (QoS). QoS is a set of performance attributes that has a substantial impact on WS requesters' expectations. Thus, it can be used as a discriminating factor of functionally-equivalent WSs. The focus of this paper is twofold: to analyze the requirements of a semantically rich QoS-based WSDM and to provide SW and constrainedbased mechanisms for enriching syntactic QoS-based WS Discovery (WSDi) algorithms. In addition, a roadmap of extending WS standard techniques for realizing semantic, functional and QoS-based WSDi is presented.*

7. Kyriakos Kritikos, Dimitris Plexousakis. *Semantic QoS-based Web Service Discovery Algorithms.* Fifth European Conference on Web Services, pages 181–190, IEEE Computer Society, 2007.

Abstract: *The success of the Web Service (WS) paradigm has led to a proliferation of available WSs, which are advertised in WS registries. While sophisticated semantic WS discovery algorithms are operating on these registries to return matchmaking results with high precision and recall, many functionally-equivalent WSs are returned. The solution to the above problem comes in terms of semantic QoS-based description and discovery of WSs. We have already presented a rich and extensible ontology language for QoS-based WS description called OWL-Q. We have also proposed a semantic QoS metric matching algorithm. Based on this algorithm, we have extended a CSP-based approach for QoS-based WS discovery. In this paper, we firstly analyze the evolution of OWL-Q and its extension with SWRL rules, we propose a modification to the metric matching algorithm and we show the way the metric alignment process takes place. Then we propose two novel semantic QoS-based WS Discovery algorithms that return matches even for over-constrained QoS-based WS requests. The first one deals with unary constraints while the second one is more generic. Finally, implementation aspects of our QoS-based WS discovery system are discussed.*

8. Qing Zhu, Shan Wang, Guorong Li, Guangqiang Liu, Xiaoyong Du. *QoS-Based Services Selecting and Optimizing Algorithms on Grid.* APWeb/WAIM Workshops, pages 156–167, 2007.

Abstract: *QoS-Based Services Selecting and Optimizing Composition between the peers play an increasingly important role to ensure interoperability on Grid environment. However, the prohibitive cost of selecting,*

matching, mapping and composing algorithm has now become a key bottleneck hindering the deployment of a wide variety of Grid services. In this paper, we present QoS-Based Services Selecting and Optimizing Composition on Grid. First, it checks requesters' semantic in order to form candidate service graph. Second, it designs service selecting and mapping algorithms for optimizing the model. Third, it creates an executed plan of optimum composition on Grid. We conducted experiments to simulate and evaluate our approach.

9. Dmytro Rud, Andreas Schmietendorf, Reiner Dumke. *Performance Modeling of WS-BPEL-Based Web Service Compositions.* Services Computing Workshops, pages 140–147, September 2006.

   Abstract: *This paper addresses quality of service aspects of Web service orchestrations created using WS-BPEL from the standpoint of a Web service integrator. A mathematical model based on operations research techniques and formal semantics of WS-BPEL is proposed to estimate and forecast the influence of the execution of orchestrated processes on utilization and throughput of individual involved nodes and of the whole system. This model is applied to the optimization of service levels agreement process between the involved parties*

10. Yan-ping Chen, Zeng-zhi Li, Qin-xue Jin, Chuang Wang. *Study on QoS Driven Web Services Composition.* Frontiers of WWW Research and Development - APWeb 2006, Lecture Notes on Computer Science, Vol. 3841, pages 702–707, Springer Verlag, 2006.

    Abstract: *Providing composed Web Services based on the QoS requirements of clients is still an urgent problem to be solved. In this paper, we try to solve this problem. Firstly, we enhanced the current WSDL to describe the QoS of services, and then gave a way to choose the proper pre-exist services based on their QoS.*

11. Kyriakos Kritikos, Dimitris Plexousakis. *Semantic QoS Metric Matching.* ECOWS, pages 265–274, 2006.

    Abstract: *As the Web Service paradigm gains popularity for its promise to transform the way business is conducted, the number of deployed Web Services grows with a fast rate. While sophisticated semantic discovery mechanisms have been invented to overcome the UDDI's syntactic discovery solution in order to provide more recallable and precise results, the amount of functionally equivalent Web Services returned is still large. The solution to this problem is the description of the QoS non-functional aspect of Web Services. QoS encompasses the performance of Web Services and can be used as a discriminator factor for refining Web Service advertisement result lists. However, most scientific efforts presented so far are purely syntactic and are not capturing all aspects of QoS-based Web Service description leading to imprecise syntactic discovery mechanisms. This paper presents a novel, rich and extensible ontology-based approach for describing QoS of Web Services that complements OWL-S. It is shown that, by using this approach and by introducing the concept of semantic QoS metric matching, QoS-based syntactic matchmaking and selection algorithms are transformed to semantic ones leading to better results.*

12. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani. *An approach for QoS-aware service composition based on genetic algorithms.* GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, pages 1069–1075, ACM, 2005.

    Abstract: *Web services are rapidly changing the landscape of software engineering. One of the most interesting challenges introduced by web services is represented by Quality Of Service (QoS)–aware composition and late–binding. This allows to bind, at run–time, a service–oriented system with a set of services that, among those providing the required features, meet some non–functional constraints, and optimize criteria such as the overall cost or response time. In other words, QoS–aware composition can be modeled as an optimization problem.We propose to adopt Genetic Algorithms to this aim. Genetic Algorithms, while being slower than integer programming, represent a more scalable choice, and are more suitable to handle generic QoS attributes. The paper describes our approach and its applicability, advantages and weaknesses, discussing results of some numerical simulations.*

13. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani. *QoS-Aware Replanning of Composite Web Services.* ICWS '05: Proceedings of the IEEE International

Conference on Web Services, pages 121–129, IEEE Computer Society, 2005.

Abstract: *Run-time service discovery and late-binding constitute some ofthe most challenging issues of service-oriented software engineering.For late-binding to be effective in the case of composite services,a QoS-aware composition mechanism is needed. This meansdetermining the set of services that, once composed, not only willperform the required functionality, but also will best contribute toachieve the level of QoS promised in Service Level Agreements (SLAs). However, QoS-aware composition relies on estimated QoS values and workflow execution paths previously obtained using a monitoring mechanism. At run-time, the actual QoS valuesmay deviate from the estimations, or the execution path may not bethe one foreseen. These changes could increase the risk of breakingSLAs and obtaining a poor QoS. Such a risk could be avoided byreplanning the service bindings of the workflow slice still to be executed. This paper proposes an approach to trigger and perform compositeservice replanning during execution. An evaluation has been performedsimulating execution and replanning on a set of composite serviceworkflows.*

14. Casey K. Fung, Patrick C. K. Hung, Guijun Wang, Richard C. Linger, Gwendolyn H. Walton. *A Study of Service Composition with QoS Management.* ICWS, pages 717–724, 2005.

    Abstract: *Quality of service (QoS) management in compositions of services requires careful consideration of QoS characteristics of the services and effective QoS management in their execution. A Web service is a software system that supports interoperable application-to-application interaction over the Internet. Web services are based on a set of XML standards such as simple object access protocol (SOAP). The interactions of SOAP messages between Web services form the theoretical model of SOAP message exchange patterns (MEP). Web Services Business Process Execution Language (WSBPEL) defines an interoperable integration model that facilitates automated process integration in intra- and inter-corporate environments. A service-level agreement (SLA) is a formal contract between a Web services requestor and provider guaranteeing quantifiable issues at defined levels only through mutual concessions. Based on a prior research work on message detail record (MDR), this paper further proposes a SOAP message tracking model for supporting QoS end-to-end management in the context of WSBPEL and SLA. This paper motivates the study of QoS management in a Web service composition framework with the evolution of a distributed toolkit in an industrial setting.*

15. Michael C. Jaeger, Gero Mühl, Sebastian Golze. *QoS-Aware Composition of Web Services: A Look at Selection Algorithms.* ICWS, pages 807–808, 2005.

    Abstract: *When a composition of Web services is designed, available services are put together to form a defined flow of executions. In a discovery process, a trader proposes available Web services as potential candidates. In a succeeding selection, for each task a trader chooses one candidate to form the optimal composition due to selection criteria. This paper discusses how the selection can consider different quality-of-service (QoS) categories to determine the most suitable candidates for the composition. If more than one category is used for optimisation, a multi-dimensional optimisation problem arises. This mentions similarities to similar combinatorial problems. Then, possible solutions are proposed and their performance is evaluated.*

16. Vikas Deora, Jianhua Shao, Gareth Shercliff, Patrick J. Stockreisser, W. A. Gray, N. J. Fiddian. *Incorporating QoS Specifications in Service Discovery.* WISE Workshops, pages 252–263, 2004.

    Abstract: *In this paper, we extend the current approaches to service discovery in a service oriented computing environment, such as Web Services and Grid, by allowing service providers and consumers to express their promises and requirements for quality of service (QoS). More specifically, we allow service providers to advertise their services in an extended DAML-S that supports quality specifications, and we allow service consumers to request services by stating required quality levels. We propose a model here for incorporating QoS specifications and requirements in service discovery, and describe how matchmaking between advertised and requested services based on functional as well as quality requirements is supported in our model.*

17. Michael C. Jaeger, Gregor Rojec-Goldmann, Gero Mühl. *QoS Aggregation for Web Service Composition using Workflow Patterns.* EDOC, pages 149–159, 2004.

    Abstract: *Contributions in the field of Web services have identified that (a) finding matches between semantic descriptions of advertised and requested services and (b) nonfunctional characteristics - the quality* *of service (QoS) - are the most crucial criteria for composition of Web services. A mechanism is introduced that determines the QoS of a Web service composition by aggregating the QoS dimensions of the individual services. This allows to verify whether a set of services selected for composition satisfies the QoS requirements for the whole composition. The aggregation performed builds upon abstract composition patterns, which represent basic structural elements of a composition, like sequence, loop, or parallel execution. This work focusses on workflow management environments. We define composition patterns that are derived from Van der Aalst's et al. comprehensive collection of workflow patterns. The resulting aggregation schema supports the same structural elements as found in workflows. Furthermore, the aggregation of several QoS dimensions is discussed.*

18. James Skene, D. Davide Lamanna, Wolfgang Emmerich. *Precise Service Level Agreements.* ICSE, pages 179–188, 2004.

    Abstract: *SLAng is an XML language for defining service level agreements, the part of a contract between the client and provider of an Internet service that describes the quality attributes that the service is required to possess. We define the semantics of SLAng precisely by modelling the syntax of the language in UML, then relating the language model to a model that describes the structure and behaviour of services. The presence of SLAng elements imposes behavioural constraints on service elements, and the precise definition of these constraints using OCL constitutes the semantic description of the language. We use the semantics to define a notion of SLA compatibility, and an extension to UML that enables the modelling of service situations as a precursor to analysis, implementation and provisioning activities.*

19. Tao Yu, Kwei-Jay Lin. *Service Selection Algorithms for Web Services with End-to-End QoS Constraints.* CEC, pages 129–136, 2004.

    Abstract: *Web services are new forms of Internet software that can be universally deployed and invoked using standard protocol. Services from different providers can be integrated to provide composite services. In this paper, we study the end-to-end QoS issues of composite service by utilizing a QoS broker that is responsible for coordinating the individual service component to meet the quality constraint. We design the service selection algorithms used by QoS brokers to meet end-to-end QoS constraints. The objective of the algorithms is to maximize the user-defined utility while meeting the end-to-end delay constraint. We model the problem as the Multiple Choice Knapsack Problem (MCKP) and provide efficient solutions. The algorithms are tested for their performance.*

20. Vikas Deora, Jianhua Shao, W. A. Gray, N. J. Fiddian. *A Quality of Service Management Framework Based on User Expectations.* ICSOC, pages 104–114, 2003.

    Abstract: *The ability to gauge the quality of a service is critical if we are to achieve the service oriented computing paradigm. Many techniques have been proposed and most of them attempt to calculate the quality of a service by collecting quality ratings from the users of the service, then combining them in one way or another. We argue that collecting quality ratings alone from the users is not sufficient for deriving a reliable or accurate quality measure for a service. This is because different users often have different expectations on the quality of a service and their ratings tend to be closely related to their expectations, i.e. how their expectations are met. In this paper, we propose a quality of service management framework based on user expectations. That is, we collect expectations as well as ratings from the users of a service, then calculate the quality of the service only at the time a request for the service is made and only using the ratings that have similar expectations. We give examples to show that our approach can result in a more accurate and meaningful measure for quality of service.*

21. Octavio Martín-Díaz, Antonio Ruiz Cortés, Amador Durán, David Benavides, Miguel Toro. *Automating the Procurement of Web Services.* ICSOC, pages 91–103, 2003.

    Abstract: *As government agencies and business become more dependent on web services, software solutions to automate their procurement gain importance. Current approaches for automating the procurement of web services suffer from an important drawback: neither uncertainty measures nor non-linear, and complex relations among parameters can be used by providers to specify quality-of-service in offers. In this paper, we look deeply into the roots of this drawback and present a proposal which overcomes it. The key point to achieve this improvement has been using the constraint programming as a formal basis, since it endows* *the model with a very powerful expressiveness. A XML-based implementation is presented along with some experimental results and comparisons with other approaches.*

22. Vladimir Tosic, Bernard Pagurek, Kruti Patel. *WSOL - A Language for the Formal Specification of Classes of Service for Web Services.* ICWS, pages 375–381, 2003.

    Abstract: *We are developing a language, Web Service Offerings Language (WSOL), for the formal specification of various constraints, management statements, and classes of service for Web Services. WSOL is an XML (Extensible Markup Language) notation compatible with WSDL (Web Services Description Language). A service offering in WSOL is a formal description of one class of service of a Web Service. It contains formal representation of various constraints: functional (pre-, post-, and future- conditions), Quality of Service (QoS, a.k.a. non-functional, extra- functional), and access rights. It also contains management statements, such as statements about prices, monetary penalties, and management responsibilities. One Web Service can be associated with multiple service offerings. For easier specification of similar service offerings, WSOL enables specification of constraint groups (CGs) and constraint group templates (CGTs). We have also developed a format for representation of dynamic relationships between service offerings. WSOL service offerings are simple contracts and SLAs (Service Level Agreements) between Web Services. Describing a Web Service in WSOL, in addition to WSDL, enables monitoring, metering, and management of Web Services. The Web Service, its consumer, or one or more designated third parties (usually SOAP message intermediaries) can meter QoS metrics and evaluate constraints in WSOL service offerings. Further, manipulation of service offerings can be used for dynamic adaptation and management of Web Service compositions. In addition, WSOL supports selection of a more appropriate Web Service and service offering for particular circumstances. The main distinctive characteristics of WSOL, compared to recent related works, are its expressive power, features that reduce run-time overhead, and orientation towards management applications.*

23. Vladimir Tosic, Kruti Patel, Bernard Pagurek. *WSOL - Web Service Offerings Language.* CAiSE '02/ WES '02: Revised Papers from the International Workshop on Web Services, E-Business, and the Semantic Web, pages 57–67, Springer-Verlag, 2002.

    Abstract: *WSOL (Web Service Offerings Language) is an XML (Extensible Markup Language) notation compatible with the WSDL (Web Services Description Language) standard. While WSDL is used for describing operations provided by Web Services, WSOL enables formal specification of multiple classes of service for one Web Service. A service offering is a formal representation of one class of service for a Web Service. As classes of service for Web Services are determined by combinations of various constraints, WSOL enables formal specification of functional constraints, some QoS (a.k.a., non-functional) constraints, simple access rights (for differentiation of service), price, and relationships with other service offerings of the same Web Service. Describing a Web Service in WSOL, in addition to WSDL, enables selection of a more appropriate Web Service and service offering for particular circumstances. Further, it supports dynamic adaptation and management of Web Service compositions using manipulation of service offerings.*

Books and Monographs:

1. J. Cardoso. *Quality of Service and Semantic Composition of Workflows.* Ph.D. Thesis, University of Georgia, 2002.

   Abstract: *Workflow management systems (WfMSs) have been used to support a variety of business processes. As organizations adopt new working models, such as e-commerce, new challenges arise for workflow systems. These challenges include support for the adequate management of quality of service (QoS) and the development of new solutions to facilitate the composition of workflow applications involving Web services. The good management of QoS directly impacts the success of organizations participating in e- commerce activities by better fulfilling customer expectations and achieving customer satisfaction. To enable adequate QoS management, research is required to develop mechanisms that specify, compute, monitor, and control the QoS of the products or services to be delivered. The composition of workflows to model e-service applications differs from the design of traditional workflows due to the number of Web services available during the composition process and to their heterogeneity. Two main problems need to be solved: how to efficiently discover Web services and how to facilitate their interoperability. To enhance WfMSs with QoS* *management, we have developed a QoS model that allows for the description of nonfunctional aspects of workflow components, from a quality of service perspective. To automatically compute the overall QoS of a workflow, we have developed a mathematical model and implemented an algorithm (SWR algorithm). Our QoS model and mathematical model have been validated with the deployment and execution of a set of production workflows in the area of genetics. The analysis of the collected data proves that our models provide a suitable framework for estimating, predicting, and analyzing the QoS of production workflows. To support, facilitate, and assist the composition of workflows involving Web services, we present a solution based on ontologies. We have developed an algorithm that workflow systems and discovery mechanisms can use to find Web services with desired interfaces and operational metrics, and to assist designers in resolving heterogeneity issues among Web services. Our approach provides an important solution to enhance Web service discovery and interoperability.*

Publications in Refereed Workshops:

1. M. Tian, A. Gramm, T. Naumowicz, H. Ritter, J. Schiller. *A Concept for QoS Integration in Web Services.* Web Information Systems Engineering Workshops, pages 149–155, December 2003.

   Abstract: *With the growing popularity of Web services, a general QoS support for Web services plays an important role for the success of this emerging technology. Unfortunately, current Web service environments do not offer comprehensive QoS support. In this paper, we present an approach that does not only enable the QoS integration in Web services, but also the selection of appropriate services based on QoS requirements regarding server and network performance. Furthermore, we present how application requirements regarding communication QoS are mapped onto the underlying QoS aware network at runtime, as well as how users can obtain real-time information about server performance in order to monitor the accomplishment of assured services, giving the user an instant QoS feedback.*

2. Sabata, B., Chatterjee, S., Davis, M., Sydir, J.J., Lawrence, T.F.. *Taxonomy for QoS specifications.* pages 100–107, 1997.

   Abstract: *It is becoming increasingly commonplace for multiple applications with different quality of service (QoS) requirements to share the resources of a distributed system. Within this environment, the resource management algorithms must take into account the QoS desired by applications and the ability of the system resources to provide it. We present a taxonomy for specifying QoS for the different components of a distributed system, from the applications down to the resources. We specify QoS as a combination of metrics and policies. QoS metrics are used to specify performance parameters, security requirements and the relative importance of the work in the system. We define three types of QoS performance parameters: timeliness, precision, and accuracy. QoS policies capture application-specific policies that govern how an application is treated by the resource manager. Examples of such policies are management policies and the levels of service. We explore each of these components of the QoS taxonomy in detail*

Technical Reports and Manuals:

1. Siddharth Bajaj, Don Box, Dave Chappell, Francisco Curbera, Glen Daniels, Phillip Hallam-Baker, Maryann Hondo, Chris Kaler, Dave Langworthy, Anthony Nadalin, Nataraj Nagaratnam, Hemma Prafullchandra, Claus von Riegen, Daniel Roth, Jeffrey Schlimmer, Chris Sharp, John Shewchuk, Asir Vedamuthu, Ümit Yal cınalp, David Orchard . *Web Services Policy Framework (WS- Policy).* IBM, March 2006.

   Abstract: *The Web Services Policy Framework (WS-Policy) provides a general purpose model and corresponding syntax to describe the policies of a Web Service. WS-Policy defines a base set of constructs that can be used and extended by other Web services specifications to describe a broad range of service requirements and capabilities.*

2. Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, Ming Xu. *Web Services Agreement Specification*

(WS-Agreement). Grid Resource Allocation Agreement Protocol, September 2005.
Abstract: *This document describes Web Services Agreement Specification (WS-Agreement), a Web Services protocol for establishing agreement between two parties, such as between a service provider and consumer, using an extensible XML language for specifying the nature of the agreement, and agreement templates to facilitate discovery of compatible agreement parties. The specification consists of three parts which may be used in a composable manner: a schema for specifying an agreement, a schema for specifying an agreement template, and a set of port types and operations for managing agreement life-cycle, including creation, expiration, and monitoring of agreement states.*

3. The OASIS Group. *Quality Model for Web Services.* The Oasis Group, September 2005.
Abstract: *The purpose of this document is to provide a model for Web services quality management and quality factors in the process of developing and using Web services. We define the consistent and systematic conceptual model of Web services quality, which may be used by intimate associates, i.e. stakeholders, developers, service providers, and customers of Web services.*

4. The OMG Group. *UML*$^{\text{TM}}$ *Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms.* Num. ptc/2005-05-02, The OMG Group, May 2005.

5. The Oasis Group. *Summary of Quality Model for Web Services.* The Oasis Group, 2005.

6. Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, Richard Franck . *Web Service Level Agreement (WSLA) Language Specification.* IBM Corporation, 2003.
Abstract: *This document describes the specification language for service level agreements for Web Services, the Web Service Level Agreement (WSLA) language. WSLAs are agreements between a service provider and a customer and as such define the obligations of the parties involved. Primarily, this is the obligation of a service provider to perform a service according to agreed-upon guarantees for IT-level service parameters (such as availability, response time and throughput) for Web Services.*
An SLA also specifies the measures to be taken in case of deviation and failure to meet the asserted service guarantees, for example, a notification of the service customer. The assertions of the service provider are based on a detailed definition of service parameters including the algorithms – how basic metrics should be measured in systems and how they are aggregated into composite metrics and SLA parameters. In addition, a WSLA can express the operations of monitoring and managing the service. This may include third parties (such as Management Service Providers) that contribute to the measurement of metrics, supervision of guarantees or even the management of deviations of service guarantees. These multi-party constellations necessitate the definition of the interactions among the parties supervising the WSLA.
However, a WSLA only covers the agreed common view of a service between the parties involved. To actually act as a participant in a WSLA, parties have various degrees of freedom to define an implementation policy for a service and its supervision. Typically, the obligations of a WSLA must be translated into system-level configuration information, which can be proprietary to each party involved.

7. Akhil Sahai, Anna Durante, Vijay Machiraju. *Towards Automated SLA Management for Web Services.* Num. HPL-2001-310, HP Laboratories, July 2002.
Abstract: *In order to automate SLA management it is essential to specify SLAs in precise and unambiguous manner as well as keep the specification flexible. While precision will help automate the process of monitoring and metric collection, flexibility will enable extending it to unforeseen service level agreement specifications.*

8. Anbazhagan Mani, Arun Nagarajan. *Understanding quality of service for Web services.* IBM, 2002.
Abstract: *With the widespread proliferation of Web services, quality of service (QoS) will become a significant factor in distinguishing the success of service providers. QoS determines the service usability and utility, both of which influence the popularity of the service. In this article, we look at the various Web service QoS requirements, bottlenecks affecting performance of Web services, approaches of providing service quality, transactional services, and a simple method of measuring response time of your Web services using the service proxy.*

9. Svend Frolund, Jari Koistinen. *QML: A Language for Quality of Service Specification.* Num. HPL-98-10, 63 pages, HP Laboratories, 1998.

   Abstract: *To be competitive, future software system must provide not only the correct functionality, but also an adequate level of quality of service (QoS). By QoS, we refer to non-functional properties, such as reliability, performance, timing, and security. To provide an adequate level of QoS, software systems need to include capabilities such as QoS negotiation, monitoring, and adaptation. These capabilities all require the expected and the provided QoS levels to be explicitly specified. QoS can be specified statically at the time of implementation, design, or dynamically at deployment or runtime. To facilitate QoS specification, we present a general Quality of service Modeling Language (QML) for defining multi-category QoS specifications for components in distributed object systems. QML is designed to support QoS in general, encompassing QoS categories such as reliability, performance, security, and timing. QoS specification in QML facilitate the static decomposition of a software system into components with precisely specified QoS boundaries. They also facilitate dynamic QoS functions, such as negotiations, monitoring, and adaptation. QML is designed for a good fit with object-oriented distributed architectures and concepts such as interfaces and inheritance. It also allows specification at a fine-grained level for operations, operation arguments, and attributes. QML enables user- defined QoS categories, and allows specifications within those categories to be associated with component interface definitions. In addition, checks can be made dynamically to determine whether one QML specification satisfies another. This mechanism allows us to dynamically match QoS requirements and offers*

15 Publications in **Semantic Web**

Articles in Refereed Journals:

1. Groenmo, R., Jaeger, M.C.. *Model-driven semantic Web service composition.* Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific, To Appear, December 2005.

   Abstract: *As the number of available Web services increases there is a growing demand to realise complex business processes by combining and reusing available Web services. The reuse and combination of services results in a composition of Web services that may also involve services provided in the Internet. With semantically described Web services, an automated matchmaking of capabilities can help identify suitable services. To address the need for semantically defined Web services, OWL-S and WSML have been proposed as competing semantic Web service languages. We show how the proposed semantic Web service languages can be utilized within a model-driven methodology for building composite Web services. In addition we combine the semantic-based discovery with the support for processing QoS requirements to apply a ranking or a selection of the candidates. The methodology describes a process which guides the developer through four phases, starting with the initial modelling, and ending with a new composite service that can be deployed and published to be consumed by other users.*

Articles in Refereed Conferences:

1. Kyriakos Kritikos, Dimitris Plexousakis. *OWL-Q for Semantic QoS-based Web Service Description and Discovery.* First International Joint Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, November 2007.

   Abstract: *Semantic Web Services are emerging for their promise to produce a more accurate and precise Web Service discovery process. However, most of research approaches focus only on the functional part of semantic Web Service description. The above fact along with the proliferation of Web Services is highly probable to lead to a situation where Web Service registries will return many functionally-equivalent Web Service advertisements for each user request. This problem can be solved with the semantic description of QoS for Web Services. QoS is a set of non-functional properties encompassing performance and network-related characteristics of resources. So it can be used for distinguishing between functionally-equivalent Web Services. Current research approaches for QoS-based Web Service description are either syntactic or poor or non-extensible. To solve this problem, we have developed a rich and extensible ontological specification called OWL-Q for semantic QoS-based Web Service description. We analyze all OWL-Q parts and reason that rules should be added in order to support property inferencing and constraint enforcement. Finally, we line out our under-development semantic framework for QoS-based Web Service description and discovery.*

2. Ester Giallonardo, Eugenio Zimeo. *More Semantics in QoS Matching.* SOCA, pages 163–171, 2007.

   Abstract: *The evolution of the Web towards a global computing environment is promoting new research efforts aimed at the formal characterization of Web Services QoS. Reasoning on QoS is a key to improve matching process during the discovery of desired services and a step towards the transformation of applications in collections of loosely coupled services virtually connected by semantic similarities. The paper presents the on QoS ontology, an openly available OWL ontology for QoS, and evaluates it in a QoS-aware matching environment. The ontology can be used to express functions of QoS metrics useful to improve the recall tied to the matching of a template request with target Web Services. To this end, the ontology introduces the concept of derivation in the matching process. This gives the possibility of matching a QoS template with published Web Services by deriving different QoS parameters when a one-to-one matching fails. The proposed matching algorithm utilizes a reasoner that exploits the ontology to avoid apparent mismatches. An experimental evaluation shows that exploiting QoS knowledge significantly improves matching recall without deteriorating precision.*

3. Srini Narayanan, Sheila A. McIlraith. *Simulation, verification and automated composition of web services.* WWW '02: Proceedings of the 11th international conference on World

Wide Web, pages 77–88, ACM, 2002.

Abstract: *Web services – Web-accessible programs and devices - are a key application area for the Semantic Web. With the proliferation of Web services and the evolution towards the Semantic Web comes the opportunity to automate various Web services tasks. Our objective is to enable markup and automated reasoning technology to describe, simulate, compose, test, and verify compositions of Web services. We take as our starting point the DAML-S DAML+OIL ontology for describing the capabilities of Web services. We define the semantics for a relevant subset of DAML-S in terms of a first-order logical language. With the semantics in hand, we encode our service descriptions in a Petri Net formalism and provide decision procedures for Web service simulation, verification and composition. We also provide an analysis of the complexity of these tasks under different restrictions to the DAML-S composite services we can describe. Finally, we present an implementation of our analysis techniques. This implementation takes as input a DAML-S description of a Web service, automatically generates a Petri Net and performs the desired analysis. Such a tool has broad applicability both as a back end to existing manual Web service composition tools, and as a stand-alone tool for Web service developers.*

16 Publications in Services

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

2. Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann. *Service-Oriented Computing: State of the Art and Research Challenges.* IEEE Computer, To Appear, November 2007.

   Abstract: *Service-oriented computing promotes the idea of assembling application components into a network of services that can be loosely coupled to create flexible, dynamic business processes and agile applications that span organizations and computing platforms. An SOC research road map provides a context for exploring ongoing research activities.*

3. Roberto Lucchi, Manuel Mazzara. *A pi-calculus based semantics for WS-BPEL .* Journal of Logic and Algebraic Programming, Vol. 70, Num. 1, pages 96–118, January 2007.

   Abstract: *Recently, the term Web services orchestration has been introduced to address some issues related to Web services composition, that is the way of defining a complex service out of simpler ones. Several proposals for describing orchestration for business processes have been presented in the last years and many of these languages make use of concepts as long-running transactions and compensations for coping with error handling. WS-BPEL 2.0, the most credited candidate for becoming a standard, provides three different mechanisms allowing to cope with abnormal situations: exception, event and compensation handling. This complexity makes it difficult to formally define the framework, thus limiting the formal reasoning about the designed applications. In this paper we advocate that three different mechanisms for error handling are not necessary and we formalize a novel orchestration language based on the idea of event notification as the unique error handling mechanism. To this end, we formally define the three BPEL mechanisms in terms of our calculus. It is possible to take advantages of this formal description in two ways. Firstly, this language represents by itself a proposal of simplification for WS-BPEL 2.0 including an unambiguous specification. Secondly, an implementor of an actual WS-BPEL 2.0 orchestration engine could implement simply this single mechanism providing all the remaining ones by compilation. With this attempt we intend to give a concrete contribute towards the improvement of the quality of the BPEL specification, the applicability of BPEL itself and the implementation of real orchestration engines. Finally, as a case study we consider some of the hundreds of open issues met by the WS-BPEL designers and we propose a solution making use of the experience gained developing our algebra.*

4. D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, P. Plebani. *PAWS: A Framework for Executing Adaptive Web-Service Processes.* IEEE Software, To Appear, 2007.

Abstract: *The processes with adaptive Web services framework couples design-time and runtime mechanisms to flexibly and adoptively execute managed Web-services-based business processes.*

5. J. Cardoso. *Complexity analysis of BPEL Web processes.* Software Process: Improvement and Practice, Vol. 12, Num. 1, pages 35–49, Whiley-Interscience, 2007.

   Abstract: *Several organizations have already realized the potential of using WS-BEPL, the Process Execution Language for Web Services, to model the behavior of Web services in business processes. WS-BPEL provides a model for describing simple or complex interactions between business partners. In some cases, WS-BPEL process designs can be highly complex, due, for example, to the vast number of Web services carried out in global markets. High complexity in a process has several undesirable drawbacks; it may result in poor understandability, more errors, defects, and exceptions, leading to processes requiring more time to be developed, tested, and maintained. Therefore, excessive complexity should be avoided. Processes that are highly complex tend be less flexible, since it is more complicated to make changes to the process. The major goal of this article is to present two metrics to analyze the control-flow complexity (CFC) of WS-BPEL Web processes. The metrics are to be used at design-time to evaluate the complexity of a process design before implementation actually takes place.*

6. Mike P. Papazoglou, Benedikt Kratz. *Web services technology in support of business transactions.* Service Oriented Computing and Applications, Vol. 1, Num. 1, pages 51–63, 2007.

   Abstract: *Advanced business applications typically involve well-defined business functions such as payment processing, shipping and tracking, determining new product offerings, granting/extending credit, managing market risk and so on. These reflect commonly standard business functions that apply to a variety of application scenarios. Although such business functions drive transactional applications between trading partners they are completely external to current Web services transaction mechanisms and are only expressed as part of application logic. To remedy this situation, this paper proposes a business-aware Web services transaction model and support mechanisms, which is driven by common business functions. The model allows expressing business functions such as payment and credit conditions, delivery conditions, business agreements stipulated in SLAs, liabilities and dispute resolution policies. It allows blending these business functions with QoS criteria such as security support to guarantee integrity of information, confidentiality, and non-repudiation.*

7. Ma, K.J.. *Web services: what's real and what's not?.* IT Professional, Vol. 7, Num. 2, pages 14–21, March 2005.

   Abstract: *The idea of abstracted, well-defined, and ubiquitously invokable services replacing proprietary interprocess communications has been a goal of system designers for a long time. The rise of Web services has led to a lot of misconceptions about how they can and cannot support the holy grail of a service-oriented architecture (SOA). This article seeks to put Web services in perspective, explaining their current capabilities and what industry can expect from them in the near term. It gives an overview of how technologies such as the Extensible Markup Language (XML), XML schemas, Extensible Stylesheet Language Transformations (XSLT), the Simple Object Access Protocol (SOAP), the Web Services Description Language (WSDL), and universal description, discovery, and integration (UDDI) fit into the equation for an SOA.*

8. Papazoglou, M.P., van den Heuvel, W.-J.. *Web services management: a survey.* Internet Computing, IEEE, Vol. 9, Num. 6, pages 58–64, November/December 2005.

   Abstract: *Solutions based on service-oriented architectures are promising in that they leverage common services and enable collaborative business processes that cross organizational boundaries. However, because Web services applications can span multiple hosts, operating systems, languages, and enterprises, it's problematic to measure, control, and manage application availability and performance. In addition to discussing the relationship of Web services management to traditional distributed systems management, this survey explores various Web services management approaches and their underlying architectural concepts.*

9. James Pasley. *How BPEL and SOA Are Changing Web Services Development.* IEEE Internet Computing, Vol. 9, Num. 3, pages 60–67, 2005.

   Abstract: *As the use of Web services grows, organizations are increasingly choosing the Business Process Execution Language for modeling business processes within the Web services architecture. In addition to* *orchestrating organizations' Web services, BPEL's strengths include asynchronous message handling, reliability, and recovery. By developing Web services with BPEL in mind, organizations can implement aspects of the service-oriented architecture that might previously have been difficult to achieve.*

10. Neal Leavitt. *Are Web Services Finally Ready to Deliver?.* IEEE Computer, Vol. 37, Num. 11, pages 14–18, November 2004.

    Abstract: *Web services, in brief, are a framework of software technologies designed to support interoperable machine-to-machine interaction over a network. Companies on different systems can use Web services to exchange information online with business partners, customers, and suppliers. Various standards organizations and industry consortia are developing Web services specifications without a unifying authority. Organizations such as the World Wide Web Consortium (W3C), the Organization for the Advancement of Structured Information Standards (OASIS), the Liberty Alliance Project, and the Web Services Interoperability Organization (WS-I) have developed or reviewed numerous standards. A primary goal of Web services is to unlock a new generation of e-commerce applications.*

11. Alexander Keller, Heiko Ludwig. *The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services.* J. Network Syst. Manage., Vol. 11, Num. 1, 2003.

    Abstract: *We describe a novel framework for specifying and monitoring Service Level Agreements (SLA) for Web Services. SLA monitoring and enforcement becomes increasingly important in a Web Service environment where enterprises rely on services that may be subscribed dynamically and on demand. For economic and practical reasons, we want an automated provisioning process for both the service itself as well as the SLA management system. It measures and monitors the QoS parameters, checks the agreed-upon service levels, and reports violations to the authorized parties involved in the SLA management process. The Web Service Level Agreement (WSLA) framework, our approach to these issues, is targeted at defining and monitoring SLAs for Web Services. Although WSLA has been designed for a Web Services environment, it is applicable as well to any inter-domain management scenario such as business process and service management or the management of networks, systems and applications in general. The WSLA framework consists of a flexible and extensible language based on XML Schema and a runtime architecture comprising several SLA monitoring services, which may be outsourced to third parties to ensure a maximum of accuracy. WSLA enables service customers and providers to unambiguously define a wide variety of SLAs, specify the SLA parameters and the way how they are measured, and relate them to managed resource instrumentations. Upon receipt of an SLA specification, the WSLA monitoring services are automatically configured to enforce the SLA. An implementation of the WSLA framework, the SLA Compliance Monitor, is publicly available as part of the IBM Web Services Toolkit.*

Articles in Refereed Conferences:

1. Sheng Chen, Liang Bao, Ping Chen. *OptBPEL: A Tool for Performance Optimization of BPEL Process.* Software Composition, pages 141–148, 2008.

   Abstract: *The Business Process Execution Language (BPEL) is now a de facto standard for specifying and executing business process for web service composition and orchestration. As more and more web services are composed using BPEL, tuning these compositions and gain better performance becomes increasingly important. This paper presents our approach for optimizing the BPEL process and introduces OptBPEL, a tool for performance optimization of BPEL process. The approach starts from the optimization of synchronization structure concerning link in BPEL. After that, some concurrency analysis techniques are applied to obtain further performance improvement. Finally, we give some experiments and prove the efficiency of these optimization algorithms used in OptBPEL.*

2. Cinzia Cappiello, Marco Comuzzi, Pierluigi Pleban. *On Automated Generation of Web Service Level Agreements .* Advanced Information Systems Engineering, Vol. 4495, Springer Verlag, June 2007.

   Abstract: *Before a service invocation takes place, an agreement between the service provider and the service user might be required. Such an agreement is the result of a negotiation process between the two parties and defines how the service invocation has to occur. Considering the Service Oriented Computing paradigm, the* *relationship among providers and users is extremely loose. Traditional agreements are likely to concern long term relationships and to be manually performed. In this paper, we propose a model to generate service level agreement on-the-fly. Just before the invocation commences, the quality of the service is negotiated in order to generate a service level agreement tied to that specific invocation. Such an approach relies on a quality model that supports both users requirements and providers capabilities definition.*

3. Fabio Barbon, Paolo Traverso, Marco Pistore, Michele Trainotti. *Run-Time Monitoring of Instances and Classes of Web Service Compositions.* ICWS, pages 63–71, 2006.

   Abstract: *The run-time monitoring of web service compositions has been widely acknowledged as a significant and challenging problem. In this paper, we propose a novel solution to the problem of monitoring web services implemented in BPEL. We devise an architecture that clearly separates the business logic of a web service from its monitoring functionality. The architecture supports both ”instance monitors” that deal with the execution of a single instance of BPEL process, as well as ”class monitors” that report aggregated information about all the instances of a BPEL process. We also define a language for the specification of instance and class monitors. The language allows for specifying boolean, statistic, and time-related properties. Finally, we devise a technique for the automatic translation of all these kinds of monitors to Java programs.*

4. Jorge Cardoso. *About the Data-Flow Complexity of Web Processes.* 6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility, pages 67–74, 2005.

   Abstract: *Abstract. Organizations are increasingly faced with the challenge of managing e-commerce and e-business applications involving Web services and Web processes. In some cases, Web processes design can be highly complex, due, for example, to the vast number of services carried out in global markets. High complexity in a process has several undesirable drawbacks, it may result in bad understandability, more errors, defects, and exceptions leading processes to need more time to develop, test, and maintain. Therefore, excessive complexity should be avoided. Flexibility and complexity are guiding principles in the design of business processes. In most business processes, flexibility and complexity are related inversely. Processes with a high complexity tend be less flexible, since it is more complicated to make changes to the process. In our previous work we have defined a metric to measure the control-flow complexity of Web processes. The major goal of this paper is to study the issues and establish the requirements for the development of a metric to analyze the data-flow complexity of Web processes.*

5. Dimka Karastoyanova, Alejandro Houspanossian, Mariano Cilia, Frank Leymann, Alejandro P. Buchmann. *Extending BPEL for Run Time Adaptability.* EDOC, pages 15–26, 2005.

   Abstract: *The existing Web Service Flow (WS-flow) technologies enable both static and dynamic binding of participating Web services (WSs) on the process model level. Adaptability on per-instance basis is not sufficiently supported and therefore must be addressed to improve process flexibility upon changes in the environment. Ad-hoc process instance changes can be enabled by swapping participating WS instances, by modifying port Types of the partners to be invoked, and by changing process logic. In this work we address the problem of dynamic binding of WSs to WS-flow instances at run time, i.e. the ability to exchange a WS instance participating in a WS-flow instance with an alternative one. The problem is additionally complicated by the fact that the execution of a process depends on its deployment. We describe the ”find and bind” mechanism, and we show its representation as a BPEL extension. We discuss the benefits that could be gained and the disadvantages it brings in. The mechanism extends and improves the existing process technologies. It facilitates a precisely controlled policy-based selection of WSs at run time and also provides for process instance repair, while maintaining simplicity. We also discuss a prototypical implementation of the presented functionality.*

6. Khaled Mahbub, George Spanoudakis. *Run-time Monitoring of Requirements for Systems Composed of Web-Services: Initial Implementation and Evaluation Experience.* ICWS, pages 257–265, 2005.

   Abstract: *This paper describes a framework supporting the runtime monitoring of requirements for systems implemented as compositions of Web-services specified in BPEL. The requirements that can be monitored are* *specified in event calculus. The paper presents an overview of the framework and describes the architecture and implementation of a tool that we have developed to operationalise it. It also presents the results of a preliminary experimental evaluation of the framework.*

7. Mike P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann, Bernd J. Krämer. *05462 Service-Oriented Computing: A Research Roadmap.* Service Oriented Computing, 2005.

   Abstract: *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.*

8. Florian Rosenberg, Schahram Dustdar. *Business Rules Integration in BPEL - A Service-Oriented Approach.* CEC, pages 476–479, 2005.

   Abstract: *Business rules change quite often. These changes cannot be handled efficiently by representing business rules embedded in the source code of the business logic. Efficient handling of rules that govern ones business is one factor for success. That is where business rules engines play an important role. The service-oriented computing paradigm is becoming more and more popular. Services offered by different providers, are composed to new services by using Web service composition languages such as BPEL. Such process-based composition languages lack the ability to use business rules managed by different business rules engines in the composition process. In this paper, we propose an approach on how to use and integrate business rules in a service-oriented way into BPEL.*

9. Vladimir Tosic, Bernard Pagurek, Kruti Patel. *WSOL - A Language for the Formal Specification of Classes of Service for Web Services.* ICWS, pages 375–381, 2003.

   Abstract: *We are developing a language, Web Service Offerings Language (WSOL), for the formal specification of various constraints, management statements, and classes of service for Web Services. WSOL is an XML (Extensible Markup Language) notation compatible with WSDL (Web Services Description Language). A service offering in WSOL is a formal description of one class of service of a Web Service. It contains formal representation of various constraints: functional (pre-, post-, and future- conditions), Quality of Service (QoS, a.k.a. non-functional, extra- functional), and access rights. It also contains management statements, such as statements about prices, monetary penalties, and management responsibilities. One Web Service can be associated with multiple service offerings. For easier specification of similar service offerings, WSOL enables specification of constraint groups (CGs) and constraint group templates (CGTs). We have also developed a format for representation of dynamic relationships between service offerings. WSOL service offerings are simple contracts and SLAs (Service Level Agreements) between Web Services. Describing a Web Service in WSOL, in addition to WSDL, enables monitoring, metering, and management of Web Services. The Web Service, its consumer, or one or more designated third parties (usually SOAP message intermediaries) can meter QoS metrics and evaluate constraints in WSOL service offerings. Further, manipulation of service offerings can be used for dynamic adaptation and management of Web Service compositions. In addition, WSOL supports selection of a more appropriate Web Service and service offering for particular circumstances. The main distinctive characteristics of WSOL, compared to recent related works, are its expressive power, features that reduce run-time overhead, and orientation towards management applications.*

10. Massimo Mecella, Francesco Parisi Presicce an Barbara Pernici. *Modeling E -service Orchestration through Petri Nets .* Technologies for E-Services, Lecture Notes in Computer Science, Vol. 2444, pages 109–134, Springer Verlag, July 2002.

    Abstract: *B2B interaction requires new forms of coordination between participating organizations. Indeed, the main requirement is that the autonomy of each participating partner is preserved during the interaction, guaranteeing at the same time that the overall goals of the common process are reached. Mechanisms for regulating distributed workflow evolution when the workflow is composed of the invocation of E-services of different organizations are needed. The E-service orchestration model proposed in this paper provides a mechanism for supporting control of process evolution in terms both of control and data flows, and for distributing and assigning process responsibilities.*

Technical Reports and Manuals:

1. Siddharth Bajaj, Don Box, Dave Chappell, Francisco Curbera, Glen Daniels, Phillip Hallam-Baker, Maryann Hondo, Chris Kaler, Dave Langworthy, Anthony Nadalin, Nataraj Nagaratnam, Hemma Prafullchandra, Claus von Riegen, Daniel Roth, Jeffrey Schlimmer, Chris Sharp, John Shewchuk, Asir Vedamuthu, Ümit Yal cınalp, David Orchard . *Web Services Policy Framework (WS- Policy).* IBM, March 2006.

   Abstract: *The Web Services Policy Framework (WS-Policy) provides a general purpose model and corresponding syntax to describe the policies of a Web Service. WS-Policy defines a base set of constructs that can be used and extended by other Web services specifications to describe a broad range of service requirements and capabilities.*

2. The OASIS Group. *Quality Model for Web Services.* The Oasis Group, September 2005.

   Abstract: *The purpose of this document is to provide a model for Web services quality management and quality factors in the process of developing and using Web services. We define the consistent and systematic conceptual model of Web services quality, which may be used by intimate associates, i.e. stakeholders, developers, service providers, and customers of Web services.*

3. The Oasis Group. *Summary of Quality Model for Web Services.* The Oasis Group, 2005.

4. Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, Richard Franck . *Web Service Level Agreement (WSLA) Language Specification.* IBM Corporation, 2003.

   Abstract: *This document describes the specification language for service level agreements for Web Services, the Web Service Level Agreement (WSLA) language. WSLAs are agreements between a service provider and a customer and as such define the obligations of the parties involved. Primarily, this is the obligation of a service provider to perform a service according to agreed-upon guarantees for IT-level service parameters (such as availability, response time and throughput) for Web Services.*

   An SLA also specifies the measures to be taken in case of deviation and failure to meet the asserted service guarantees, for example, a notification of the service customer. The assertions of the service provider are based on a detailed definition of service parameters including the algorithms – how basic metrics should be measured in systems and how they are aggregated into composite metrics and SLA parameters. In addition, a WSLA can express the operations of monitoring and managing the service. This may include third parties (such as Management Service Providers) that contribute to the measurement of metrics, supervision of guarantees or even the management of deviations of service guarantees. These multi-party constellations necessitate the definition of the interactions among the parties supervising the WSLA.

   However, a WSLA only covers the agreed common view of a service between the parties involved. To actually act as a participant in a WSLA, parties have various degrees of freedom to define an implementation policy for a service and its supervision. Typically, the obligations of a WSLA must be translated into system-level configuration information, which can be proprietary to each party involved.

5. Akhil Sahai, Anna Durante, Vijay Machiraju. *Towards Automated SLA Management for Web Services.* Num. HPL-2001-310, HP Laboratories, July 2002.

   Abstract: *In order to automate SLA management it is essential to specify SLAs in precise and unambiguous* *manner as well as keep the specification flexible. While precision will help automate the process of monitoring and metric collection, flexibility will enable extending it to unforeseen service level agreement specifications.*

6. Anbazhagan Mani, Arun Nagarajan. *Understanding quality of service for Web services.* IBM, 2002.

   Abstract: *With the widespread proliferation of Web services, quality of service (QoS) will become a significant factor in distinguishing the success of service providers. QoS determines the service usability and utility, both of which influence the popularity of the service. In this article, we look at the various Web service QoS requirements, bottlenecks affecting performance of Web services, approaches of providing service quality, transactional services, and a simple method of measuring response time of your Web services using the service proxy.*

17 Publications in **Service Level Agreement**

Articles in Refereed Journals:

1. D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, P. Plebani. *PAWS: A Framework for Executing Adaptive Web-Service Processes.* IEEE Software, To Appear, 2007.
   Abstract: *The processes with adaptive Web services framework couples design-time and runtime mechanisms to flexibly and adoptively execute managed Web-services-based business processes.*

2. Giorgia Lodi, Fabio Panzieri, Davide Rossi, Elisa Turrini. *SLA-Driven Clustering of QoS-Aware Application Servers.* IEEE Trans. Software Eng., Vol. 33, Num. 3, pages 186–197, 2007.
   Abstract: *In this paper, we discuss the design, implementation, and experimental evaluation of a middleware architecture for enabling service level agreement (SLA)-driven clustering of QoS-aware application servers. Our middleware architecture supports application server technologies with dynamic resource management: application servers can dynamically change the amount of clustered resources assigned to hosted applications on-demand so as to meet application-level quality of service (QoS) requirements. These requirements can include timeliness, availability, and high throughput and are specified in SLAs. A prototype of our architecture has been implemented using the open-source J2EE application server JBoss. The evaluation of this prototype shows that our approach makes possible JBoss' resource usage optimization and allows JBoss to effectively meet the QoS requirements of the applications it hosts, i.e., to honor the SLAs of those applications.*

3. Wenhui Sun, Jinyu Zhang, Feng Liu. *WS-SLA: A Framework for Web Services Oriented Service Level Agreements.* Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference on, To Appear, May 2006.
   Abstract: *Service level agreements (SLA) becomes the prevailing business model for delivery of a large number of products and services in telecommunication and IT arenas. We presents an integrated service level agreement model applied to operating supporting system (OSS), on the basis of WfMC workflow reference model and service oriented architecture (SOA). And an application scenario instance of SLA system architecture is described by a set of operations models by Business Process Execution Language (BPEL)*

4. Alexander Keller, Heiko Ludwig. *The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services.* J. Network Syst. Manage., Vol. 11, Num. 1, 2003.
   Abstract: *We describe a novel framework for specifying and monitoring Service Level Agreements (SLA) for Web Services. SLA monitoring and enforcement becomes increasingly important in a Web Service environment where enterprises rely on services that may be subscribed dynamically and on demand. For economic and practical reasons, we want an automated provisioning process for both the service itself as well as the SLA management system. It measures and monitors the QoS parameters, checks the agreed-upon service levels, and reports violations to the authorized parties involved in the SLA management process. The Web Service Level Agreement (WSLA) framework, our approach to these issues, is targeted at defining and monitoring SLAs for Web Services. Although WSLA has been designed for a Web Services environment, it is applicable as well to any inter-domain management scenario such as business process and service management or the management of networks, systems and applications in general. The WSLA framework consists of a flexible and extensible language based on XML Schema and a runtime architecture comprising several SLA monitoring services, which may be outsourced to third parties to ensure a maximum of accuracy. WSLA enables service customers and providers to unambiguously define a wide variety of SLAs, specify the SLA parameters and the way how they are measured, and relate them to managed resource instrumentations. Upon receipt of an SLA specification, the WSLA monitoring services are automatically configured to enforce the SLA. An implementation of the WSLA framework, the SLA Compliance Monitor, is publicly available as part of the IBM Web Services Toolkit.*

Articles in Refereed Conferences:

1. Cinzia Cappiello, Marco Comuzzi, Pierluigi Pleban. *On Automated Generation of Web Service Level Agreements .* Advanced Information Systems Engineering, Vol. 4495, Springer Verlag, June 2007.

   Abstract: *Before a service invocation takes place, an agreement between the service provider and the service user might be required. Such an agreement is the result of a negotiation process between the two parties and defines how the service invocation has to occur. Considering the Service Oriented Computing paradigm, the relationship among providers and users is extremely loose. Traditional agreements are likely to concern long term relationships and to be manually performed. In this paper, we propose a model to generate service level agreement on-the-fly. Just before the invocation commences, the quality of the service is negotiated in order to generate a service level agreement tied to that specific invocation. Such an approach relies on a quality model that supports both users requirements and providers capabilities definition.*

2. Elisabetta Di Nitto, Massimiliano Di Penta, Alessio Gambi, Gianluca Ripa, Maria Luisa Villani. *Negotiation of Service Level Agreements: An Architecture and a Search-Based Approach.* ICSOC, pages 295–306, 2007.

   Abstract: *Software systems built by composing existing services are more and more capturing the interest of researchers and practitioners. The envisaged long term scenario is that services, offered by some competing providers, are chosen by some consumers and used for their own purpose, possibly, in conjunction with other services. In the case the consumer is not anymore satisfied by the performance of some service, he can try to replace it with some other service. This implies the creation of a global market of services and poses new requirements concerning validation of exploited services, security of transactions engaged with services, trustworthiness, creation and negotiation of Service Level Agreements with these services. In this paper we focus on the last aspect and present our approach for negotiation of Service Level Agreements. Our architecture supports the actuation of various negotiation processes and offers a search-based algorithm to assist the negotiating parts in the achievement of an agreement.*

3. Mohan Baruwal Chhetri, Jian Lin, SukKeong Goh, Jian Ying Zhang, Ryszard Kowalczyk, Jun Yan. *A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation ofWeb Service Composition.* ASWEC, pages 90–99, 2006.

   Abstract: *Recent progress in the field of Web services has made it possible to integrate inter-organizational and heterogeneous services on the Web at runtime. If a user request cannot be satisfied by a single Web service, it is (or should be) possible to combine existing services in order to fulfill the request. However, there are several challenging issues that need to be addressed before this can be realized in the true sense. One of them is the ability to ensure end-to-end QoS of a Web service composition. There is a need for a SLA negotiation system which can ensure the autonomous QoS negotiation of Web service compositions irrespective of the application domain. In this paper we propose agent-based coordinated-negotiation architecture to ensure collective functionality, end-to-end QoS and the stateful coordination of complex services. We describe a prototype implementation to demonstrate how this architecture can be used in different application domains. We have also demonstrated how the negotiation system on the service provider's side can be implemented both as an agent based negotiation system and as a Web service based negotiation system.*

4. Nicole Oldham, Kunal Verma, Amit Sheth, Farshad Hakimpour. *Semantic WS-agreement partner selection.* WWW '06: Proceedings of the 15th international conference on World Wide Web, pages 697–706, ACM, 2006.

   Abstract: *In a dynamic service oriented environment it is desirable for service consumers and providers to offer and obtain guarantees regarding their capabilities and requirements. WS-Agreement defines a language and protocol for establishing agreements between two parties. The agreements are complex and expressive to the extent that the manual matching of these agreements would be expensive both in time and resources. It is essential to develop a method for matching agreements automatically. This work presents the framework and implementation of an innovative tool for the matching providers and consumers based on WS-Agreements. The approach utilizes Semantic Web technologies to achieve rich and accurate matches. A key feature is the novel and flexible approach for achieving user personalized matches.*

5. Rocco De Nicola, Gianluigi Ferrari, Ugo Montanari, Rosario Pugliese, Emilio Tuosto. *A Basic Calculus for Modelling Service Level Agreements.* International Conference on Co-

ordination Models and Languages, LNCS, Vol. 3454, pages 33–48, Springer Verlag, April 2005.

Abstract: *The definition of suitable abstractions and models for identifying, understanding and managing Quality of Service (QoS) constraints is a challenging issue of the Service Oriented Computing paradigm. In this paper we introduce a process calculus where QoS attributes are first class objects. We identify a minimal set of primitives that allow capturing in an abstract way the ability to control and coordinate services in presence of QoS constraints.*

6. James Skene, D. Davide Lamanna, Wolfgang Emmerich. *Precise Service Level Agreements.* ICSE, pages 179–188, 2004.

   Abstract: *SLAng is an XML language for defining service level agreements, the part of a contract between the client and provider of an Internet service that describes the quality attributes that the service is required to possess. We define the semantics of SLAng precisely by modelling the syntax of the language in UML, then relating the language model to a model that describes the structure and behaviour of services. The presence of SLAng elements imposes behavioural constraints on service elements, and the precise definition of these constraints using OCL constitutes the semantic description of the language. We use the semantics to define a notion of SLA compatibility, and an extension to UML that enables the modelling of service situations as a precursor to analysis, implementation and provisioning activities.*

Technical Reports and Manuals:

1. Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, Ming Xu. *Web Services Agreement Specification (WS-Agreement).* Grid Resource Allocation Agreement Protocol, September 2005.

   Abstract: *This document describes Web Services Agreement Specification (WS-Agreement), a Web Services protocol for establishing agreement between two parties, such as between a service provider and consumer, using an extensible XML language for specifying the nature of the agreement, and agreement templates to facilitate discovery of compatible agreement parties. The specification consists of three parts which may be used in a composable manner: a schema for specifying an agreement, a schema for specifying an agreement template, and a set of port types and operations for managing agreement life-cycle, including creation, expiration, and monitoring of agreement states.*

2. Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, Richard Franck . *Web Service Level Agreement (WSLA) Language Specification.* IBM Corporation, 2003.

   Abstract: *This document describes the specification language for service level agreements for Web Services, the Web Service Level Agreement (WSLA) language. WSLAs are agreements between a service provider and a customer and as such define the obligations of the parties involved. Primarily, this is the obligation of a service provider to perform a service according to agreed-upon guarantees for IT-level service parameters (such as availability, response time and throughput) for Web Services.*

   An SLA also specifies the measures to be taken in case of deviation and failure to meet the asserted service guarantees, for example, a notification of the service customer. The assertions of the service provider are based on a detailed definition of service parameters including the algorithms – how basic metrics should be measured in systems and how they are aggregated into composite metrics and SLA parameters. In addition, a WSLA can express the operations of monitoring and managing the service. This may include third parties (such as Management Service Providers) that contribute to the measurement of metrics, supervision of guarantees or even the management of deviations of service guarantees. These multi-party constellations necessitate the definition of the interactions among the parties supervising the WSLA.

   However, a WSLA only covers the agreed common view of a service between the parties involved. To actually act as a participant in a WSLA, parties have various degrees of freedom to define an implementation policy for a service and its supervision. Typically, the obligations of a WSLA must be translated into system-level configuration information, which can be proprietary to each party involved.

3. Akhil Sahai, Anna Durante, Vijay Machiraju. *Towards Automated SLA Management for Web Services.* Num. HPL-2001-310, HP Laboratories, July 2002.

Abstract: *In order to automate SLA management it is essential to specify SLAs in precise and unambiguous manner as well as keep the specification flexible. While precision will help automate the process of monitoring and metric collection, flexibility will enable extending it to unforeseen service level agreement specifications.*

18 Publications in **Service Aggregation and Composition**

Articles in Refereed Journals:

1. Ardagna, D., Pernici, B.. *Adaptive Service Composition in Flexible Processes.* Software Engineering, IEEE Transactions on, Vol. 33, Num. 6, pages 369–384, June 2007.

   Abstract: *In advanced service oriented systems, complex applications, described as abstract business processes, can be executed by invoking a number of available Web services. End users can specify different preferences and constraints and service selection can be performed dynamically identifying the best set of services available at runtime. In this paper, we introduce a new modeling approach to the Web service selection problem that is particularly effective for large processes and when QoS constraints are severe. In the model, the Web service selection problem is formalized as a mixed integer linear programming problem, loops peeling is adopted in the optimization, and constraints posed by stateful Web services are considered. Moreover, negotiation techniques are exploited to identify a feasible solution of the problem, if one does not exist. Experimental results compare our method with other solutions proposed in the literature and demonstrate the effectiveness of our approach toward the identification of an optimal solution to the QoS constrained Web service selection problem*

2. Roberto Lucchi, Manuel Mazzara. *A pi-calculus based semantics for WS-BPEL .* Journal of Logic and Algebraic Programming, Vol. 70, Num. 1, pages 96–118, January 2007.

   Abstract: *Recently, the term Web services orchestration has been introduced to address some issues related to Web services composition, that is the way of defining a complex service out of simpler ones. Several proposals for describing orchestration for business processes have been presented in the last years and many of these languages make use of concepts as long-running transactions and compensations for coping with error handling. WS-BPEL 2.0, the most credited candidate for becoming a standard, provides three different mechanisms allowing to cope with abnormal situations: exception, event and compensation handling. This complexity makes it difficult to formally define the framework, thus limiting the formal reasoning about the designed applications. In this paper we advocate that three different mechanisms for error handling are not necessary and we formalize a novel orchestration language based on the idea of event notification as the unique error handling mechanism. To this end, we formally define the three BPEL mechanisms in terms of our calculus. It is possible to take advantages of this formal description in two ways. Firstly, this language represents by itself a proposal of simplification for WS-BPEL 2.0 including an unambiguous specification. Secondly, an implementor of an actual WS-BPEL 2.0 orchestration engine could implement simply this single mechanism providing all the remaining ones by compilation. With this attempt we intend to give a concrete contribute towards the improvement of the quality of the BPEL specification, the applicability of BPEL itself and the implementation of real orchestration engines. Finally, as a case study we consider some of the hundreds of open issues met by the WS-BPEL designers and we propose a solution making use of the experience gained developing our algebra.*

3. Groenmo, R., Jaeger, M.C.. *Model-driven semantic Web service composition.* Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific, To Appear, December 2005.

   Abstract: *As the number of available Web services increases there is a growing demand to realise complex business processes by combining and reusing available Web services. The reuse and combination of services results in a composition of Web services that may also involve services provided in the Internet. With semantically described Web services, an automated matchmaking of capabilities can help identify suitable services. To address the need for semantically defined Web services, OWL-S and WSML have been proposed as competing semantic Web service languages. We show how the proposed semantic Web service languages can be utilized within a model-driven methodology for building composite Web services. In addition we combine the semantic-based discovery with the support for processing QoS requirements to apply a ranking or a selection of the candidates. The methodology describes a process which guides the developer through four phases, starting with the initial modelling, and ending with a new composite service that can be deployed and published to be consumed by other users.*

4. Schahram Dustdar, Wolfgang Schreiner. *A survey on web services composition.* IJWGS, Vol. 1, Num. 1, pages 1–30, 2005.

   Abstract: *Due to the web services' heterogeneous nature, which stems from the definition of several XML-based standards to overcome platform and language dependence, web services have become an emerging and promising technology to design and build complex inter-enterprise business applications out of single web-based software components. To establish the existence of a global component market, in order to enforce extensive software reuse, service composition experienced increasing interest in doing a lot of research effort. This paper discusses the urgent need for service composition, the required technologies to perform service composition. It also presents several different composition strategies, based on some currently existing composition platforms and frameworks, re-presenting first implementations of state-of-the-art technologies, and gives an outlook to essential future research work.*

5. Xiulan Yu, Long Zhang, Ying Li, Ying Chen. *WSCE: a flexible Web Service Composition Environment.* Web Services, 2004. Proceedings. IEEE International Conference on, To Appear, July 2004.

   Abstract: *In this paper, we propose the concepts of virtual partner and inspector into the Web services composition. Virtual partner, as an IT level concept, is a Web service (pseudo Web service) using the same interface with the actual partner but different binding message. A virtual partner can be invoked directly by a business process described by BPEL, so that the BPEL programmer can test both application's functionality and non functionality performance early in the development cycle to avoid any problems in the final runtime, or test the selection of their partners in business level design. The IT virtual partners provide developers with a range of the techniques which let them explore every aspect of their program. Inspector is proposed when using the third-party process engine. An inspector itself is also a Web service. The programmer can register any required output information in it. The IT virtual partner and the inspector concepts have been integrated in our WSCE, a flexible Web Services Composition Environment for a business process. WSCE is a prototype of autonomic modeling and simulation environment. With the help of a third-party BPEL engine, it provides programmer with concepts and tools to facilitate business process programming.*

6. Liangzhao Zeng, Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.. *QoS-aware middleware for Web services composition.* Software Engineering, IEEE Transactions on, Vol. 30, Num. 5, pages 311–327, May 2004.

   Abstract: *The paradigmatic shift from a Web of manual interactions to a Web of programmatic interactions driven by Web services is creating unprecedented opportunities for the formation of online business-to-business (B2B) collaborations. In particular, the creation of value-added services by composition of existing ones is gaining a significant momentum. Since many available Web services provide overlapping or identical functionality, albeit with different quality of service (QoS), a choice needs to be made to determine which services are to participate in a given composite service. This paper presents a middleware platform which addresses the issue of selecting Web services for the purpose of their composition in a way that maximizes user satisfaction expressed as utility functions over QoS attributes, while satisfying the constraints set by the user and by the structure of the composite service. Two selection approaches are described and compared: one based on local (task-level) selection of services and the other based on global allocation of tasks to services using integer programming.*

7. Mariya Koshkina, Franck van Breugel. *Modelling and verifying web service orchestration by means of the concurrency workbench.* SIGSOFT Softw. Eng. Notes, Vol. 29, Num. 5, pages 1–10, ACM, 2004.

   Abstract: *Verification techniques like model checking, preorder checking and equivalence checking are shown to be relevant to web service orchestration. The Concurrency Workbench of the New Century (CWB) is a verification tool that supports these verification techniques. By means of the Process Algebra Compiler (PAC), the CWB is modified to support the BPE-calculus. The BPE-calculus is a small language, based on BPEL4WS, to express web service orchestration. Both the syntax and the semantics of the BPE-calculus are formally defined. These are subsequently used as input for the PAC. As output, the PAC produces modules that are incorporated into the CWB so that it supports the BPE-calculus and, hence, provides a verification tool for web service orchestration.*

8. Milanovic, N., Malek, M.. *Current solutions for Web service composition.* Internet Computing, IEEE, Vol. 8, Num. 6, pages 51–59, November/December 2004.

   Abstract: *Web service composition lets developers create applications on top of service-oriented computing's native description, discovery, and communication capabilities. Such applications are rapidly deployable and offer developers reuse possibilities and users seamless access to a variety of complex services. There are many existing approaches to service composition, ranging from abstract methods to those aiming to be industry standards. The authors describe four key issues for Web service composition.*

9. W. van der Aalst. *Don't go with the flow: Web services composition standards exposed.* IEEE Intelligent Systems, To Appear, Jan/Feb 2003.

   Abstract: *The recently released Business Process Execution Language for Web Services (BPEL4WS) is said to combine the best of other standards for web services composition such as WSFL from IBM and XLANG of Microsoft. BPEL4WS allows for a mixture of block structured and graph structured process models thus making the language expressive at the price of being complex. Although BPEL4WS is not such a bad proposal by itself, it is remarkable how much attention this standard receives while the more fundamental issues and problems such as semantics, expressiveness, and adequacy do not get the attention they deserve. Having a standard is a very good idea. However, there are too many of them and most of them die before becoming mature. A simple indicator of this development is the increasing length of acronyms: PDL, XPDL, BPSS, EDOC, BPML, WSDL, WSCI, ebXML, and BPEL4WS are just some of the acronyms referring to various standards in the domain. Another problem is that these languages typically have no clearly defined semantics. The only way to overcome these problems is to critically evaluate the so-called standards for web services composition, i.e., Don't go with the flow!*

Articles in Refereed Conferences:

1. Gero Decker, Oliver Kopp, Frank Leymann, Mathias Weske. *BPEL4Chor: Extending BPEL for Modeling Choreographies.* ICWS, pages 296–303, 2007.

   Abstract: *The Business Process Execution Language (BPEL) is a language to orchestrate web services into a single business process. In a choreography view, several processes are interconnected and their interaction behavior is described from a global perspective. This paper shows how BPEL can be extended for defining choreographies. The proposed extensions (BPEL4Chor) distinguish between three aspects: (i) participant behavior descriptions, i.e. control flow dependencies in each participant, (ii) the participant topology, i.e. the existing participants and their interconnection using message links and (iii) participant groundings, i.e. concrete configurations for data formats and port types. As BPEL itself is used unchanged, the extensions facilitate a seamless integration between service choreographies and orchestrations. The suitability of the extensions is validated by assessing their support for the Service Interaction Patterns.*

2. Anca Muscholl, Igor Walukiewicz. *A Lower Bound on Web Services Composition.* FoSSaCS, pages 274–286, 2007.

   Abstract: *A web service is modeled here as a finite state machine. A composition problem for web services is to decide if a given web service can be constructed from a given set of web services; where the construction is understood as a simulation of the specification by a fully asynchronous product of the given services. We show an EXPTIME-lower bound for this problem, thus matching the known upper bound. Our result also applies to richer models of web services, such as the Roman model.*

3. Yan-ping Chen, Zeng-zhi Li, Qin-xue Jin, Chuang Wang. *Study on QoS Driven Web Services Composition.* Frontiers of WWW Research and Development - APWeb 2006, Lecture Notes on Computer Science, Vol. 3841, pages 702–707, Springer Verlag, 2006.

   Abstract: *Providing composed Web Services based on the QoS requirements of clients is still an urgent problem to be solved. In this paper, we try to solve this problem. Firstly, we enhanced the current WSDL to describe the QoS of services, and then gave a way to choose the proper pre-exist services based on their QoS.*

4. Howard Foster, Sebastian Uchitel, Jeff Magee, Jeff Kramer. *LTSA-WS: a tool for model-based verification of web service compositions and choreography.* ICSE '06: Proceedings of the 28th international conference on Software engineering, pages 771–774, ACM, 2006.

   Abstract: *In this paper we describe a tool for a model-based approach to verifying compositions of web service implementations. The tool supports verification of properties created from design specifications and implementation models to confirm expected results from the viewpoints of both the designer and implementer. Scenarios are modeled in UML, in the form of Message Sequence Charts (MSCs), and then compiled into the Finite State Process (FSP) process algebra to concisely model the required behavior. BPEL4WS implementations are mechanically translated to FSP to allow an equivalence trace verification process to be performed. By providing early design verification and validation, the implementation, testing and deployment of web service compositions can be eased through the understanding of the behavior exhibited by the composition. The approach is implemented as a plug-in for the Eclipse development environment providing cooperating tools for specification, formal modeling, verification and validation of the composition process.*

5. Mohsen Rouached, Olivier Perrin, Claude Godart. *Towards Formal Verification of Web Service Composition.* Business Process Management, pages 257–273, 2006.

   Abstract: *Web services composition is an emerging paradigm for enabling application integration within and across organizational boundaries. Current Web services composition proposals, such as BPML, WS-BPEL, WSCI, and OWL-S, provide solutions for describing the control and data flows in Web service composition. However, such proposals remain at the descriptive level, without providing any kind of mechanisms or tool support for analysis and verification. Therefore, there is a growing interest for the verification techniques which enable designers to test and repair design errors even before actual running of the service, or allow designers to detect erroneous properties and formally verify whether the service process design does have certain desired properties. In this paper, we propose to verify Web services composition using an event driven approach. We assume Web services that are coordinated by a composition process expressed in WSBPEL and we use Event Calculus to specify the properties and requirements to be monitored.*

6. Aysu Betin-Can, Tevfik Bultan, Xiang Fu. *Design for verification for asynchronously communicating Web services.* WWW, pages 750–759, 2005.

   Abstract: *We present a design for verification approach to developing reliable web services. We focus on composite web services which consist of asynchronously communicating peers. Our goal is to automatically verify properties of interactions among such peers. We propose a design pattern that eases the development of such web services and enables a modular, assume-guarantee style verification strategy. In the proposed design pattern, each peer is associated with a behavioral interface description which specifies how that peer will interact with other peers. Using these peer interfaces we automatically generate BPEL specifications to publish for interoperability. Assuming that the participating peers behave according to their interfaces, we verify safety and liveness properties about the global behavior of the composite web service during behavior verification. During interface verification, we check that each peer implementation conforms to its interface. Using the modularity in the proposed design pattern, we are able to perform the interface verification of each peer and the behavior verification as separate steps. Our experiments show that, using this modular approach, one can automatically and efficiently verify web service implementations.*

7. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani. *An approach for QoS-aware service composition based on genetic algorithms.* GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, pages 1069–1075, ACM, 2005.

   Abstract: *Web services are rapidly changing the landscape of software engineering. One of the most interesting challenges introduced by web services is represented by Quality Of Service (QoS)–aware composition and late–binding. This allows to bind, at run–time, a service–oriented system with a set of services that, among those providing the required features, meet some non–functional constraints, and optimize criteria such as the overall cost or response time. In other words, QoS–aware composition can be modeled as an optimization problem.We propose to adopt Genetic Algorithms to this aim. Genetic Algorithms, while being slower than integer programming, represent a more scalable choice, and are more suitable to handle generic QoS attributes. The paper describes our approach and its applicability, advantages and weaknesses, discussing results of some numerical simulations.*

8. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani. *QoS-Aware Replanning of Composite Web Services.* ICWS '05: Proceedings of the IEEE International Conference on Web Services, pages 121–129, IEEE Computer Society, 2005.

   Abstract: *Run-time service discovery and late-binding constitute some ofthe most challenging issues of service-oriented software engineering.For late-binding to be effective in the case of composite services,a QoS-aware composition mechanism is needed. This meansdetermining the set of services that, once composed, not only willperform the required functionality, but also will best contribute toachieve the level of QoS promised in Service Level Agreements (SLAs). However, QoS-aware composition relies on estimated QoS values and workflow execution paths previously obtained using a monitoring mechanism. At run-time, the actual QoS valuesmay deviate from the estimations, or the execution path may not bethe one foreseen. These changes could increase the risk of breakingSLAs and obtaining a poor QoS. Such a risk could be avoided byreplanning the service bindings of the workflow slice still to be executed. This paper proposes an approach to trigger and perform compositeservice replanning during execution. An evaluation has been performedsimulating execution and replanning on a set of composite serviceworkflows.*

9. Casey K. Fung, Patrick C. K. Hung, Guijun Wang, Richard C. Linger, Gwendolyn H. Walton. *A Study of Service Composition with QoS Management.* ICWS, pages 717–724, 2005.

   Abstract: *Quality of service (QoS) management in compositions of services requires careful consideration of QoS characteristics of the services and effective QoS management in their execution. A Web service is a software system that supports interoperable application-to-application interaction over the Internet. Web services are based on a set of XML standards such as simple object access protocol (SOAP). The interactions of SOAP messages between Web services form the theoretical model of SOAP message exchange patterns (MEP). Web Services Business Process Execution Language (WSBPEL) defines an interoperable integration model that facilitates automated process integration in intra- and inter-corporate environments. A service-level agreement (SLA) is a formal contract between a Web services requestor and provider guaranteeing quantifiable issues at defined levels only through mutual concessions. Based on a prior research work on message detail record (MDR), this paper further proposes a SOAP message tracking model for supporting QoS end-to-end management in the context of WSBPEL and SLA. This paper motivates the study of QoS management in a Web service composition framework with the evolution of a distributed toolkit in an industrial setting.*

10. Hai Huang, Wei-Tek Tsai, Raymond Paul, Yinong Chen. *Automated Model Checking and Testing for Composite Web Services..* 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2005), pages 300–307, IEEE Computer Society, 2005.

    Abstract: *Web Services form a new distributed computing paradigm. Collaborative verification and validation are important when Web Services from differen vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web Services by model checking the proces model of OWL-S (Web Ontology Language for Web Services) and to validate them by the test cases automatically generated in the model checking process. We extend the BLAST, a model checker tha handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

11. Michael C. Jaeger, Gero Mühl, Sebastian Golze. *QoS-Aware Composition of Web Services: A Look at Selection Algorithms.* ICWS, pages 807–808, 2005.

    Abstract: *When a composition of Web services is designed, available services are put together to form a defined flow of executions. In a discovery process, a trader proposes available Web services as potential candidates. In a succeeding selection, for each task a trader chooses one candidate to form the optimal composition due to selection criteria. This paper discusses how the selection can consider different quality-of-service (QoS) categories to determine the most suitable candidates for the composition. If more than one category is used for optimisation, a multi-dimensional optimisation problem arises. This mentions similarities to similar combinatorial problems. Then, possible solutions are proposed and their performance is evaluated.*

12. Rohit Aggarwal, Kunal Verma, John A. Miller, William Milnor. *Constraint Driven Web Service Composition in METEOR-S.* IEEE SCC, pages 23–30, 2004.

    Abstract: *Creating Web processes using Web service technology gives us the opportunity for selecting new services which best suit our need at the moment. Doing this automatically would require us to quantify our criteria for selection. In addition, there are challenging issues of correctness and optimality. We present a Constraint Driven Web Service Composition tool in METEOR-S, which allows the process designers to bind Web Services to an abstract process, based on business and process constraints and generate an executable process. Our approach is to reduce much of the service composition problem to a constraint satisfaction problem. It uses a multi-phase approach for constraint analysis. This work was done as part of the METEOR-S framework, which aims to support the complete lifecycle of semantic Web processes.*

13. Jinghai Rao, Xiaomeng Su. *A Survey of Automated Web Service Composition Methods.* SWSWPC, pages 43–54, 2004.

    Abstract: *In todays Web, Web services are created and updated on the fly. Its already beyond the human ability to analysis them and generate the composition plan manually. A number of approaches have been proposed to tackle that problem. Most of them are inspired by the researches in cross-enterprise workflow and AI planning. This paper gives an overview of recent research efforts of automatic Web service composition both from the workflow and AI planning research community.*

14. Howard Foster, Sebastián Uchitel, Jeff Magee, Jeff Kramer. *Model-based Verification of Web Service Compositions..* 18th IEEE International Conference on Automated Software Engineering (ASE 2003) , pages 152–163, IEEE Computer Society, 2003.

    Abstract: *In this paper we discuss a model-based approach to verifying web service compositions for web service implementations. The approach supports verification against specification models and assigns semantics to the behavior of implementation models so as to confirm expected results for both the designer and implementer. Specifications of the design are modeled in UML, in the form of Message Sequence Charts (MSCs), and mechanically compiled into the Finite State Process notation (FSP) to concisely describe and reason about the concurrent programs. Implementations are mechanically translated to FSP to allow a trace equivalence verification process to be performed. By providing early design verification, the implementation, testing and deployment of web service compositions can be eased through the understanding of the differences, limitations and undesirable traces allowed by the composition. The approach is supported by a suite of cooperating tools for specification, formal modeling and trace animation of the composition workflow.*

15. Rania Khalaf, Frank Leymann. *On Web Services Aggregation.* TES, pages 1–13, 2003.

    Abstract: *The Web services framework is enabling applications from different providers to be offered as services that can be used and composed in a loosely-coupled manner. Subsequently, the aggregation of services to form composite applications and maximize reuse is key. While choreography has received the most attention, services often need to be aggregated in a much less constrained manner. As a number of different mechanisms emerge to create these aggregations, their relation to each other and to prior work is useful when deciding how to create an aggregation, as well as in extending the models themselves and proposing new ones. In this paper, we discuss Web services aggregation by presenting a first-step classification based on the approaches taken by the different proposed aggregation techniques. Finally, a number of models are presented that are created from combinations of the above.*

16. Srini Narayanan, Sheila A. McIlraith. *Simulation, verification and automated composition of web services.* WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 77–88, ACM, 2002.

    Abstract: *Web services – Web-accessible programs and devices - are a key application area for the Semantic Web. With the proliferation of Web services and the evolution towards the Semantic Web comes the opportunity to automate various Web services tasks. Our objective is to enable markup and automated reasoning technology to describe, simulate, compose, test, and verify compositions of Web services. We take as our starting point the DAML-S DAML+OIL ontology for describing the capabilities of Web services. We define the semantics for a relevant subset of DAML-S in terms of a first-order logical language. With the semantics in hand, we encode our service descriptions in a Petri Net formalism and provide decision* *procedures for Web service simulation, verification and composition. We also provide an analysis of the complexity of these tasks under different restrictions to the DAML-S composite services we can describe. Finally, we present an implementation of our analysis techniques. This implementation takes as input a DAML-S description of a Web service, automatically generates a Petri Net and performs the desired analysis. Such a tool has broad applicability both as a back end to existing manual Web service composition tools, and as a stand-alone tool for Web service developers.*

Books and Monographs:

1. J. Cardoso. *Quality of Service and Semantic Composition of Workflows.* Ph.D. Thesis, University of Georgia, 2002.

   Abstract: *Workflow management systems (WfMSs) have been used to support a variety of business processes. As organizations adopt new working models, such as e-commerce, new challenges arise for workflow systems. These challenges include support for the adequate management of quality of service (QoS) and the development of new solutions to facilitate the composition of workflow applications involving Web services. The good management of QoS directly impacts the success of organizations participating in e- commerce activities by better fulfilling customer expectations and achieving customer satisfaction. To enable adequate QoS management, research is required to develop mechanisms that specify, compute, monitor, and control the QoS of the products or services to be delivered. The composition of workflows to model e-service applications differs from the design of traditional workflows due to the number of Web services available during the composition process and to their heterogeneity. Two main problems need to be solved: how to efficiently discover Web services and how to facilitate their interoperability. To enhance WfMSs with QoS management, we have developed a QoS model that allows for the description of nonfunctional aspects of workflow components, from a quality of service perspective. To automatically compute the overall QoS of a workflow, we have developed a mathematical model and implemented an algorithm (SWR algorithm). Our QoS model and mathematical model have been validated with the deployment and execution of a set of production workflows in the area of genetics. The analysis of the collected data proves that our models provide a suitable framework for estimating, predicting, and analyzing the QoS of production workflows. To support, facilitate, and assist the composition of workflows involving Web services, we present a solution based on ontologies. We have developed an algorithm that workflow systems and discovery mechanisms can use to find Web services with desired interfaces and operational metrics, and to assist designers in resolving heterogeneity issues among Web services. Our approach provides an important solution to enhance Web service discovery and interoperability.*

19 Publications in **Service Architecture**

Articles in Refereed Journals:

1. Papazoglou, M.P., van den Heuvel, W.-J.. *Web services management: a survey.* Internet Computing, IEEE, Vol. 9, Num. 6, pages 58–64, November/December 2005.

   Abstract: *Solutions based on service-oriented architectures are promising in that they leverage common services and enable collaborative business processes that cross organizational boundaries. However, because Web services applications can span multiple hosts, operating systems, languages, and enterprises, it's problematic to measure, control, and manage application availability and performance. In addition to discussing the relationship of Web services management to traditional distributed systems management, this survey explores various Web services management approaches and their underlying architectural concepts.*

2. James Pasley. *How BPEL and SOA Are Changing Web Services Development.* IEEE Internet Computing, Vol. 9, Num. 3, pages 60–67, 2005.

   Abstract: *As the use of Web services grows, organizations are increasingly choosing the Business Process Execution Language for modeling business processes within the Web services architecture. In addition to orchestrating organizations' Web services, BPEL's strengths include asynchronous message handling, reliability, and recovery. By developing Web services with BPEL in mind, organizations can implement aspects of the service-oriented architecture that might previously have been difficult to achieve.*

3. Frank E. Ritter, Richard M. Young. *Embodied models as simulated users: introduction to this special issue on using cognitive models to improve interface design.* Int. J. Hum.-Comput. Stud., Vol. 55, Num. 1, pages 1–14, 2001.

   Abstract: *Cognitive models provide a means for applying what is known from psychology to the design of interfaces, thereby improving their quality and usability. Existing uses of models include predicting time and errors for users to perform tasks, acting as embedded assistants to help users perform their tasks, and serving as surrogate users. Treating the design of human-computer interfaces as a form of engineering design requires the development and application of user models. A recent trend is for models to be built within the fixed framework of a cognitive architecture, which has been extended by the addition of simulated eyes and hands, enabling the construction of embodied models. Being embodied allows models to interact directly with interfaces. The resulting models can be used to evaluate the interfaces they use, and serve as explanations of users' behavior. The papers in this Special Issue point to a new route for the future, one in which models built within embodied cognitive architectures provide information for the design of better interfaces.*

Articles in Refereed Conferences:

1. Dimka Karastoyanova, Branimir Wetzstein, Tammo van Lessen, Daniel Wutke, Jörg Nitzsche, Frank Leymann. *Semantic Service Bus: Architecture and Implementation of a Next Generation Middleware.* ICDE Workshops, pages 347–354, 2007.

   Abstract: *In this paper we present a middleware for the service oriented architecture, called the Semantic Service Bus. It is an advanced middleware possessing enhanced features, as compared to the conventional service buses. It is distinguished by the fact that it uses semantic description of service capabilities, and requirements towards services to enable more elaborate service discovery, selection, routing, composition and data mediation. The contributions of the paper are the conceptual architecture of the Semantic Service Bus and a prototypical implementation supporting different semantic Web service technologies (OWL-S and WSMO) and conventional Web services. Since mission critical application scenarios (for SOA) involve complex orchestrations of services, we have chosen to utilize semantically annotated service orchestrations as the applications to employ this middleware.*

Technical Reports and Manuals:

1. I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich . *The Open Grid Services Architecture, Version 1.0.* ANL, IBM, Fujitsu, NeSC, CCLRC-RAL, UVa, Hitachi, Intel, HP, 2005.

   Abstract: *Successful realization of the Open Grid Services Architecture (OGSA) vision of a broadly applicable and adopted framework for distributed system integration, virtualization, and management requires the definition of a core set of interfaces, behaviors, resource models, and bindings. This document, produced by the OGSA working group within the Global Grid Forum (GGF), provides a first version of this OGSA definition. The document focuses on requirements and the scope of important capabilities required to support Grid systems and applications in both e-science and e-business. The capabilities described are Execution Management, Data, Resource Management, Security, Self-Management, and Information. The description of the capabilities is at a high-level and includes, to some extent, the interrelationships between the capabilities.*

20 Publications in **Service Design**

Articles in Refereed Journals:

1. Mike P. Papazoglou, Willem-Jan van den Heuvel. *Service-oriented design and development methodology.* Int. J. Web Eng. Technol., Vol. 2, Num. 4, pages 412–442, 2006.

   Abstract: *Service Oriented Architectures (SOA) are rapidly emerging as the premier integration and architectural approach in contemporary, complex, heterogeneous computing environments. SOA is not simply about deploying software: it also requires that organisations evaluate their business models, come up with service-oriented analysis and design techniques, deployment and support plans, and carefully evaluate partner/customer/supplier relationships. Since SOA is based on open standards and is frequently realised using Web Services (WS), developing meaningful WS and business process specifications is an important requirement for SOA applications that leverage WS. Designers and developers cannot be expected to oversee a complex service-oriented development project without relying on a sound design and development methodology. This paper provides an overview of the methods and techniques used in service-oriented design and development. The aim of this paper is to examine a service development methodology from the point of view of both service producers and requesters and review the range of elements in this methodology that are available to them.*

Articles in Refereed Conferences:

1. Aysu Betin-Can, Tevfik Bultan, Xiang Fu. *Design for verification for asynchronously communicating Web services.* WWW, pages 750–759, 2005.

   Abstract: *We present a design for verification approach to developing reliable web services. We focus on composite web services which consist of asynchronously communicating peers. Our goal is to automatically verify properties of interactions among such peers. We propose a design pattern that eases the development of such web services and enables a modular, assume-guarantee style verification strategy. In the proposed design pattern, each peer is associated with a behavioral interface description which specifies how that peer will interact with other peers. Using these peer interfaces we automatically generate BPEL specifications to publish for interoperability. Assuming that the participating peers behave according to their interfaces, we verify safety and liveness properties about the global behavior of the composite web service during behavior verification. During interface verification, we check that each peer implementation conforms to its interface. Using the modularity in the proposed design pattern, we are able to perform the interface verification of each peer and the behavior verification as separate steps. Our experiments show that, using this modular approach, one can automatically and efficiently verify web service implementations.*

21 Publications in **Service Discovery**

Articles in Refereed Journals:

1. Tao Yu, Kwei-Jay Lin. *Service selection algorithms for Web services with end-to-end QoS constraints.* Inf. Syst. E-Business Management, Vol. 3, Num. 2, pages 103–126, 2005.

   Abstract: *Web services are new forms of Internet software that can be universally deployed and invoked using standard protocols. Services from different providers can be integrated into a composite service regardless of their locations, platforms, and/or execution speeds to implement complex business processes and transactions. In this paper, we study the end-to-end QoS issues of composite services by utilizing a QoS broker that is responsible for selecting and coordinating the individual service component. We design the service selection algorithms used by QoS brokers to construct the optimal composite service. The objective of the algorithms is to maximize the user-defined utility function value while meeting the end-to-end delay constraint. We propose two solution approaches to the service selection problem: the combinatorial approach, by modeling the problem as the Multiple Choice Knapsack Problem (MCKP), and the graph approach, by modeling the problem as the constrained shortest path problem in the graph theory. We study efficient solutions for each approach.*

2. Shuping Ran. *A model for web services discovery with QoS.* SIGecom Exch., Vol. 4, Num. 1, pages 1–10, ACM, 2003.

   Abstract: *Web services technology has generated a lot interest, but its adoption rate has been slow. This paper discusses issues related to this slow take up and argues that quality of services is one of the contributing factors. The paper proposes a new Web services discovery model in which the functional and non-functional requirements (i.e. quality of services) are taken into account for the service discovery. The proposed model should give Web services consumers some confidence about the quality of service of the discovered Web services.*

Articles in Refereed Conferences:

1. Kyriakos Kritikos, Dimitris Plexousakis. *Evaluation of QoS-Based Web Service Matchmaking Algorithms.* International Conference on Services Computing, 2008.

   Abstract: *Web Service (WS) discovery is a prerequisite for achieving WS composition and orchestration. Although a lot of esearch has been conducted on the functional discovery of WSs, the proposed techniques fall short when faced with the foreseen increase in the number of (potentially functionally- equivalent) WSs. The above situation can be resolved with the addition of non-functional (Quality of Service (QoS)) discovery mechanisms to WS discovery engines. QoS-based WS matchmaking algorithms have been devised for this reason. However, they are either slow - as they are based on ontology reasoners - or produce inaccurate results. Inaccuracy is caused both by the syntactic matching of QoS concepts and by wrong matchmaking metrics. In this paper, we present two Constraint Programming (CP) QoS-based WS discovery algorithms for unary constrained WS specifications that produce accurate results with good performance. We also evaluate these algorithms on matchmaking time, precision and recall in different settings in order to demonstrate their efficiency and accuracy.*

2. Kyriakos Kritikos, Dimitris Plexousakis. *OWL-Q for Semantic QoS-based Web Service Description and Discovery.* First International Joint Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, November 2007.

   Abstract: *Semantic Web Services are emerging for their promise to produce a more accurate and precise Web Service discovery process. However, most of research approaches focus only on the functional part of semantic Web Service description. The above fact along with the proliferation of Web Services is highly probable to lead to a situation where Web Service registries will return many functionally-equivalent Web Service advertisements for each user request. This problem can be solved with the semantic description of QoS for Web Services. QoS is a set of non-functional properties encompassing performance and network-related characteristics of resources. So it can be used for distinguishing between functionally-equivalent Web*

Services. Current research approaches for QoS-based Web Service description are either syntactic or poor or non-extensible. To solve this problem, we have developed a rich and extensible ontological specification called OWL-Q for semantic QoS-based Web Service description. We analyze all OWL-Q parts and reason that rules should be added in order to support property inferencing and constraint enforcement. Finally, we line out our under-development semantic framework for QoS-based Web Service description and discovery.

3. Ester Giallonardo, Eugenio Zimeo. *More Semantics in QoS Matching.* SOCA, pages 163–171, 2007.

   Abstract: *The evolution of the Web towards a global computing environment is promoting new research efforts aimed at the formal characterization of Web Services QoS. Reasoning on QoS is a key to improve matching process during the discovery of desired services and a step towards the transformation of applications in collections of loosely coupled services virtually connected by semantic similarities. The paper presents the on QoS ontology, an openly available OWL ontology for QoS, and evaluates it in a QoS-aware matching environment. The ontology can be used to express functions of QoS metrics useful to improve the recall tied to the matching of a template request with target Web Services. To this end, the ontology introduces the concept of derivation in the matching process. This gives the possibility of matching a QoS template with published Web Services by deriving different QoS parameters when a one-to-one matching fails. The proposed matching algorithm utilizes a reasoner that exploits the ontology to avoid apparent mismatches. An experimental evaluation shows that exploiting QoS knowledge significantly improves matching recall without deteriorating precision.*

4. Kyriakos Kritikos, Dimitris Plexousakis. *Requirements for QoS-based Web Service Description and Discovery.* COMPSAC (2), pages 467–472, 2007.

   Abstract: *The goal of Service Oriented Architectures is to enable the creation of business applications through the automatic discovery and composition of independently developed and deployed (Web) services. Automatic discovery of Web Services (WSs) can be achieved by incorporating semantics into a richer WS description model (WSDM) and the use of Semantic Web (SW) technologies in the WS matchmaking and selection models. A sufficiently rich WSDM should encompass not only functional but also non-functional aspects like Quality of Service (QoS). QoS is a set of performance attributes that has a substantial impact on WS requesters' expectations. Thus, it can be used as a discriminating factor of functionally-equivalent WSs. The focus of this paper is twofold: to analyze the requirements of a semantically rich QoS-based WSDM and to provide SW and constrainedbased mechanisms for enriching syntactic QoS-based WS Discovery (WSDi) algorithms. In addition, a roadmap of extending WS standard techniques for realizing semantic, functional and QoS-based WSDi is presented.*

5. Kyriakos Kritikos, Dimitris Plexousakis. *Semantic QoS-based Web Service Discovery Algorithms.* Fifth European Conference on Web Services, pages 181–190, IEEE Computer Society, 2007.

   Abstract: *The success of the Web Service (WS) paradigm has led to a proliferation of available WSs, which are advertised in WS registries. While sophisticated semantic WS discovery algorithms are operating on these registries to return matchmaking results with high precision and recall, many functionally-equivalent WSs are returned. The solution to the above problem comes in terms of semantic QoS-based description and discovery of WSs. We have already presented a rich and extensible ontology language for QoS-based WS description called OWL-Q. We have also proposed a semantic QoS metric matching algorithm. Based on this algorithm, we have extended a CSP-based approach for QoS-based WS discovery. In this paper, we firstly analyze the evolution of OWL-Q and its extension with SWRL rules, we propose a modification to the metric matching algorithm and we show the way the metric alignment process takes place. Then we propose two novel semantic QoS-based WS Discovery algorithms that return matches even for over-constrained QoS-based WS requests. The first one deals with unary constraints while the second one is more generic. Finally, implementation aspects of our QoS-based WS discovery system are discussed.*

6. Kyriakos Kritikos, Dimitris Plexousakis. *Semantic QoS Metric Matching.* ECOWS, pages 265–274, 2006.

   Abstract: *As the Web Service paradigm gains popularity for its promise to transform the way business* *is conducted, the number of deployed Web Services grows with a fast rate. While sophisticated semantic discovery mechanisms have been invented to overcome the UDDI's syntactic discovery solution in order to provide more recallable and precise results, the amount of functionally equivalent Web Services returned is still large. The solution to this problem is the description of the QoS non-functional aspect of Web Services. QoS encompasses the performance of Web Services and can be used as a discriminator factor for refining Web Service advertisement result lists. However, most scientific efforts presented so far are purely syntactic and are not capturing all aspects of QoS-based Web Service description leading to imprecise syntactic discovery mechanisms. This paper presents a novel, rich and extensible ontology-based approach for describing QoS of Web Services that complements OWL-S. It is shown that, by using this approach and by introducing the concept of semantic QoS metric matching, QoS-based syntactic matchmaking and selection algorithms are transformed to semantic ones leading to better results.*

7. Vikas Deora, Jianhua Shao, Gareth Shercliff, Patrick J. Stockreisser, W. A. Gray, N. J. Fiddian. *Incorporating QoS Specifications in Service Discovery.* WISE Workshops, pages 252–263, 2004.

   Abstract: *In this paper, we extend the current approaches to service discovery in a service oriented computing environment, such as Web Services and Grid, by allowing service providers and consumers to express their promises and requirements for quality of service (QoS). More specifically, we allow service providers to advertise their services in an extended DAML-S that supports quality specifications, and we allow service consumers to request services by stating required quality levels. We propose a model here for incorporating QoS specifications and requirements in service discovery, and describe how matchmaking between advertised and requested services based on functional as well as quality requirements is supported in our model.*

8. Gwen Salaün, Lucas Bordeaux, Marco Schaerf. *Describing and Reasoning on Web Services using Process Algebra.* ICWS, 2004.

   Abstract: *We argue that essential facets of web services, and especially those useful to understand their interaction, can be described using process-algebraic notations. Web service description and execution languages such as BPEL are essentially process description languages; they are based on primitives for behaviour description and message exchange which can also be found in more abstract process algebras. One legitimate question is therefore whether the formal approach and the sophisticated tools introduced for process algebra can be used to improve the effectiveness and the reliability of web service development. Our investigations suggest a positive answer, and we claim that process algebras provide a very complete and satisfactory assistance to the whole process of web service development. We show on a case study that readily available tools based on process algebra are effective at verifying that web services conform their requirements and respect properties. We advocate their use both at the design stage and for reverse engineering issues. More prospectively, we discuss how they can be helpful to tackle choreography issues.*

9. Vladimir Tosic, Kruti Patel, Bernard Pagurek. *WSOL - Web Service Offerings Language.* CAiSE '02/ WES '02: Revised Papers from the International Workshop on Web Services, E-Business, and the Semantic Web, pages 57–67, Springer-Verlag, 2002.

   Abstract: *WSOL (Web Service Offerings Language) is an XML (Extensible Markup Language) notation compatible with the WSDL (Web Services Description Language) standard. While WSDL is used for describing operations provided by Web Services, WSOL enables formal specification of multiple classes of service for one Web Service. A service offering is a formal representation of one class of service for a Web Service. As classes of service for Web Services are determined by combinations of various constraints, WSOL enables formal specification of functional constraints, some QoS (a.k.a., non-functional) constraints, simple access rights (for differentiation of service), price, and relationships with other service offerings of the same Web Service. Describing a Web Service in WSOL, in addition to WSDL, enables selection of a more appropriate Web Service and service offering for particular circumstances. Further, it supports dynamic adaptation and management of Web Service compositions using manipulation of service offerings.*

22 Publications in **Specification**

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

Articles in Refereed Conferences:

1. Wei-Tek Tsai, Yinong Chen, Raymond A. Paul. *Specification-Based Verification and Validation of Web Services and Service-Oriented Operating Systems.* WORDS, pages 139–147, 2005.

   Abstract: *Service-Oriented Architecture (SOA) and Web Services (WS) have received significant attention recently. Even though WS are based on open standards and support software interoperability, but the trustworthy issues of WS has actually limited the growth of WS applications as organizations do not trust those WS developed by other vendors and at the same time they do not have access to the source code. This paper addressed this issue by proposing several solutions including specification-based verification and validation, collaborative testing, and group testing. The key concept is that it is possible to provide a comprehensive evaluation of WS even if their source code is not available.*

Technical Reports and Manuals:

1. The OMG Group. *UML$^{\text{TM}}$ Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms.* Num. ptc/2005-05-02, The OMG Group, May 2005.

2. Akhil Sahai, Anna Durante, Vijay Machiraju. *Towards Automated SLA Management for Web Services.* Num. HPL-2001-310, HP Laboratories, July 2002.

   Abstract: *In order to automate SLA management it is essential to specify SLAs in precise and unambiguous manner as well as keep the specification flexible. While precision will help automate the process of monitoring and metric collection, flexibility will enable extending it to unforeseen service level agreement specifications.*

3. Svend Frolund, Jari Koistinen. *QML: A Language for Quality of Service Specification.* Num. HPL-98-10, 63 pages, HP Laboratories, 1998.

   Abstract: *To be competitive, future software system must provide not only the correct functionality, but also an adequate level of quality of service (QoS). By QoS, we refer to non-functional properties, such as reliability, performance, timing, and security. To provide an adequate level of QoS, software systems need to include capabilities such as QoS negotiation, monitoring, and adaptation. These capabilities all require the expected and the provided QoS levels to be explicitly specified. QoS can be specified statically at the time* *of implementation, design, or dynamically at deployment or runtime. To facilitate QoS specification, we present a general Quality of service Modeling Language (QML) for defining multi-category QoS specifications for components in distributed object systems. QML is designed to support QoS in general, encompassing QoS categories such as reliability, performance, security, and timing. QoS specification in QML facilitate the static decomposition of a software system into components with precisely specified QoS boundaries. They also facilitate dynamic QoS functions, such as negotiations, monitoring, and adaptation. QML is designed for a good fit with object-oriented distributed architectures and concepts such as interfaces and inheritance. It also allows specification at a fine-grained level for operations, operation arguments, and attributes. QML enables user- defined QoS categories, and allows specifications within those categories to be associated with component interface definitions. In addition, checks can be made dynamically to determine whether one QML specification satisfies another. This mechanism allows us to dynamically match QoS requirements and offers*

23 Publications in **Testing**

Articles in Refereed Conferences:

1. Yuan Yuan, Zhongjie Li, Wei Sun. *A Graph-Search Based Approach to BPEL4WS Test Generation.* ICSEA, 14 pages, 2006.

   Abstract: *Business Process Execution Language for Web Services (BPEL4WS) is a kind of concurrent programming languages with several special features that raise special challenges for verification and testing. This paper proposes a graph-search based approach to BPEL test case generation, which effectively deals with BPEL concurrency semantics. This approach defines an extension of CFG (Control Flow Graph) - BPEL Flow Graph (BFG) - to represent a BPEL program in a graphical model. Then concurrent test paths can be generated by traversing the BFG model, and test data for each path can be generated using a constraint solving method. Finally test paths and data are combined into complete test cases.*

24 Publications in **Verification**

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

2. Bianculli, D., Ghezzi, C., Spoletini, P.. *A Model Checking Approach to Verify BPEL4WS Workflows.* Service-Oriented Computing and Applications, 2007. SOCA '07. IEEE International Conference on, To Appear, June 2007.

   Abstract: *The increasing diffusion of service oriented computing in critical business transactions demands reliability and correctness of the workflow logic representing web service orchestrations. We present an approach for the formal verification of workflow-based compositions of web services, described in BPEL4WS. Workflow processes can be verified in isolation, assuming that the external services invoked are known only through their interface. It is also possible to verify that the actual composition of two or more processes behaves correctly. We can verify deadlock freedom, properties expressed as data-bound assertions written in WS-CoL, a specification language for web services, and LTL temporal properties. Our approach is based on the software model checker Bogor, whose language supports the modeling of all BPEL4WS constructs. We provide an empirical evaluation of our approach and we compare the results with other BPEL4WS model checking tools.*

3. Chun Ouyanga, Eric Verbeekb, Wil M.P. van der Aalst, Stephan Breutel, Marlon Dumas, Arthur H.M. ter Hofstede. *Formal semantics and analysis of control flow in WS-BPEL.* Science of Computer Programming, Vol. 67, Num. 2-3, pages 162–198, April 2007.

   Abstract: *Web service composition refers to the creation of new (Web) services by combining functionalities provided by existing ones. A number of domain-specific languages for service composition have been proposed, with consensus being formed around a process-oriented language known as WS-BPEL (or BPEL). The kernel of BPEL consists of simple communication primitives that may be combined using control-flow constructs expressing sequence, branching, parallelism, synchronization, etc. We present a comprehensive and rigorously defined mapping of BPEL constructs onto Petri net structures, and use this for the analysis of various dynamic properties related to unreachable activities, conflicting messages, garbage collection, conformance checking, and deadlocks and lifelocks in interaction processes. We use a mapping onto Petri nets because this allows us to use existing theoretical results and analysis tools. Unlike approaches based on finite state machines, we do not need to construct the state space, and can use structural analysis (e.g., transition invariants) instead. We have implemented a tool that translates BPEL processes into Petri nets and then applies Petri-net-based analysis techniques. This tool has been tested on different examples, and has been used to answer a variety of questions.*

4. Wil M. P. van der Aalst. *Business alignment: using process mining as a tool for Delta analysis and conformance testing.* Requir. Eng., Vol. 10, Num. 3, pages 198–211, 2005.

   Abstract: *Increasingly, business processes are being controlled and/or monitored by information systems. As a result, many business processes leave their ”footprints” in transactional information systems, i.e., business events are recorded in so-called event logs. Process mining aims at improving this by providing techniques and tools for discovering process, control, data, organizational, and social structures from event logs, i.e., the basic idea of process mining is to diagnose business processes by mining event logs for knowledge. In this paper we focus on the potential use of process mining for measuring business alignment, i.e., comparing the real behavior of an information system or its users with the intended or expected behavior. We identify two ways to create and/or maintain the fit between business processes and supporting information systems: Delta analysis and conformance testing. Delta analysis compares the discovered model (i.e., an abstraction derived from the actual process) with some predefined processes model (e.g., the workflow model or reference model used to configure the system). Conformance testing attempts to quantify the ”fit” between the event log and some predefined processes model. In this paper, we show that Delta analysis and conformance testing can be used to analyze business alignment as long as the actual events are logged and users have some control over the process.*

5. Mariya Koshkina, Franck van Breugel. *Modelling and verifying web service orchestration by means of the concurrency workbench.* SIGSOFT Softw. Eng. Notes, Vol. 29, Num. 5, pages 1–10, ACM, 2004.

   Abstract: *Verification techniques like model checking, preorder checking and equivalence checking are shown to be relevant to web service orchestration. The Concurrency Workbench of the New Century (CWB) is a verification tool that supports these verification techniques. By means of the Process Algebra Compiler (PAC), the CWB is modified to support the BPE-calculus. The BPE-calculus is a small language, based on BPEL4WS, to express web service orchestration. Both the syntax and the semantics of the BPE-calculus are formally defined. These are subsequently used as input for the PAC. As output, the PAC produces modules that are incorporated into the CWB so that it supports the BPE-calculus and, hence, provides a verification tool for web service orchestration.*

6. Marco Pistore, Marco Roveri, Paolo Busetta. *Requirements-Driven Verification of Web Services.* Electr. Notes Theor. Comput. Sci., Vol. 105, pages 95–108, 2004.

   Abstract: *We propose a requirements-driven approach to the design and verification of Web services. The proposed methodology starts from a requirements model, which defines a business domain at a ”strategic” level, describing the participating actors, their mutual dependencies, goals, requirements, and expectations. This business requirements model is then refined into a business process model. In this refinement, definitions of the processes carried out by the actors of the domain are added to the model in the form of BPEL4WS code. We show how to exploit model checking techniques for the verification of the specification, both at the requirements and at the process level. At the requirements level, model checking is used to validate the specification against a set of queries specified by the designer; at the process level, it is used to verify if the BPEL4WS processes satisfy the constraints described in the requirements model.*

7. Gian Luigi Ferrari, Stefania Gnesi, Ugo Montanari, Marco Pistore. *A model-checking verification environment for mobile processes.* ACM Trans. Softw. Eng. Methodol., Vol. 12, Num. 4, pages 440–473, 2003.

   Abstract: *This article presents a semantic-based environment for reasoning about the behavior of mobile systems. The verification environment, called HAL, exploits a novel automata-like model that allows finite-state verification of systems specified in the π-calculus. The HAL system is able to interface with several efficient toolkits (e.g. model-checkers) to determine whether or not certain properties hold for a given specification. We report experimental results on some case studies.*

Articles in Refereed Conferences:

1. A. Rozinat, W.M.P. van der Aalst. *Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models.* Business Process Management Workshops,

Lecture Notes in Computer Science, Vol. 3812, pages 163–176, February 2006.

Abstract: *Most information systems log events (e.g., transaction logs, audit trails) to audit and monitor the processes they support. At the same time, many of these processes have been explicitly modeled. For example, SAP R/3 logs events in transaction logs and there are EPCs (Event-driven Process Chains) describing the so-called reference models. These reference models describe how the system should be used. The coexistence of event logs and process models raises an interesting question: "Does the event log conform to the process model and vice versa?". This paper demonstrates that there is not a simple answer to this question. To tackle the problem, we distinguish two dimensions of conformance: fitness (the event log may be the result of the process modeled) and appropriateness (the model is a likely candidate from a structural and behavioral point of view). Different metrics have been defined and a Conformance Checker has been implemented within the ProM Framework.*

2. Wil M. P. van der Aalst, Marlon Dumas, Chun Ouyang, Anne Rozinat, H. M. W. Verbeek. *Choreography Conformance Checking: An Approach based on BPEL and Petri Nets.* The Role of Business Processes in Service Oriented Architectures, 2006.

   Abstract: *Recently, languages such as BPEL and CDL have been proposed to describe the way services can interact from a behavioral perspective. 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 paper investigates a possible usage of behavioral service descriptions, namely as input for conformance checking. Conformance checking is the act of verifying whether one or more parties stick to the agreed-upon behavior by observing the actual behavior, e.g., the exchange of messages between all parties. This paper shows that it is possible to translate BPEL business protocols to Petri nets and to relate SOAP messages to transitions in the Petri net. As a result, Petri net-based conformance checking techniques can be used to quantify fitness (whether the observed behavior is possible in the business protocol) and appropriateness (whether the observed behavior "overfits" or "underfits" the business protocol). Moreover, non-conformance can be visualized to pinpoint deviations. The approach has been implemented in the context of the ProM n framework.*

3. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, Marco Montali, Sergio Storari, Paolo Torroni. *Computational Logic for Run-Time Verification of Web Services Choreographies: Exploiting the OCS-SI Tool.* WS-FM, pages 58–72, 2006.

   Abstract: *In this work, we investigate the feasibility of using a framework based on computational logic, and mainly defined in the context of Multi-Agent Systems for Global Computing (SOCS UE Project), for modeling choreographies of Web Services with respect to the conversational aspect. One of the fundamental motivations of using computational logic, beside its declarative and highly expressive nature, is given by its operational counterpart, that can provide a proof-theoretic framework able to verify the consistency of services designed in a cooperative and incremental manner. In particular, in this paper we show that suitable "Social Integrity Constraints", introduced in the SOCS social model, can be used for specifying global protocols at the choreography level. In this way, we can use a suitable tool, derived from the proof-procedure defined in the context of the SOCS project, to check at run-time whether a set of existing services behave in a conformant manner w.r.t. the defined choreography.*

4. Howard Foster, Sebastian Uchitel, Jeff Magee, Jeff Kramer. *LTSA-WS: a tool for model-based verification of web service compositions and choreography.* ICSE '06: Proceedings of the 28th international conference on Software engineering, pages 771–774, ACM, 2006.

   Abstract: *In this paper we describe a tool for a model-based approach to verifying compositions of web service implementations. The tool supports verification of properties created from design specifications and implementation models to confirm expected results from the viewpoints of both the designer and implementer. Scenarios are modeled in UML, in the form of Message Sequence Charts (MSCs), and then compiled into the Finite State Process (FSP) process algebra to concisely model the required behavior. BPEL4WS implementations are mechanically translated to FSP to allow an equivalence trace verification process to be performed. By providing early design verification and validation, the implementation, testing and deployment of web service compositions can be eased through the understanding of the behavior exhibited by the composition. The approach is implemented as a plug-in for the Eclipse development environment providing cooperating tools for specification, formal modeling, verification and validation of the composition process.*

5. Raman Kazhamiakin, Paritosh K. Pandya, Marco Pistore. *Representation, Verification, and Computation of Timed Properties in Web Service Compositions.* ICWS, pages 497–504, 2006.

   Abstract: *In this paper we address the problem of qualitative and quantitative analysis of timing aspects of Web service compositions defined as a set of BPEL4WS processes. We introduce a formalism, called Web Service Timed State Transition Systems (WSTTS), to capture the timed behavior of the composite web services. We also exploit an interval temporal logic to express complex timed assumptions and requirements on the system's behavior. Building on top of this formalization, we provide techniques and tools for modelchecking BPEL4WS compositions against time-related requirements. We also present a symbolic algorithm that can be used to compute duration bounds of behavioral intervals that satisfy such requirements. We perform a preliminary experimental evaluation of our approach and tools with the help of an e-Government case study.*

6. Marina Mongiello, Daniela Castelluccia. *Modelling and verification of BPEL business processes.* Fourth Workshop on Model-Based Development of Computer-Based Systems, 2006.

   Abstract: *A business process is a complex Web service with functions provided by different Web services, which are already existing in Web and are dynamically integrated for granting a more complex business task. For this reason, business processes have become more and more diffuse in B2B and B2C domains, so that the importance of their activities asks for a high-level of reliability. Methods and tools for supporting automatic system verification and validation could be useful. Among the techniques of automatic verification, we choose model checking method, because we applied it efficiently for verification of a single Web service and in this paper we extend the area of application also in business processes. Descriptions of the behavior of a business process are coded using a standard language, BPEL4WS, that has broadly spread because it is able to describe a business process as both an executable process and an abstract process. Therefore, we model a BPEL description of a generic business process with a formal model and we formalize correctness properties about the reliability of the business process design. Also, we build a framework that performs automatic verification of formal models of business processes through NuSMV model checker. If there is a violation of correctness specifications, NuSMV provides counter-examples, so we can locate errors and effect right changes for correcting business process design.*

7. Mohsen Rouached, Olivier Perrin, Claude Godart. *Towards Formal Verification of Web Service Composition.* Business Process Management, pages 257–273, 2006.

   Abstract: *Web services composition is an emerging paradigm for enabling application integration within and across organizational boundaries. Current Web services composition proposals, such as BPML, WS-BPEL, WSCI, and OWL-S, provide solutions for describing the control and data flows in Web service composition. However, such proposals remain at the descriptive level, without providing any kind of mechanisms or tool support for analysis and verification. Therefore, there is a growing interest for the verification techniques which enable designers to test and repair design errors even before actual running of the service, or allow designers to detect erroneous properties and formally verify whether the service process design does have certain desired properties. In this paper, we propose to verify Web services composition using an event driven approach. We assume Web services that are coordinated by a composition process expressed in WSBPEL and we use Event Calculus to specify the properties and requirements to be monitored.*

8. Huang, H., Tsai, W.-T., Paul, R.. *Automated model checking and testing for composite Web services.* pages 300–307, May 2005.

   Abstract: *Web services form a new distributed computing paradigm. Collaborative verification and validation are important when Web services from different vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web services by model checking the process model of OWL-S (Web ontology language for Web services) and to validate them by the test cases automatically generated in the model checking process. We extend the BLAST, a model checker that handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

9. Bernd-Holger Schlingloff, Axel Martens, Karsten Schmidt. *Modeling and Model Checking Web Services.* Vol. 126, pages 3–26, Elsevier, March 2005.

   Abstract: *We give an overview on web services and the web service technology stack. We then show how to build Petri net models of web services formulated in the specification language BPEL4WS. We define an abstract correctness criterion for these models and study the automated verification according to this criterion. Finally, we relate correctness of web service models to the model checking problem for alternating temporal logics.*

10. Aysu Betin-Can, Tevfik Bultan, Xiang Fu. *Design for verification for asynchronously communicating Web services.* WWW, pages 750–759, 2005.

    Abstract: *We present a design for verification approach to developing reliable web services. We focus on composite web services which consist of asynchronously communicating peers. Our goal is to automatically verify properties of interactions among such peers. We propose a design pattern that eases the development of such web services and enables a modular, assume-guarantee style verification strategy. In the proposed design pattern, each peer is associated with a behavioral interface description which specifies how that peer will interact with other peers. Using these peer interfaces we automatically generate BPEL specifications to publish for interoperability. Assuming that the participating peers behave according to their interfaces, we verify safety and liveness properties about the global behavior of the composite web service during behavior verification. During interface verification, we check that each peer implementation conforms to its interface. Using the modularity in the proposed design pattern, we are able to perform the interface verification of each peer and the behavior verification as separate steps. Our experiments show that, using this modular approach, one can automatically and efficiently verify web service implementations.*

11. Hai Huang, Wei-Tek Tsai, Raymond Paul, Yinong Chen. *Automated Model Checking and Testing for Composite Web Services..* 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2005), pages 300–307, IEEE Computer Society, 2005.

    Abstract: *Web Services form a new distributed computing paradigm. Collaborative verification and validation are important when Web Services from differen vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web Services by model checking the proces model of OWL-S (Web Ontology Language for Web Services) and to validate them by the test cases automatically generated in the model checking process. We extend the BLAST, a model checker tha handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

12. Wei-Tek Tsai, Yinong Chen, Raymond A. Paul. *Specification-Based Verification and Validation of Web Services and Service-Oriented Operating Systems.* WORDS, pages 139–147, 2005.

    Abstract: *Service-Oriented Architecture (SOA) and Web Services (WS) have received significant attention recently. Even though WS are based on open standards and support software interoperability, but the trustworthy issues of WS has actually limited the growth of WS applications as organizations do not trust those WS developed by other vendors and at the same time they do not have access to the source code. This paper addressed this issue by proposing several solutions including specification-based verification and validation, collaborative testing, and group testing. The key concept is that it is possible to provide a comprehensive evaluation of WS even if their source code is not available.*

13. Foster, H., Uchitel, S., Magee, J., Kramer, J.. *Compatibility verification for Web service choreography.* pages 738–741, July 2004.

    Abstract: *In this paper we discuss a model-based approach to verifying process interactions for coordinated Web service compositions. The approach uses finite state machine representations of Web service orchestrations and assigns semantics to the distributed process interactions. The move towards implementing Web service compositions by multiple interested parties as a form of distributed system architecture motivates* *the need for supporting compatibility verification of activities and transactions in all the processes. The described approach is supported by a suite of cooperating tools for specification, formal modeling and providing verification results from orchestrated Web service interactions.*

14. Jesus Arias-Fisteus, Luis Sanchez Fernandez, Carlos Delgado Kloos. *Formal Verification of BPEL4WS Business Collaborations.* E-Commerce and Web Technologies, 5th International Conference, EC-Web 2004, Proceedings, Lecture Notes in Computer Science, Vol. 3182, pages 76–85, Springer Verlag, 2004.

    Abstract: *Web services are a very appropriate communication mechanism to perform distributed business processes among several organisations. These processes should be reliable, because a failure in them can cause high economic losses. To increase their reliability at design time, we have developed VERBUS, a framework for the formal verification of business processes. VERBUS can automatically translate business process definitions to specifications verifiable in several available tools. It is based on a modular and extensible architecture: new process definition languages and verification tools can be added easily to the framework. The prototype of VERBUS presented in this work can verify BPEL4WS process specifications, by translating them to Promela. The Promela specifications are verified with the well known model checker Spin. In this paper we describe the general architecture of VERBUS and how BPEL4WS specifications are translated and verified. The explanation is completed by describing what types of properties can be verified and providing an overview of the implementation.*

15. Tevfik Bultan, Xiang Fu, Jianwen Su. *Tools for Automated Verification of Web Services.* ATVA, pages 8–10, 2004.

16. Xiang Fu, Tevfik Bultan, Jianwen Su. *Analysis of Interacting BPEL Web Services.* WWW'04: Proceedings of the 13th International Conference on World Wide Web, pages 621–630, ACM Press, 2004.

    Abstract: *This paper presents a set of tools and techniques for analyzing interactions of composite web services which are specified in BPEL and communicate through asynchronous XML messages. We model the interactions of composite web services as conversations, the global sequence of messages ex- changed by the web services. As opposed to earlier work, our tool-set handles rich data manipulation via XPath expressions. This allows us to verify designs at a more detailed level and check properties about message content. We present a framework where BPEL specifications of web ser- vices are translated to an intermediate representation, followed by the translation of the intermediate representation to a verification language. As an intermediate representation we use guarded automata augmented with unbounded queues for incoming messages, where the guards are expressed as XPath expressions. As the target verification language we use Promela, input language of the model checker SPIN. Since SPIN model checker is a finite-state verification tool we can only achieve partial verification by fixing the sizes of the input queues in the translation. We propose the concept of synchronizability to address this problem. We show that if a composite web service is synchronizable, then its conversation set remains same when asynchronous communication is replaced with synchronous communication. We give a set of sufficient conditions that guarantee synchronizability and that can be checked statically. Based on our synchronizability results, we show that a large class of composite web services with unbounded input queues can be completely verified using a finite state model checker such as SPIN.*

17. Xiang Fu, Tevfik Bultan, Jianwen Su. *Model checking XML manipulating software.* ISSTA, pages 252–262, 2004.

    Abstract: *The use of XML as the de facto data exchange standard has allowed integration of heterogeneous web based software systems regardless of implementation platforms and programming languages. On the other hand, the rich tree-structured data representation, and the expressive XML query languages (such as XPath) make formal specification and verification of software systems that manipulate XML data a challenge. In this paper, we present our initial efforts in automated verification of XML data manipulation operations using the SPIN model checker. We present algorithms for translating (bounded) XML data and XPath expressions to Promela, the input language of SPIN. The techniques presented in this paper constitute the basis of our Web Service Analysis Tool (WSAT) which verifies LTL properties of composite web services.*

18. Howard Foster, Sebastián Uchitel, Jeff Magee, Jeff Kramer. *Model-based Verification of Web Service Compositions..* 18th IEEE International Conference on Automated Software Engineering (ASE 2003) , pages 152–163, IEEE Computer Society, 2003.

    Abstract: *In this paper we discuss a model-based approach to verifying web service compositions for web service implementations. The approach supports verification against specification models and assigns semantics to the behavior of implementation models so as to confirm expected results for both the designer and implementer. Specifications of the design are modeled in UML, in the form of Message Sequence Charts (MSCs), and mechanically compiled into the Finite State Process notation (FSP) to concisely describe and reason about the concurrent programs. Implementations are mechanically translated to FSP to allow a trace equivalence verification process to be performed. By providing early design verification, the implementation, testing and deployment of web service compositions can be eased through the understanding of the differences, limitations and undesirable traces allowed by the composition. The approach is supported by a suite of cooperating tools for specification, formal modeling and trace animation of the composition workflow.*

19. Srini Narayanan, Sheila A. McIlraith. *Simulation, verification and automated composition of web services.* WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 77–88, ACM, 2002.

    Abstract: *Web services – Web-accessible programs and devices - are a key application area for the Semantic Web. With the proliferation of Web services and the evolution towards the Semantic Web comes the opportunity to automate various Web services tasks. Our objective is to enable markup and automated reasoning technology to describe, simulate, compose, test, and verify compositions of Web services. We take as our starting point the DAML-S DAML+OIL ontology for describing the capabilities of Web services. We define the semantics for a relevant subset of DAML-S in terms of a first-order logical language. With the semantics in hand, we encode our service descriptions in a Petri Net formalism and provide decision procedures for Web service simulation, verification and composition. We also provide an analysis of the complexity of these tasks under different restrictions to the DAML-S composite services we can describe. Finally, we present an implementation of our analysis techniques. This implementation takes as input a DAML-S description of a Web service, automatically generates a Petri Net and performs the desired analysis. Such a tool has broad applicability both as a back end to existing manual Web service composition tools, and as a stand-alone tool for Web service developers.*

20. Wil M. P. van der Aalst. *Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques.* Business Process Management, pages 161–183, 2000.

    Abstract: *Workflow management systems facilitate the everyday operation of business processes by taking care of the logistic control of work. In contrast to traditional information systems, they attempt to support frequent changes of the workflows at hand. Therefore, the need for analysis methods to verify the correctness of workflows is becoming more prominent. In this chapter we present a method based on Petri nets. This analysis method exploits the structure of the Petri net to find potential errors in the design of the workflow. Moreover, the analysis method allows for the compositional verification of workflows.*

Books and Monographs:

1. Mariya Koshkina. *Verification of Business Processes for Web Services.* Ms. Thesis, York University - Department of Computer Science, October 2003.

    Abstract: *The Business Process Execution Language for Web Services (BPEL4WS or simply BPEL) is a recently developed language, which is used to specify interactions between web services. Among its features it allows specification of concurrent behavior. Erroneous specification can lead to such problems as deadlock. In our research we focus on the concurrency mechanism in BPEL. Our main goal is to analyze processes in order to detect possible deadlocks. To achieve this we introduce a process algebra called the BPE-calculus. It is a small language which captures all the BPEL features relevant to the analysis. This process algebra is modelled using a labeled transition system. An existing verification tool called the Concurrency Workbench is customized to use our BPE-calculus. This tool allows us to verify many properties of BPE-calculus processes specified in a logic called the mu-calculus, including deadlock freedom.*

Articles in Books and Other Collections:

1. Natasha Sharygina, Daniel Kröning. *Model Checking with Abstraction for Web Services.* Test and Analysis of Web Services, pages 121–145, 2007.

   Abstract: *Web services are highly distributed programs and, thus, are prone to concurrency-related errors. Model checking is a powerful technique to identify flaws in concurrent systems. However, the existing model checkers have only very limited support for the programming languages and communication mechanisms used by typical implementations of web services. This chapter presents a formalization of communication semantics geared for web services, and an automated way to extract formal models from programs implementing web services for automatic formal analysis. The formal models are analyzed by means of a symbolic model checker that implements automatic abstraction refinement. Our implementation takes one or more PHP5 programs as input, and is able to verify joint properties of these programs running concurrently.*

Publications in Refereed Workshops:

1. S. Nakajima. *Model-Checking Behavioral Specification of BPEL Applications.* Proceedings of the International Workshop on Web Languages and Formal Methods, WLFM 2005, 2005.

   Abstract: *To provide a framework to compose lots of specialised services flexibly, BPEL is proposed to describe Web service flows. Since the Web service flow description is basically a distributed collaboration, writing correct programs in BPEL is not easy. Verifying BPEL program prior to its execution is essential. This paper proposes a method to extract the behavioral specification from a BPEL appliation program and to analyze it by using the SPIN model checker. With the adequate abstraction method and support for DPE, the method can analyze all the four example cases in the BPEL standard document.*

2. S. Nakajima. *Model-Checking Verification for Reliable Web Service.* OOPSLA Workshop on Object-Oriented Web Services, 2002.

   Abstract: *Model-checking is a promising technique for the verification and validation of software systems. Web service, an emerging technology in the Internet, is an autonomous server that may offer an individual service. It sometimes requires to combine more than one to meet our requirements. WSFL(Web Services Flow Language) is proposed as a language to provide means to describe Web service aggregation. We are interested in how much the software model-checking technique can be used as a basis for raising reliability of Web service, Web service flow descriptions in particular. Our experience shows that faulty flow descriptions can be identified with the proposed method. The method is also very helpful in studying an alternative semantics of the WSFL in regard to the handling of dataflows.*

25 Publications in **Validation**

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

Articles in Refereed Conferences:

1. A. Rozinat, W.M.P. van der Aalst. *Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models.* Business Process Management Workshops, Lecture Notes in Computer Science, Vol. 3812, pages 163–176, February 2006.

   Abstract: *Most information systems log events (e.g., transaction logs, audit trails) to audit and monitor the processes they support. At the same time, many of these processes have been explicitly modeled. For example, SAP R/3 logs events in transaction logs and there are EPCs (Event-driven Process Chains) describing the so-called reference models. These reference models describe how the system should be used. The coexistence of event logs and process models raises an interesting question: "Does the event log conform to the process model and vice versa?". This paper demonstrates that there is not a simple answer to this question. To tackle the problem, we distinguish two dimensions of conformance: fitness (the event log may be the result of the process modeled) and appropriateness (the model is a likely candidate from a structural and behavioral point of view). Different metrics have been defined and a Conformance Checker has been implemented within the ProM Framework.*

2. Huang, H., Tsai, W.-T., Paul, R.. *Automated model checking and testing for composite Web services.* pages 300–307, May 2005.

   Abstract: *Web services form a new distributed computing paradigm. Collaborative verification and validation are important when Web services from different vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web services by model checking the process model of OWL-S (Web ontology language for Web services) and to validate them by the test cases automatically generated in the model checking process. We extend the BLAST, a model checker that handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

3. Wei-Tek Tsai, Yinong Chen, Raymond A. Paul. *Specification-Based Verification and Validation of Web Services and Service-Oriented Operating Systems.* WORDS, pages 139–147, 2005.

   Abstract: *Service-Oriented Architecture (SOA) and Web Services (WS) have received significant attention* *recently. Even though WS are based on open standards and support software interoperability, but the trustworthy issues of WS has actually limited the growth of WS applications as organizations do not trust those WS developed by other vendors and at the same time they do not have access to the source code. This paper addressed this issue by proposing several solutions including specification-based verification and validation, collaborative testing, and group testing. The key concept is that it is possible to provide a comprehensive evaluation of WS even if their source code is not available.*

4. Roozbeh Farahbod, Uwe Glässer, Mona Vajihollahi. *Specification and Validation of the Business Process Execution Language for Web Services.* Abstract State Machines, pages 78–94, 2004.

   Abstract: *We formally define an abstract executable semantics for the Business Process Execution Language for Web Services in terms of a distributed ASM. The goal of this work is to support the design and standardization of the language. There is a need for formalism. It will allow us to not only reason about the current specification and related issues, but also uncover issues that would otherwise go unnoticed. Empirical deduction is not sufficient. - Issue 42, OASIS WSBPEL TC. The language definition assumes an infrastructure for running Web services on some asynchronous communication architecture. A business process is built on top of a collection of Web services performing continuous interactions with the outside world by sending and receiving messages over a communication network. The underlying execution model is characterized by its concurrent and reactive behavior making it particularly difficult to predict dynamic system properties with a sufficient degree of detail and precision under all circumstances.*

26 Publications in **Web Services**

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

2. Roberto Lucchi, Manuel Mazzara. *A pi-calculus based semantics for WS-BPEL .* Journal of Logic and Algebraic Programming, Vol. 70, Num. 1, pages 96–118, January 2007.

   Abstract: *Recently, the term Web services orchestration has been introduced to address some issues related to Web services composition, that is the way of defining a complex service out of simpler ones. Several proposals for describing orchestration for business processes have been presented in the last years and many of these languages make use of concepts as long-running transactions and compensations for coping with error handling. WS-BPEL 2.0, the most credited candidate for becoming a standard, provides three different mechanisms allowing to cope with abnormal situations: exception, event and compensation handling. This complexity makes it difficult to formally define the framework, thus limiting the formal reasoning about the designed applications. In this paper we advocate that three different mechanisms for error handling are not necessary and we formalize a novel orchestration language based on the idea of event notification as the unique error handling mechanism. To this end, we formally define the three BPEL mechanisms in terms of our calculus. It is possible to take advantages of this formal description in two ways. Firstly, this language represents by itself a proposal of simplification for WS-BPEL 2.0 including an unambiguous specification. Secondly, an implementor of an actual WS-BPEL 2.0 orchestration engine could implement simply this single mechanism providing all the remaining ones by compilation. With this attempt we intend to give a concrete contribute towards the improvement of the quality of the BPEL specification, the applicability of BPEL itself and the implementation of real orchestration engines. Finally, as a case study we consider some of the hundreds of open issues met by the WS-BPEL designers and we propose a solution making use of the experience gained developing our algebra.*

3. D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, P. Plebani. *PAWS: A Framework for Executing Adaptive Web-Service Processes.* IEEE Software, To Appear, 2007.

   Abstract: *The processes with adaptive Web services framework couples design-time and runtime mechanisms to flexibly and adoptively execute managed Web-services-based business processes.*

4. Mike P. Papazoglou, Benedikt Kratz. *Web services technology in support of business transactions.* Service Oriented Computing and Applications, Vol. 1, Num. 1, pages 51–63, 2007.

   Abstract: *Advanced business applications typically involve well-defined business functions such as payment processing, shipping and tracking, determining new product offerings, granting/extending credit, managing market risk and so on. These reflect commonly standard business functions that apply to a variety of application scenarios. Although such business functions drive transactional applications between trading partners*

they are completely external to current Web services transaction mechanisms and are only expressed as part of application logic. To remedy this situation, this paper proposes a business-aware Web services transaction model and support mechanisms, which is driven by common business functions. The model allows expressing business functions such as payment and credit conditions, delivery conditions, business agreements stipulated in SLAs, liabilities and dispute resolution policies. It allows blending these business functions with QoS criteria such as security support to guarantee integrity of information, confidentiality, and non-repudiation.

5. Ma, K.J.. *Web services: what's real and what's not?.* IT Professional, Vol. 7, Num. 2, pages 14–21, March 2005.

   Abstract: *The idea of abstracted, well-defined, and ubiquitously invokable services replacing proprietary interprocess communications has been a goal of system designers for a long time. The rise of Web services has led to a lot of misconceptions about how they can and cannot support the holy grail of a service-oriented architecture (SOA). This article seeks to put Web services in perspective, explaining their current capabilities and what industry can expect from them in the near term. It gives an overview of how technologies such as the Extensible Markup Language (XML), XML schemas, Extensible Stylesheet Language Transformations (XSLT), the Simple Object Access Protocol (SOAP), the Web Services Description Language (WSDL), and universal description, discovery, and integration (UDDI) fit into the equation for an SOA.*

6. James Pasley. *How BPEL and SOA Are Changing Web Services Development.* IEEE Internet Computing, Vol. 9, Num. 3, pages 60–67, 2005.

   Abstract: *As the use of Web services grows, organizations are increasingly choosing the Business Process Execution Language for modeling business processes within the Web services architecture. In addition to orchestrating organizations' Web services, BPEL's strengths include asynchronous message handling, reliability, and recovery. By developing Web services with BPEL in mind, organizations can implement aspects of the service-oriented architecture that might previously have been difficult to achieve.*

7. Neal Leavitt. *Are Web Services Finally Ready to Deliver?.* IEEE Computer, Vol. 37, Num. 11, pages 14–18, November 2004.

   Abstract: *Web services, in brief, are a framework of software technologies designed to support interoperable machine-to-machine interaction over a network. Companies on different systems can use Web services to exchange information online with business partners, customers, and suppliers. Various standards organizations and industry consortia are developing Web services specifications without a unifying authority. Organizations such as the World Wide Web Consortium (W3C), the Organization for the Advancement of Structured Information Standards (OASIS), the Liberty Alliance Project, and the Web Services Interoperability Organization (WS-I) have developed or reviewed numerous standards. A primary goal of Web services is to unlock a new generation of e-commerce applications.*

8. Alexander Keller, Heiko Ludwig. *The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services.* J. Network Syst. Manage., Vol. 11, Num. 1, 2003.

   Abstract: *We describe a novel framework for specifying and monitoring Service Level Agreements (SLA) for Web Services. SLA monitoring and enforcement becomes increasingly important in a Web Service environment where enterprises rely on services that may be subscribed dynamically and on demand. For economic and practical reasons, we want an automated provisioning process for both the service itself as well as the SLA management system. It measures and monitors the QoS parameters, checks the agreed-upon service levels, and reports violations to the authorized parties involved in the SLA management process. The Web Service Level Agreement (WSLA) framework, our approach to these issues, is targeted at defining and monitoring SLAs for Web Services. Although WSLA has been designed for a Web Services environment, it is applicable as well to any inter-domain management scenario such as business process and service management or the management of networks, systems and applications in general. The WSLA framework consists of a flexible and extensible language based on XML Schema and a runtime architecture comprising several SLA monitoring services, which may be outsourced to third parties to ensure a maximum of accuracy. WSLA enables service customers and providers to unambiguously define a wide variety of SLAs, specify the SLA parameters and the way how they are measured, and relate them to managed resource instrumentations. Upon receipt of an SLA specification, the WSLA monitoring services are automatically configured to enforce*

the SLA. An implementation of the WSLA framework, the SLA Compliance Monitor, is publicly available as part of the IBM Web Services Toolkit.

Articles in Refereed Conferences:

1. Fabio Barbon, Paolo Traverso, Marco Pistore, Michele Trainotti. *Run-Time Monitoring of Instances and Classes of Web Service Compositions.* ICWS, pages 63–71, 2006.

   Abstract: *The run-time monitoring of web service compositions has been widely acknowledged as a significant and challenging problem. In this paper, we propose a novel solution to the problem of monitoring web services implemented in BPEL. We devise an architecture that clearly separates the business logic of a web service from its monitoring functionality. The architecture supports both "instance monitors" that deal with the execution of a single instance of BPEL process, as well as "class monitors" that report aggregated information about all the instances of a BPEL process. We also define a language for the specification of instance and class monitors. The language allows for specifying boolean, statistic, and time-related properties. Finally, we devise a technique for the automatic translation of all these kinds of monitors to Java programs.*

2. Dimka Karastoyanova, Alejandro Houspanossian, Mariano Cilia, Frank Leymann, Alejandro P. Buchmann. *Extending BPEL for Run Time Adaptability.* EDOC, pages 15–26, 2005.

   Abstract: *The existing Web Service Flow (WS-flow) technologies enable both static and dynamic binding of participating Web services (WSs) on the process model level. Adaptability on per-instance basis is not sufficiently supported and therefore must be addressed to improve process flexibility upon changes in the environment. Ad-hoc process instance changes can be enabled by swapping participating WS instances, by modifying port Types of the partners to be invoked, and by changing process logic. In this work we address the problem of dynamic binding of WSs to WS-flow instances at run time, i.e. the ability to exchange a WS instance participating in a WS-flow instance with an alternative one. The problem is additionally complicated by the fact that the execution of a process depends on its deployment. We describe the "find and bind" mechanism, and we show its representation as a BPEL extension. We discuss the benefits that could be gained and the disadvantages it brings in. The mechanism extends and improves the existing process technologies. It facilitates a precisely controlled policy-based selection of WSs at run time and also provides for process instance repair, while maintaining simplicity. We also discuss a prototypical implementation of the presented functionality.*

3. Khaled Mahbub, George Spanoudakis. *Run-time Monitoring of Requirements for Systems Composed of Web-Services: Initial Implementation and Evaluation Experience.* ICWS, pages 257–265, 2005.

   Abstract: *This paper describes a framework supporting the runtime monitoring of requirements for systems implemented as compositions of Web-services specified in BPEL. The requirements that can be monitored are specified in event calculus. The paper presents an overview of the framework and describes the architecture and implementation of a tool that we have developed to operationalise it. It also presents the results of a preliminary experimental evaluation of the framework.*

4. Vladimir Tosic, Bernard Pagurek, Kruti Patel. *WSOL - A Language for the Formal Specification of Classes of Service for Web Services.* ICWS, pages 375–381, 2003.

   Abstract: *We are developing a language, Web Service Offerings Language (WSOL), for the formal specification of various constraints, management statements, and classes of service for Web Services. WSOL is an XML (Extensible Markup Language) notation compatible with WSDL (Web Services Description Language). A service offering in WSOL is a formal description of one class of service of a Web Service. It contains formal representation of various constraints: functional (pre-, post-, and future- conditions), Quality of Service (QoS, a.k.a. non-functional, extra- functional), and access rights. It also contains management statements, such as statements about prices, monetary penalties, and management responsibilities. One Web Service can be associated with multiple service offerings. For easier specification of similar service offerings, WSOL enables specification of constraint groups (CGs) and constraint group templates (CGTs). We have also*

developed a format for representation of dynamic relationships between service offerings. WSOL service offerings are simple contracts and SLAs (Service Level Agreements) between Web Services. Describing a Web Service in WSOL, in addition to WSDL, enables monitoring, metering, and management of Web Services. The Web Service, its consumer, or one or more designated third parties (usually SOAP message intermediaries) can meter QoS metrics and evaluate constraints in WSOL service offerings. Further, manipulation of service offerings can be used for dynamic adaptation and management of Web Service compositions. In addition, WSOL supports selection of a more appropriate Web Service and service offering for particular circumstances. The main distinctive characteristics of WSOL, compared to recent related works, are its expressive power, features that reduce run-time overhead, and orientation towards management applications.

Technical Reports and Manuals:

1. Siddharth Bajaj, Don Box, Dave Chappell, Francisco Curbera, Glen Daniels, Phillip Hallam-Baker, Maryann Hondo, Chris Kaler, Dave Langworthy, Anthony Nadalin, Nataraj Nagaratnam, Hemma Prafullchandra, Claus von Riegen, Daniel Roth, Jeffrey Schlimmer, Chris Sharp, John Shewchuk, Asir Vedamuthu, Ümit Yal cınalp, David Orchard . *Web Services Policy Framework (WS- Policy).* IBM, March 2006.
   Abstract: *The Web Services Policy Framework (WS-Policy) provides a general purpose model and corresponding syntax to describe the policies of a Web Service. WS-Policy defines a base set of constructs that can be used and extended by other Web services specifications to describe a broad range of service requirements and capabilities.*

2. The OASIS Group. *Quality Model for Web Services.* The Oasis Group, September 2005.
   Abstract: *The purpose of this document is to provide a model for Web services quality management and quality factors in the process of developing and using Web services. We define the consistent and systematic conceptual model of Web services quality, which may be used by intimate associates, i.e. stakeholders, developers, service providers, and customers of Web services.*

3. The Oasis Group. *Summary of Quality Model for Web Services.* The Oasis Group, 2005.

4. Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, Richard Franck . *Web Service Level Agreement (WSLA) Language Specification.* IBM Corporation, 2003.
   Abstract: *This document describes the specification language for service level agreements for Web Services, the Web Service Level Agreement (WSLA) language. WSLAs are agreements between a service provider and a customer and as such define the obligations of the parties involved. Primarily, this is the obligation of a service provider to perform a service according to agreed-upon guarantees for IT-level service parameters (such as availability, response time and throughput) for Web Services.*
   An SLA also specifies the measures to be taken in case of deviation and failure to meet the asserted service guarantees, for example, a notification of the service customer. The assertions of the service provider are based on a detailed definition of service parameters including the algorithms – how basic metrics should be measured in systems and how they are aggregated into composite metrics and SLA parameters. In addition, a WSLA can express the operations of monitoring and managing the service. This may include third parties (such as Management Service Providers) that contribute to the measurement of metrics, supervision of guarantees or even the management of deviations of service guarantees. These multi-party constellations necessitate the definition of the interactions among the parties supervising the WSLA.
   However, a WSLA only covers the agreed common view of a service between the parties involved. To actually act as a participant in a WSLA, parties have various degrees of freedom to define an implementation policy for a service and its supervision. Typically, the obligations of a WSLA must be translated into system-level configuration information, which can be proprietary to each party involved.

5. Akhil Sahai, Anna Durante, Vijay Machiraju. *Towards Automated SLA Management for Web Services.* Num. HPL-2001-310, HP Laboratories, July 2002.
   Abstract: *In order to automate SLA management it is essential to specify SLAs in precise and unambiguous*

manner as well as keep the specification flexible. While precision will help automate the process of monitoring and metric collection, flexibility will enable extending it to unforeseen service level agreement specifications.

6. Anbazhagan Mani, Arun Nagarajan. *Understanding quality of service for Web services.* IBM, 2002.

   Abstract: *With the widespread proliferation of Web services, quality of service (QoS) will become a significant factor in distinguishing the success of service providers. QoS determines the service usability and utility, both of which influence the popularity of the service. In this article, we look at the various Web service QoS requirements, bottlenecks affecting performance of Web services, approaches of providing service quality, transactional services, and a simple method of measuring response time of your Web services using the service proxy.*

27 Publications in **Workflow**

Articles in Refereed Journals:

1. Bianculli, D., Ghezzi, C., Spoletini, P.. *A Model Checking Approach to Verify BPEL4WS Workflows.* Service-Oriented Computing and Applications, 2007. SOCA '07. IEEE International Conference on, To Appear, June 2007.

   Abstract: *The increasing diffusion of service oriented computing in critical business transactions demands reliability and correctness of the workflow logic representing web service orchestrations. We present an approach for the formal verification of workflow-based compositions of web services, described in BPEL4WS. Workflow processes can be verified in isolation, assuming that the external services invoked are known only through their interface. It is also possible to verify that the actual composition of two or more processes behaves correctly. We can verify deadlock freedom, properties expressed as data-bound assertions written in WS-CoL, a specification language for web services, and LTL temporal properties. Our approach is based on the software model checker Bogor, whose language supports the modeling of all BPEL4WS constructs. We provide an empirical evaluation of our approach and we compare the results with other BPEL4WS model checking tools.*

2. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski, Alistair P. Barros. *Workflow Patterns.* Distributed and Parallel Databases, Vol. 14, Num. 1, pages 5–51, 2003.

   Abstract: *Differences in features supported by the various contemporary commercial workflow management systems point to different insights of suitability and different levels of expressive power. The challenge, which we undertake in this paper, is to systematically address workflow requirements, from basic to complex. Many of the more complex requirements identified, recur quite frequently in the analysis phases of workflow projects, however their implementation is uncertain in current products. Requirements for workflow languages are indicated through workflow patterns. In this context, patterns address business requirements in an imperative workflow style expression, but are removed from specific workflow languages. The paper describes a number of workflow patterns addressing what we believe identify comprehensive workflow functionality. These patterns provide the basis for an in-depth comparison of a number of commercially availablework flow management systems. As such, this paper can be seen as the academic response to evaluations made by prestigious consulting companies. Typically, these evaluations hardly consider the workflow modeling language and routing capabilities, and focus more on the purely technical and commercial aspects.*

Articles in Refereed Conferences:

1. Michael C. Jaeger, Gregor Rojec-Goldmann, Gero Mühl. *QoS Aggregation for Web Service Composition using Workflow Patterns.* EDOC, pages 149–159, 2004.

   Abstract: *Contributions in the field of Web services have identified that (a) finding matches between semantic descriptions of advertised and requested services and (b) nonfunctional characteristics - the quality of service (QoS) - are the most crucial criteria for composition of Web services. A mechanism is introduced that determines the QoS of a Web service composition by aggregating the QoS dimensions of the individual services. This allows to verify whether a set of services selected for composition satisfies the QoS requirements for the whole composition. The aggregation performed builds upon abstract composition patterns, which represent basic structural elements of a composition, like sequence, loop, or parallel execution. This work focusses on workflow management environments. We define composition patterns that are derived from Van der Aalst's et al. comprehensive collection of workflow patterns. The resulting aggregation schema supports the same structural elements as found in workflows. Furthermore, the aggregation of several QoS dimensions is discussed.*

2. Wil M. P. van der Aalst, Alistair P. Barros, Arthur H. M. ter Hofstede, Bartek Kiepuszewski. *Advanced Workflow Patterns.* CoopIS, pages 18–29, 2000.

   Abstract: *Conventional workflow functionality like task sequencing, split parallelism, join synchronization and iteration have proven effective for business process automation and have widespread support in current* *workflow products. However, newer requirements for workflows are enconntered in practice, opening grave uncertainties about the extensions for current languages. Different concepts, although outwardly appearing to be more or less the same, are based on different paradigms, have fundamentally different semantics and different levels of applicability -more specialized for modeling or more generalized for workflow engine posit. By way of developmental insight of new requirements, we define workflow patterns which are described imperatively but independently of current workflow languages. These patterns provide the basis for an in-depth comparison of 12 workflow management systems. As such, the work reported in this paper can be seen as the academic response to evaluations made by prestigious consulting companies. Typically, these evaluations hardly consider the workflow modeling language and routing capabilities and focus more on the purely technical and commercial aspects.*

3. Wil M. P. van der Aalst. *Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques.* Business Process Management, pages 161–183, 2000.
   Abstract: *Workflow management systems facilitate the everyday operation of business processes by taking care of the logistic control of work. In contrast to traditional information systems, they attempt to support frequent changes of the workflows at hand. Therefore, the need for analysis methods to verify the correctness of workflows is becoming more prominent. In this chapter we present a method based on Petri nets. This analysis method exploits the structure of the Petri net to find potential errors in the design of the workflow. Moreover, the analysis method allows for the compositional verification of workflows.*

Books and Monographs:

1. J. Cardoso. *Quality of Service and Semantic Composition of Workflows.* Ph.D. Thesis, University of Georgia, 2002.
   Abstract: *Workflow management systems (WfMSs) have been used to support a variety of business processes. As organizations adopt new working models, such as e-commerce, new challenges arise for workflow systems. These challenges include support for the adequate management of quality of service (QoS) and the development of new solutions to facilitate the composition of workflow applications involving Web services. The good management of QoS directly impacts the success of organizations participating in e- commerce activities by better fulfilling customer expectations and achieving customer satisfaction. To enable adequate QoS management, research is required to develop mechanisms that specify, compute, monitor, and control the QoS of the products or services to be delivered. The composition of workflows to model e-service applications differs from the design of traditional workflows due to the number of Web services available during the composition process and to their heterogeneity. Two main problems need to be solved: how to efficiently discover Web services and how to facilitate their interoperability. To enhance WfMSs with QoS management, we have developed a QoS model that allows for the description of nonfunctional aspects of workflow components, from a quality of service perspective. To automatically compute the overall QoS of a workflow, we have developed a mathematical model and implemented an algorithm (SWR algorithm). Our QoS model and mathematical model have been validated with the deployment and execution of a set of production workflows in the area of genetics. The analysis of the collected data proves that our models provide a suitable framework for estimating, predicting, and analyzing the QoS of production workflows. To support, facilitate, and assist the composition of workflows involving Web services, we present a solution based on ontologies. We have developed an algorithm that workflow systems and discovery mechanisms can use to find Web services with desired interfaces and operational metrics, and to assist designers in resolving heterogeneity issues among Web services. Our approach provides an important solution to enhance Web service discovery and interoperability.*

Chapter 2

Publications by Year

1 Publications in 2008

Articles in Refereed Conferences:

1. Moreno Marzolla, Raffaela Mirandola. *Performance Prediction of Web Service Workflows.* Software Architectures, Components, and Applications, Lecture Notes in Computer Science, Vol. 4880, pages 127–144, January 2008.

   Abstract: *Web Services play an important role in the SOA paradigm, as they allow services to be selected on-the-fly to build applications out of existing components. In this scenario, the BPEL notation can be used as an orchestration language which allows the user to describe interactions with Web Services in a standard way. The performance of a BPEL workflow is a very important factor for deciding which components must be selected, or to choose whether a given sequence of interactions can provide the requested quality of service. Due to its very dynamic nature, workflow performance evaluation can not be accomplished using traditional, heavy-weight techniques. In this paper we present a multi-view approach for the performance prediction of service-based applications encompassing both users and service provider(s) perspectives. As a first step towards the realization of this integrated framework we present an efficient approach for performance assessment of Web Service workflows described using the BPEL notation. Starting from annotated BPEL and WSDL specifications, we derive performance bounds on response time and throughput. In such a way users are able to assess the efficiency of a BPEL workflow, while service provider(s) can perform sizing studies or estimate performance gains of alternative upgrades to existing systems. To bring this approach to fruition we developed a prototype tool called bpel2qnbound, using which we analyze a simple case study.*

2. Sheng Chen, Liang Bao, Ping Chen. *OptBPEL: A Tool for Performance Optimization of BPEL Process.* Software Composition, pages 141–148, 2008.

   Abstract: *The Business Process Execution Language (BPEL) is now a de facto standard for specifying and executing business process for web service composition and orchestration. As more and more web services are composed using BPEL, tuning these compositions and gain better performance becomes increasingly important. This paper presents our approach for optimizing the BPEL process and introduces OptBPEL, a tool for performance optimization of BPEL process. The approach starts from the optimization of synchronization structure concerning link in BPEL. After that, some concurrency analysis techniques are applied to obtain further performance improvement. Finally, we give some experiments and prove the efficiency of these optimization algorithms used in OptBPEL.*

3. Kyriakos Kritikos, Dimitris Plexousakis. *Evaluation of QoS-Based Web Service Matchmaking Algorithms.* International Conference on Services Computing, 2008.

   Abstract: *Web Service (WS) discovery is a prerequisite for achieving WS composition and orchestration. Although a lot of esearch has been conducted on the functional discovery of WSs, the proposed techniques fall short when faced with the foreseen increase in the number of (potentially functionally- equivalent) WSs. The above situation can be resolved with the addition of non-functional (Quality of Service (QoS)) discovery mechanisms to WS discovery engines. QoS-based WS matchmaking algorithms have been devised for this reason. However, they are either slow - as they are based on ontology reasoners - or produce inaccurate results. Inaccuracy is caused both by the syntactic matching of QoS concepts and by wrong matchmaking metrics. In this paper, we present two Constraint Programming (CP) QoS-based WS discovery algorithms for unary constrained WS specifications that produce accurate results with good performance. We also evaluate these algorithms on matchmaking time, precision and recall in different settings in order to demonstrate their efficiency and accuracy.*

2 Publications in 2007

Articles in Refereed Journals:

1. L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, P. Spoletini. *Validation of web service compositions.* IET Software, Vol. 1, Num. 6, pages 219–232, December 2007.

   Abstract: *Web services support software architectures that can evolve dynamically. In particular, in this paper the focus is on architectures where services are composed (orchestrated) through a workflow described in the business process execution language (BPEL). It is assumed that the resulting composite service refers to external services through assertions that specify their expected functional and non-functional properties. On the basis of these assertions, the composite service may be verified at design time by checking that it ensures certain relevant properties. Because of the dynamic nature of web services and the multiple stakeholders involved in their provision, however, the external services may evolve dynamically, and even unexpectedly. They may become inconsistent with respect to the assertions against which the workflow was verified during development. As a consequence, validation of the composition must extend to run time. In this work, an assertion language, called assertion language for BPEL process interactions (ALBERT), is introduced; it can be used to specify both functional and non-functional properties. An environment which supports design-time verification of ALBERT assertions for BPEL workflows via model checking is also described. At run time, the assertions can be turned into checks that a software monitor performs on the composite system to verify that it continues to guarantee its required properties. A TeleAssistance application is provided as a running example to illustrate our validation framework.*

2. Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann. *Service-Oriented Computing: State of the Art and Research Challenges.* IEEE Computer, To Appear, November 2007.

   Abstract: *Service-oriented computing promotes the idea of assembling application components into a network of services that can be loosely coupled to create flexible, dynamic business processes and agile applications that span organizations and computing platforms. An SOC research road map provides a context for exploring ongoing research activities.*

3. Ardagna, D., Pernici, B.. *Adaptive Service Composition in Flexible Processes.* Software Engineering, IEEE Transactions on, Vol. 33, Num. 6, pages 369–384, June 2007.

   Abstract: *In advanced service oriented systems, complex applications, described as abstract business processes, can be executed by invoking a number of available Web services. End users can specify different preferences and constraints and service selection can be performed dynamically identifying the best set of services available at runtime. In this paper, we introduce a new modeling approach to the Web service selection problem that is particularly effective for large processes and when QoS constraints are severe. In the model, the Web service selection problem is formalized as a mixed integer linear programming problem, loops peeling is adopted in the optimization, and constraints posed by stateful Web services are considered. Moreover, negotiation techniques are exploited to identify a feasible solution of the problem, if one does not exist. Experimental results compare our method with other solutions proposed in the literature and demonstrate the effectiveness of our approach toward the identification of an optimal solution to the QoS constrained Web service selection problem*

4. Bianculli, D., Ghezzi, C., Spoletini, P.. *A Model Checking Approach to Verify BPEL4WS Workflows.* Service-Oriented Computing and Applications, 2007. SOCA '07. IEEE International Conference on, To Appear, June 2007.

   Abstract: *The increasing diffusion of service oriented computing in critical business transactions demands reliability and correctness of the workflow logic representing web service orchestrations. We present an approach for the formal verification of workflow-based compositions of web services, described in BPEL4WS. Workflow processes can be verified in isolation, assuming that the external services invoked are known only through their interface. It is also possible to verify that the actual composition of two or more processes behaves correctly. We can verify deadlock freedom, properties expressed as data-bound assertions written in WS-CoL, a specification language for web services, and LTL temporal properties. Our approach is based on* *the software model checker Bogor, whose language supports the modeling of all BPEL4WS constructs. We provide an empirical evaluation of our approach and we compare the results with other BPEL4WS model checking tools.*

5. Chun Ouyanga, Eric Verbeekb, Wil M.P. van der Aalst, Stephan Breutel, Marlon Dumas, Arthur H.M. ter Hofstede. *Formal semantics and analysis of control flow in WS-BPEL.* Science of Computer Programming, Vol. 67, Num. 2-3, pages 162–198, April 2007.

   Abstract: *Web service composition refers to the creation of new (Web) services by combining functionalities provided by existing ones. A number of domain-specific languages for service composition have been proposed, with consensus being formed around a process-oriented language known as WS-BPEL (or BPEL). The kernel of BPEL consists of simple communication primitives that may be combined using control-flow constructs expressing sequence, branching, parallelism, synchronization, etc. We present a comprehensive and rigorously defined mapping of BPEL constructs onto Petri net structures, and use this for the analysis of various dynamic properties related to unreachable activities, conflicting messages, garbage collection, conformance checking, and deadlocks and lifelocks in interaction processes. We use a mapping onto Petri nets because this allows us to use existing theoretical results and analysis tools. Unlike approaches based on finite state machines, we do not need to construct the state space, and can use structural analysis (e.g., transition invariants) instead. We have implemented a tool that translates BPEL processes into Petri nets and then applies Petri-net-based analysis techniques. This tool has been tested on different examples, and has been used to answer a variety of questions.*

6. Roberto Lucchi, Manuel Mazzara. *A pi-calculus based semantics for WS-BPEL .* Journal of Logic and Algebraic Programming, Vol. 70, Num. 1, pages 96–118, January 2007.

   Abstract: *Recently, the term Web services orchestration has been introduced to address some issues related to Web services composition, that is the way of defining a complex service out of simpler ones. Several proposals for describing orchestration for business processes have been presented in the last years and many of these languages make use of concepts as long-running transactions and compensations for coping with error handling. WS-BPEL 2.0, the most credited candidate for becoming a standard, provides three different mechanisms allowing to cope with abnormal situations: exception, event and compensation handling. This complexity makes it difficult to formally define the framework, thus limiting the formal reasoning about the designed applications. In this paper we advocate that three different mechanisms for error handling are not necessary and we formalize a novel orchestration language based on the idea of event notification as the unique error handling mechanism. To this end, we formally define the three BPEL mechanisms in terms of our calculus. It is possible to take advantages of this formal description in two ways. Firstly, this language represents by itself a proposal of simplification for WS-BPEL 2.0 including an unambiguous specification. Secondly, an implementor of an actual WS-BPEL 2.0 orchestration engine could implement simply this single mechanism providing all the remaining ones by compilation. With this attempt we intend to give a concrete contribute towards the improvement of the quality of the BPEL specification, the applicability of BPEL itself and the implementation of real orchestration engines. Finally, as a case study we consider some of the hundreds of open issues met by the WS-BPEL designers and we propose a solution making use of the experience gained developing our algebra.*

7. D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, P. Plebani. *PAWS: A Framework for Executing Adaptive Web-Service Processes.* IEEE Software, To Appear, 2007.

   Abstract: *The processes with adaptive Web services framework couples design-time and runtime mechanisms to flexibly and adoptively execute managed Web-services-based business processes.*

8. Stefano Bistarelli, Ugo Montanari, Francesca Rossi, Francesco Santini. *Unicast and Multicast Qos Routing with Soft Constraint Logic Programming.* CoRR, Vol. abs/0704.1783, 2007.

   Abstract: *We present a formal model to represent and solve the unicast/multicast routing problem in networks with Quality of Service (QoS) requirements. To attain this, first we translate the network adapting it to a weighted graph (unicast) or and-or graph (multicast), where the weight on a connector corresponds to the multidimensional cost of sending a packet on the related network link: each component of the weights vector represents a different QoS metric value (e.g. bandwidth, cost, delay, packet loss). The second step* *consists in writing this graph as a program in Soft Constraint Logic Programming (SCLP): the engine of this framework is then able to find the best paths/trees by optimizing their costs and solving the constraints imposed on them (e.g. delay ¡ 40msec), thus finding a solution to QoS routing problems. Moreover, c-semiring structures are a convenient tool to model QoS metrics. At last, we provide an implementation of the framework over scale-free networks and we suggest how the performance can be improved.*

9. J. Cardoso. *Complexity analysis of BPEL Web processes.* Software Process: Improvement and Practice, Vol. 12, Num. 1, pages 35–49, Whiley-Interscience, 2007.

   Abstract: *Several organizations have already realized the potential of using WS-BEPL, the Process Execution Language for Web Services, to model the behavior of Web services in business processes. WS-BPEL provides a model for describing simple or complex interactions between business partners. In some cases, WS-BPEL process designs can be highly complex, due, for example, to the vast number of Web services carried out in global markets. High complexity in a process has several undesirable drawbacks; it may result in poor understandability, more errors, defects, and exceptions, leading to processes requiring more time to be developed, tested, and maintained. Therefore, excessive complexity should be avoided. Processes that are highly complex tend be less flexible, since it is more complicated to make changes to the process. The major goal of this article is to present two metrics to analyze the control-flow complexity (CFC) of WS-BPEL Web processes. The metrics are to be used at design-time to evaluate the complexity of a process design before implementation actually takes place.*

10. Giorgia Lodi, Fabio Panzieri, Davide Rossi, Elisa Turrini. *SLA-Driven Clustering of QoS-Aware Application Servers.* IEEE Trans. Software Eng., Vol. 33, Num. 3, pages 186–197, 2007.

    Abstract: *In this paper, we discuss the design, implementation, and experimental evaluation of a middleware architecture for enabling service level agreement (SLA)-driven clustering of QoS-aware application servers. Our middleware architecture supports application server technologies with dynamic resource management: application servers can dynamically change the amount of clustered resources assigned to hosted applications on-demand so as to meet application-level quality of service (QoS) requirements. These requirements can include timeliness, availability, and high throughput and are specified in SLAs. A prototype of our architecture has been implemented using the open-source J2EE application server JBoss. The evaluation of this prototype shows that our approach makes possible JBoss' resource usage optimization and allows JBoss to effectively meet the QoS requirements of the applications it hosts, i.e., to honor the SLAs of those applications.*

11. Fabrizio Montesi, Claudio Guidi, Roberto Lucchi, Gianluigi Zavattaro. *JOLIE: a Java Orchestration Language Interpreter Engine.* Electr. Notes Theor. Comput. Sci., Vol. 181, pages 19–33, 2007.

    Abstract: *Service oriented computing is an emerging paradigm for programming distributed applications based on services. Services are simple software elements that supply their functionalities by exhibiting their interfaces and that can be invoked by exploiting simple communication primitives. The emerging mechanism exploited in service oriented computing for composing services -in order to provide more complex functionalities- is by means of orchestrators. An orchestrator is able to invoke and coordinate other services by exploiting typical workflow patterns such as parallel composition, sequencing and choices. Examples of orchestration languages are XLANG [IBM, "XLANG: Web Services for Business Process Design," http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm] and WS-BPEL [OASIS, "Web Services Business Process Execution Language Version 2.0, Working Draft," http://www.oasis-open.org/committees/download.php/10347/wsbpe specification-draft-120204.htm]. In this paper we present JOLIE, an interpreter and engine for orchestration programs. The main novelties of JOLIE are that it provides an easy to use development environment (because it supports a more programmer friendly C/Java-like syntax instead of an XML-based syntax) and it is based on a solid mathematical underlying model (developed in previous works of the authors [N. Busi, R. Gorrieri, C. Guidi, R. Lucchi and G. Zavattaro, Towards a formal framework for Choreography, in: Proc. of 3rd International Workshop on Distributed and Mobile Collaboration (DMC 2005) (2005), N. Busi, R. Gorrieri, C. Guidi, R. Lucchi and G. Zavattaro, Choreography and orchestration conformance for system design, in: Proc. of 8th International Conference on Coordination Models and Languages (COORDINATION'06), LNCS to appear, 2006, C. Guidi and R. Lucchi, Mobility mechanisms in service oriented* *computing, in: Proc. of 8th International Conference on on Formal Methods for Open Object-Based Distributed Systems (FMOODS'06), LNCS to appear, 2006]).*

12. Mike P. Papazoglou, Benedikt Kratz. *Web services technology in support of business transactions.* Service Oriented Computing and Applications, Vol. 1, Num. 1, pages 51–63, 2007.
Abstract: *Advanced business applications typically involve well-defined business functions such as payment processing, shipping and tracking, determining new product offerings, granting/extending credit, managing market risk and so on. These reflect commonly standard business functions that apply to a variety of application scenarios. Although such business functions drive transactional applications between trading partners they are completely external to current Web services transaction mechanisms and are only expressed as part of application logic. To remedy this situation, this paper proposes a business-aware Web services transaction model and support mechanisms, which is driven by common business functions. The model allows expressing business functions such as payment and credit conditions, delivery conditions, business agreements stipulated in SLAs, liabilities and dispute resolution policies. It allows blending these business functions with QoS criteria such as security support to guarantee integrity of information, confidentiality, and non-repudiation.*

Articles in Refereed Conferences:

1. Kyriakos Kritikos, Dimitris Plexousakis. *OWL-Q for Semantic QoS-based Web Service Description and Discovery.* First International Joint Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, November 2007.
Abstract: *Semantic Web Services are emerging for their promise to produce a more accurate and precise Web Service discovery process. However, most of research approaches focus only on the functional part of semantic Web Service description. The above fact along with the proliferation of Web Services is highly probable to lead to a situation where Web Service registries will return many functionally-equivalent Web Service advertisements for each user request. This problem can be solved with the semantic description of QoS for Web Services. QoS is a set of non-functional properties encompassing performance and network-related characteristics of resources. So it can be used for distinguishing between functionally-equivalent Web Services. Current research approaches for QoS-based Web Service description are either syntactic or poor or non-extensible. To solve this problem, we have developed a rich and extensible ontological specification called OWL-Q for semantic QoS-based Web Service description. We analyze all OWL-Q parts and reason that rules should be added in order to support property inferencing and constraint enforcement. Finally, we line out our under-development semantic framework for QoS-based Web Service description and discovery.*

2. Cinzia Cappiello, Marco Comuzzi, Pierluigi Pleban. *On Automated Generation of Web Service Level Agreements .* Advanced Information Systems Engineering, Vol. 4495, Springer Verlag, June 2007.
Abstract: *Before a service invocation takes place, an agreement between the service provider and the service user might be required. Such an agreement is the result of a negotiation process between the two parties and defines how the service invocation has to occur. Considering the Service Oriented Computing paradigm, the relationship among providers and users is extremely loose. Traditional agreements are likely to concern long term relationships and to be manually performed. In this paper, we propose a model to generate service level agreement on-the-fly. Just before the invocation commences, the quality of the service is negotiated in order to generate a service level agreement tied to that specific invocation. Such an approach relies on a quality model that supports both users requirements and providers capabilities definition.*

3. Jonatha Anselmi, Danilo Ardagna, Paolo Cremonesi. *A QoS-based selection approach of autonomic grid services.* SOCP '07: Proceedings of the 2007 workshop on Service-oriented computing performance: aspects, issues, and approaches, pages 1–8, ACM, 2007.
Abstract: *The Web service composition (WSC) is the process of building an instance of an abstract workflow by combining appropriate Web services that satisfies given QoS requirements. In general, QoS requirements consists of a number of constraints. The selection process requires global optimization and can be formalized as a mixed integer linear programming problem which cannot be solved in polynomial time. However, since the number of submitted workflows is large and the QoS is highly dynamic, the fast*

selection of composite Web Services is particularly important. In this paper, we present a QoS broker-based framework for Web services execution in autonomic grid environments. The main goal of the framework is to support the broker in selecting Web services based on the required QoS. To achieve this goal, we propose a novel approach: since successive composed Web services requests can have the same task to Web service assignment, we address the Multiple Instance WSC (MI-WSC) problem optimizing simultaneously the set of requests which will be submitted to the system in the successive time interval instead of independently computing a solution for each incoming request. Experimental results show that the proposed algorithm has better performance with respect to existing techniques. Moreover, the qualities of the selected composite Web services are not significantly different from the optimal ones.

4. Gero Decker, Oliver Kopp, Frank Leymann, Mathias Weske. *BPEL4Chor: Extending BPEL for Modeling Choreographies.* ICWS, pages 296–303, 2007.

   Abstract: *The Business Process Execution Language (BPEL) is a language to orchestrate web services into a single business process. In a choreography view, several processes are interconnected and their interaction behavior is described from a global perspective. This paper shows how BPEL can be extended for defining choreographies. The proposed extensions (BPEL4Chor) distinguish between three aspects: (i) participant behavior descriptions, i.e. control flow dependencies in each participant, (ii) the participant topology, i.e. the existing participants and their interconnection using message links and (iii) participant groundings, i.e. concrete configurations for data formats and port types. As BPEL itself is used unchanged, the extensions facilitate a seamless integration between service choreographies and orchestrations. The suitability of the extensions is validated by assessing their support for the Service Interaction Patterns.*

5. Ester Giallonardo, Eugenio Zimeo. *More Semantics in QoS Matching.* SOCA, pages 163–171, 2007.

   Abstract: *The evolution of the Web towards a global computing environment is promoting new research efforts aimed at the formal characterization of Web Services QoS. Reasoning on QoS is a key to improve matching process during the discovery of desired services and a step towards the transformation of applications in collections of loosely coupled services virtually connected by semantic similarities. The paper presents the on QoS ontology, an openly available OWL ontology for QoS, and evaluates it in a QoS-aware matching environment. The ontology can be used to express functions of QoS metrics useful to improve the recall tied to the matching of a template request with target Web Services. To this end, the ontology introduces the concept of derivation in the matching process. This gives the possibility of matching a QoS template with published Web Services by deriving different QoS parameters when a one-to-one matching fails. The proposed matching algorithm utilizes a reasoner that exploits the ontology to avoid apparent mismatches. An experimental evaluation shows that exploiting QoS knowledge significantly improves matching recall without deteriorating precision.*

6. Dimka Karastoyanova, Branimir Wetzstein, Tammo van Lessen, Daniel Wutke, Jörg Nitzsche, Frank Leymann. *Semantic Service Bus: Architecture and Implementation of a Next Generation Middleware.* ICDE Workshops, pages 347–354, 2007.

   Abstract: *In this paper we present a middleware for the service oriented architecture, called the Semantic Service Bus. It is an advanced middleware possessing enhanced features, as compared to the conventional service buses. It is distinguished by the fact that it uses semantic description of service capabilities, and requirements towards services to enable more elaborate service discovery, selection, routing, composition and data mediation. The contributions of the paper are the conceptual architecture of the Semantic Service Bus and a prototypical implementation supporting different semantic Web service technologies (OWL-S and WSMO) and conventional Web services. Since mission critical application scenarios (for SOA) involve complex orchestrations of services, we have chosen to utilize semantically annotated service orchestrations as the applications to employ this middleware.*

7. Kyriakos Kritikos, Dimitris Plexousakis. *Requirements for QoS-based Web Service Description and Discovery.* COMPSAC (2), pages 467–472, 2007.

   Abstract: *The goal of Service Oriented Architectures is to enable the creation of business applications through the automatic discovery and composition of independently developed and deployed (Web) services.*

Automatic discovery of Web Services (WSs) can be achieved by incorporating semantics into a richer WS description model (WSDM) and the use of Semantic Web (SW) technologies in the WS matchmaking and selection models. A sufficiently rich WSDM should encompass not only functional but also non-functional aspects like Quality of Service (QoS). QoS is a set of performance attributes that has a substantial impact on WS requesters' expectations. Thus, it can be used as a discriminating factor of functionally-equivalent WSs. The focus of this paper is twofold: to analyze the requirements of a semantically rich QoS-based WSDM and to provide SW and constrainedbased mechanisms for enriching syntactic QoS-based WS Discovery (WSDi) algorithms. In addition, a roadmap of extending WS standard techniques for realizing semantic, functional and QoS-based WSDi is presented.

8. Kyriakos Kritikos, Dimitris Plexousakis. *Semantic QoS-based Web Service Discovery Algorithms.* Fifth European Conference on Web Services, pages 181–190, IEEE Computer Society, 2007.

   Abstract: *The success of the Web Service (WS) paradigm has led to a proliferation of available WSs, which are advertised in WS registries. While sophisticated semantic WS discovery algorithms are operating on these registries to return matchmaking results with high precision and recall, many functionally-equivalent WSs are returned. The solution to the above problem comes in terms of semantic QoS-based description and discovery of WSs. We have already presented a rich and extensible ontology language for QoS-based WS description called OWL-Q. We have also proposed a semantic QoS metric matching algorithm. Based on this algorithm, we have extended a CSP-based approach for QoS-based WS discovery. In this paper, we firstly analyze the evolution of OWL-Q and its extension with SWRL rules, we propose a modification to the metric matching algorithm and we show the way the metric alignment process takes place. Then we propose two novel semantic QoS-based WS Discovery algorithms that return matches even for over-constrained QoS-based WS requests. The first one deals with unary constraints while the second one is more generic. Finally, implementation aspects of our QoS-based WS discovery system are discussed.*

9. Anca Muscholl, Igor Walukiewicz. *A Lower Bound on Web Services Composition.* FoSSaCS, pages 274–286, 2007.

   Abstract: *A web service is modeled here as a finite state machine. A composition problem for web services is to decide if a given web service can be constructed from a given set of web services; where the construction is understood as a simulation of the specification by a fully asynchronous product of the given services. We show an EXPTIME-lower bound for this problem, thus matching the known upper bound. Our result also applies to richer models of web services, such as the Roman model.*

10. Elisabetta Di Nitto, Massimiliano Di Penta, Alessio Gambi, Gianluca Ripa, Maria Luisa Villani. *Negotiation of Service Level Agreements: An Architecture and a Search-Based Approach.* ICSOC, pages 295–306, 2007.

    Abstract: *Software systems built by composing existing services are more and more capturing the interest of researchers and practitioners. The envisaged long term scenario is that services, offered by some competing providers, are chosen by some consumers and used for their own purpose, possibly, in conjunction with other services. In the case the consumer is not anymore satisfied by the performance of some service, he can try to replace it with some other service. This implies the creation of a global market of services and poses new requirements concerning validation of exploited services, security of transactions engaged with services, trustworthiness, creation and negotiation of Service Level Agreements with these services. In this paper we focus on the last aspect and present our approach for negotiation of Service Level Agreements. Our architecture supports the actuation of various negotiation processes and offers a search-based algorithm to assist the negotiating parts in the achievement of an agreement.*

11. Qing Zhu, Shan Wang, Guorong Li, Guangqiang Liu, Xiaoyong Du. *QoS-Based Services Selecting and Optimizing Algorithms on Grid.* APWeb/WAIM Workshops, pages 156–167, 2007.

    Abstract: *QoS-Based Services Selecting and Optimizing Composition between the peers play an increasingly important role to ensure interoperability on Grid environment. However, the prohibitive cost of selecting, matching, mapping and composing algorithm has now become a key bottleneck hindering the deployment* *of a wide variety of Grid services. In this paper, we present QoS-Based Services Selecting and Optimizing Composition on Grid. First, it checks requesters' semantic in order to form candidate service graph. Second, it designs service selecting and mapping algorithms for optimizing the model. Third, it creates an executed plan of optimum composition on Grid. We conducted experiments to simulate and evaluate our approach.*

Articles in Books and Other Collections:

1. Natasha Sharygina, Daniel Kröning. *Model Checking with Abstraction for Web Services.* Test and Analysis of Web Services, pages 121–145, 2007.

   Abstract: *Web services are highly distributed programs and, thus, are prone to concurrency-related errors. Model checking is a powerful technique to identify flaws in concurrent systems. However, the existing model checkers have only very limited support for the programming languages and communication mechanisms used by typical implementations of web services. This chapter presents a formalization of communication semantics geared for web services, and an automated way to extract formal models from programs implementing web services for automatic formal analysis. The formal models are analyzed by means of a symbolic model checker that implements automatic abstraction refinement. Our implementation takes one or more PHP5 programs as input, and is able to verify joint properties of these programs running concurrently.*

Technical Reports and Manuals:

1. Oliver Kopp, Rania Khalaf, Frank Leymann. *Reaching Definitions Analysis Respecting Dead Path Elimination Semantics in BPEL Processes.* Num. 2007/04, Institut für Architektur von Anwendungssystemen, November 2007.

   Abstract: *The Business Process Execution Language for Web Services (BPEL) is a workflow language geared towards Service Oriented Computing. BPEL provides basic workflow capabilities, such as the ability to impose control on a set of activities via explicit links, as well as advanced features such as recovery and event handling. The area of workflow often requires understanding the data dependencies as well as the control dependencies between activities. This aids in business process design as well as in analysis and rengineering. BPEL, however, uses shared variables to model data. Activities read and write data to these variables, i.e. there is no explicit data link construct. In order to draw out the flow of data between the activities of a BPEL process, one must therefore perform data-flow analysis on that process. To address this problem, we present an algorithm that statically determines such data dependencies. "Statically" means, that we do not determine the dependencies per process instance at runtime, but on the process model without creating any instances. Mainstream data-flow analysis techniques are presented in [ALSU06, Muc97, NNH04]. However, these techniques cannot be directly applied to a BPEL process, since BPEL supports both parallelism and dead path elimination (DPE, [CDG+ 03, OAS07, CKLW03, BK05, LR00]). DPE is a technique used in BPEL to propagate activity disablement down paths that can no longer be executed. The algorithm in this paper consists of reaching definitions analysis dealing with DPE, enabling it to reduce the number of data dependencies when compared to approaches not dealing with DPE.*

2. Diane Jordan, John Evdemon, Alexandre Alves, Assaf Arkin, Sid Askary, Charlton Barreto, Ben Bloch, Francisco Curbera, Mark Ford, Yaron Goland, Alejandro Guzar, Neelakantan Kartha, Canyang Kevin Liu, Rania Khalaf, Dieter Knig, Mike Marin, Vinkesh Mehta, Satish Thatte, Danny van der Rijn, Prasad Yendluri, Alex Yiu . *Web Services Business Process Execution Language Version 2.0.* IBM, Microsoft, BEA, Intalio, Individual, Adobe Systems, Systinet, Active Endpoints, JBoss, Sterling Commerce, SAP, Deloitte, TIBCO Software, webMethods, Oracle, 2007.

   Abstract: *This document defines a language for specifying business process behavior based on Web Services. This language is called Web Services Business Process Execution Language (abbreviated to WS-BPEL in the rest of this document). Processes in WS-BPEL export and import functionality by using Web Service interfaces exclusively. Business processes can be described in two ways. Executable business processes model actual behavior of a participant in a business interaction. Abstract business processes are partially specified processes that are not intended to be executed. An Abstract Process may hide some of the required concrete* *operational details. Abstract Processes serve a descriptive role, with more than one possible use case, including observable behavior and process template. WS-BPEL is meant to be used to model the behavior of both Executable and Abstract Processes. WS-BPEL provides a language for the specification of Executable and Abstract business processes. By doing so, it extends the Web Services interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to- business spaces.*

3 Publications in 2006

Articles in Refereed Journals:

1. Wenhui Sun, Jinyu Zhang, Feng Liu. *WS-SLA: A Framework for Web Services Oriented Service Level Agreements.* Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference on, To Appear, May 2006.

   Abstract: *Service level agreements (SLA) becomes the prevailing business model for delivery of a large number of products and services in telecommunication and IT arenas. We presents an integrated service level agreement model applied to operating supporting system (OSS), on the basis of WfMC workflow reference model and service oriented architecture (SOA). And an application scenario instance of SLA system architecture is described by a set of operations models by Business Process Execution Language (BPEL)*

2. Christel Baier, Marjan Sirjani, Farhad Arbab, Jan J. M. M. Rutten. *Modeling component connectors in Reo by constraint automata.* Sci. Comput. Program., Vol. 61, Num. 2, pages 75–113, 2006.

   Abstract: *In this paper we introduce constraint automata and propose them as an operational model for Reo, an exogenous coordination language for compositional construction of component connectors based on a calculus of channels. By providing composition operators for constraint automata and defining notions of equivalence and refinement relations for them, this paper covers the foundations for building tools to address concerns such as the automated construction of the automaton for a given component connector, equivalence checking or containment checking of the behavior of two given connectors, and verification of coordination mechanisms.*

3. Mike P. Papazoglou, Willem-Jan van den Heuvel. *Service-oriented design and development methodology.* Int. J. Web Eng. Technol., Vol. 2, Num. 4, pages 412–442, 2006.

   Abstract: *Service Oriented Architectures (SOA) are rapidly emerging as the premier integration and architectural approach in contemporary, complex, heterogeneous computing environments. SOA is not simply about deploying software: it also requires that organisations evaluate their business models, come up with service-oriented analysis and design techniques, deployment and support plans, and carefully evaluate partner/customer/supplier relationships. Since SOA is based on open standards and is frequently realised using Web Services (WS), developing meaningful WS and business process specifications is an important requirement for SOA applications that leverage WS. Designers and developers cannot be expected to oversee a complex service-oriented development project without relying on a sound design and development methodology. This paper provides an overview of the methods and techniques used in service-oriented design and development. The aim of this paper is to examine a service development methodology from the point of view of both service producers and requesters and review the range of elements in this methodology that are available to them.*

Articles in Refereed Conferences:

1. Dmytro Rud, Andreas Schmietendorf, Reiner Dumke. *Performance Modeling of WS-BPEL-Based Web Service Compositions.* Services Computing Workshops, pages 140–147, September 2006.

   Abstract: *This paper addresses quality of service aspects of Web service orchestrations created using WS-BPEL from the standpoint of a Web service integrator. A mathematical model based on operations research techniques and formal semantics of WS-BPEL is proposed to estimate and forecast the influence of the execution of orchestrated processes on utilization and throughput of individual involved nodes and of the whole system. This model is applied to the optimization of service levels agreement process between the involved parties*

2. A. Rozinat, W.M.P. van der Aalst. *Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models.* Business Process Management Workshops,

Lecture Notes in Computer Science, Vol. 3812, pages 163–176, February 2006.

Abstract: *Most information systems log events (e.g., transaction logs, audit trails) to audit and monitor the processes they support. At the same time, many of these processes have been explicitly modeled. For example, SAP R/3 logs events in transaction logs and there are EPCs (Event-driven Process Chains) describing the so-called reference models. These reference models describe how the system should be used. The coexistence of event logs and process models raises an interesting question: "Does the event log conform to the process model and vice versa?". This paper demonstrates that there is not a simple answer to this question. To tackle the problem, we distinguish two dimensions of conformance: fitness (the event log may be the result of the process modeled) and appropriateness (the model is a likely candidate from a structural and behavioral point of view). Different metrics have been defined and a Conformance Checker has been implemented within the ProM Framework.*

3. Wil M. P. van der Aalst, Marlon Dumas, Chun Ouyang, Anne Rozinat, H. M. W. Verbeek. *Choreography Conformance Checking: An Approach based on BPEL and Petri Nets.* The Role of Business Processes in Service Oriented Architectures, 2006.

   Abstract: *Recently, languages such as BPEL and CDL have been proposed to describe the way services can interact from a behavioral perspective. 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 paper investigates a possible usage of behavioral service descriptions, namely as input for conformance checking. Conformance checking is the act of verifying whether one or more parties stick to the agreed-upon behavior by observing the actual behavior, e.g., the exchange of messages between all parties. This paper shows that it is possible to translate BPEL business protocols to Petri nets and to relate SOAP messages to transitions in the Petri net. As a result, Petri net-based conformance checking techniques can be used to quantify fitness (whether the observed behavior is possible in the business protocol) and appropriateness (whether the observed behavior "overfits" or "underfits" the business protocol). Moreover, non-conformance can be visualized to pinpoint deviations. The approach has been implemented in the context of the ProM n framework.*

4. Marco Alberti, Federico Chesani, Marco Gavanelli, Evelina Lamma, Paola Mello, Marco Montali, Sergio Storari, Paolo Torroni. *Computational Logic for Run-Time Verification of Web Services Choreographies: Exploiting the OCS-SI Tool.* WS-FM, pages 58–72, 2006.

   Abstract: *In this work, we investigate the feasibility of using a framework based on computational logic, and mainly defined in the context of Multi-Agent Systems for Global Computing (SOCS UE Project), for modeling choreographies of Web Services with respect to the conversational aspect. One of the fundamental motivations of using computational logic, beside its declarative and highly expressive nature, is given by its operational counterpart, that can provide a proof-theoretic framework able to verify the consistency of services designed in a cooperative and incremental manner. In particular, in this paper we show that suitable "Social Integrity Constraints", introduced in the SOCS social model, can be used for specifying global protocols at the choreography level. In this way, we can use a suitable tool, derived from the proof-procedure defined in the context of the SOCS project, to check at run-time whether a set of existing services behave in a conformant manner w.r.t. the defined choreography.*

5. Fabio Barbon, Paolo Traverso, Marco Pistore, Michele Trainotti. *Run-Time Monitoring of Instances and Classes of Web Service Compositions.* ICWS, pages 63–71, 2006.

   Abstract: *The run-time monitoring of web service compositions has been widely acknowledged as a significant and challenging problem. In this paper, we propose a novel solution to the problem of monitoring web services implemented in BPEL. We devise an architecture that clearly separates the business logic of a web service from its monitoring functionality. The architecture supports both "instance monitors" that deal with the execution of a single instance of BPEL process, as well as "class monitors" that report aggregated information about all the instances of a BPEL process. We also define a language for the specification of instance and class monitors. The language allows for specifying boolean, statistic, and time-related properties. Finally, we devise a technique for the automatic translation of all these kinds of monitors to Java programs.*

6. Patricia Bouyer, Pierre-Alain Reynier, Serge Haddad. *Extended Timed Automata and Time Petri Nets.* ACSD, pages 91–100, 2006.

   Abstract: *Timed Automata (TA) and Time Petri Nets (TPN) are two well-established formal models for real-time systems. Recently, a linear transformation of TA to TPNs preserving reachability properties and timed languages has been proposed, which does however not extend to larger classes of TA which would allow diagonal constraints or more general resets of clocks. Though these features do not add expressiveness, they yield exponentially more concise models. In this work, we propose two translations: one from extended TA to TPNs whose size is either linear or quadratic in the size of the original TA, depending on the features which are allowed; another one from a parallel composition of TA to TPNs, which is also linear. As a consequence, we get that TPNs are exponentially more concise than TA.*

7. Yan-ping Chen, Zeng-zhi Li, Qin-xue Jin, Chuang Wang. *Study on QoS Driven Web Services Composition.* Frontiers of WWW Research and Development - APWeb 2006, Lecture Notes on Computer Science, Vol. 3841, pages 702–707, Springer Verlag, 2006.

   Abstract: *Providing composed Web Services based on the QoS requirements of clients is still an urgent problem to be solved. In this paper, we try to solve this problem. Firstly, we enhanced the current WSDL to describe the QoS of services, and then gave a way to choose the proper pre-exist services based on their QoS.*

8. Mohan Baruwal Chhetri, Jian Lin, SukKeong Goh, Jian Ying Zhang, Ryszard Kowalczyk, Jun Yan. *A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation ofWeb Service Composition.* ASWEC, pages 90–99, 2006.

   Abstract: *Recent progress in the field of Web services has made it possible to integrate inter-organizational and heterogeneous services on the Web at runtime. If a user request cannot be satisfied by a single Web service, it is (or should be) possible to combine existing services in order to fulfill the request. However, there are several challenging issues that need to be addressed before this can be realized in the true sense. One of them is the ability to ensure end-to-end QoS of a Web service composition. There is a need for a SLA negotiation system which can ensure the autonomous QoS negotiation of Web service compositions irrespective of the application domain. In this paper we propose agent-based coordinated-negotiation architecture to ensure collective functionality, end-to-end QoS and the stateful coordination of complex services. We describe a prototype implementation to demonstrate how this architecture can be used in different application domains. We have also demonstrated how the negotiation system on the service provider's side can be implemented both as an agent based negotiation system and as a Web service based negotiation system.*

9. Gero Decker, Johannes Maria Zaha, Marlon Dumas. *Execution Semantics for Service Choreographies.* WS-FM, pages 163–177, 2006.

   Abstract: *A service choreography is a model of interactions in which a set of services engage to achieve a goal, seen from the perspective of an ideal observer that records all messages exchanged between these services. Choreographies have been put forward as a starting point for building service-oriented systems since they provide a global picture of the system's behavior. In previous work we presented a language for service choreography modeling targeting the early phases of the development lifecycle. This paper provides an execution semantics for this language in terms of a mapping to PI-calculus. This formal semantics provides a basis for analyzing choreographies. The paper reports on experiences using the semantics to detect unreachable interactions.*

10. Howard Foster, Sebastian Uchitel, Jeff Magee, Jeff Kramer. *LTSA-WS: a tool for model-based verification of web service compositions and choreography.* ICSE '06: Proceedings of the 28th international conference on Software engineering, pages 771–774, ACM, 2006.

    Abstract: *In this paper we describe a tool for a model-based approach to verifying compositions of web service implementations. The tool supports verification of properties created from design specifications and implementation models to confirm expected results from the viewpoints of both the designer and implementer. Scenarios are modeled in UML, in the form of Message Sequence Charts (MSCs), and then compiled into the Finite State Process (FSP) process algebra to concisely model the required behavior. BPEL4WS implementations are mechanically translated to FSP to allow an equivalence trace verification process to be performed.*

By providing early design verification and validation, the implementation, testing and deployment of web service compositions can be eased through the understanding of the behavior exhibited by the composition. The approach is implemented as a plug-in for the Eclipse development environment providing cooperating tools for specification, formal modeling, verification and validation of the composition process.

11. Raman Kazhamiakin, Paritosh K. Pandya, Marco Pistore. *Representation, Verification, and Computation of Timed Properties in Web Service Compositions.* ICWS, pages 497–504, 2006.

    Abstract: *In this paper we address the problem of qualitative and quantitative analysis of timing aspects of Web service compositions defined as a set of BPEL4WS processes. We introduce a formalism, called Web Service Timed State Transition Systems (WSTTS), to capture the timed behavior of the composite web services. We also exploit an interval temporal logic to express complex timed assumptions and requirements on the system's behavior. Building on top of this formalization, we provide techniques and tools for modelchecking BPEL4WS compositions against time-related requirements. We also present a symbolic algorithm that can be used to compute duration bounds of behavioral intervals that satisfy such requirements. We perform a preliminary experimental evaluation of our approach and tools with the help of an e-Government case study.*

12. Kyriakos Kritikos, Dimitris Plexousakis. *Semantic QoS Metric Matching.* ECOWS, pages 265–274, 2006.

    Abstract: *As the Web Service paradigm gains popularity for its promise to transform the way business is conducted, the number of deployed Web Services grows with a fast rate. While sophisticated semantic discovery mechanisms have been invented to overcome the UDDI's syntactic discovery solution in order to provide more recallable and precise results, the amount of functionally equivalent Web Services returned is still large. The solution to this problem is the description of the QoS non-functional aspect of Web Services. QoS encompasses the performance of Web Services and can be used as a discriminator factor for refining Web Service advertisement result lists. However, most scientific efforts presented so far are purely syntactic and are not capturing all aspects of QoS-based Web Service description leading to imprecise syntactic discovery mechanisms. This paper presents a novel, rich and extensible ontology-based approach for describing QoS of Web Services that complements OWL-S. It is shown that, by using this approach and by introducing the concept of semantic QoS metric matching, QoS-based syntactic matchmaking and selection algorithms are transformed to semantic ones leading to better results.*

13. Marina Mongiello, Daniela Castelluccia. *Modelling and verification of BPEL business processes.* Fourth Workshop on Model-Based Development of Computer-Based Systems, 2006.

    Abstract: *A business process is a complex Web service with functions provided by different Web services, which are already existing in Web and are dynamically integrated for granting a more complex business task. For this reason, business processes have become more and more diffuse in B2B and B2C domains, so that the importance of their activities asks for a high-level of reliability. Methods and tools for supporting automatic system verification and validation could be useful. Among the techniques of automatic verification, we choose model checking method, because we applied it efficiently for verification of a single Web service and in this paper we extend the area of application also in business processes. Descriptions of the behavior of a business process are coded using a standard language, BPEL4WS, that has broadly spread because it is able to describe a business process as both an executable process and an abstract process. Therefore, we model a BPEL description of a generic business process with a formal model and we formalize correctness properties about the reliability of the business process design. Also, we build a framework that performs automatic verification of formal models of business processes through NuSMV model checker. If there is a violation of correctness specifications, NuSMV provides counter-examples, so we can locate errors and effect right changes for correcting business process design.*

14. Nicole Oldham, Kunal Verma, Amit Sheth, Farshad Hakimpour. *Semantic WS-agreement partner selection.* WWW '06: Proceedings of the 15th international conference on World Wide Web, pages 697–706, ACM, 2006.

    Abstract: *In a dynamic service oriented environment it is desirable for service consumers and providers to*

offer and obtain guarantees regarding their capabilities and requirements. WS-Agreement defines a language and protocol for establishing agreements between two parties. The agreements are complex and expressive to the extent that the manual matching of these agreements would be expensive both in time and resources. It is essential to develop a method for matching agreements automatically. This work presents the framework and implementation of an innovative tool for the matching providers and consumers based on WS-Agreements. The approach utilizes Semantic Web technologies to achieve rich and accurate matches. A key feature is the novel and flexible approach for achieving user personalized matches.

15. Mohsen Rouached, Olivier Perrin, Claude Godart. *Towards Formal Verification of Web Service Composition.* Business Process Management, pages 257–273, 2006.

    Abstract: *Web services composition is an emerging paradigm for enabling application integration within and across organizational boundaries. Current Web services composition proposals, such as BPML, WS-BPEL, WSCI, and OWL-S, provide solutions for describing the control and data flows in Web service composition. However, such proposals remain at the descriptive level, without providing any kind of mechanisms or tool support for analysis and verification. Therefore, there is a growing interest for the verification techniques which enable designers to test and repair design errors even before actual running of the service, or allow designers to detect erroneous properties and formally verify whether the service process design does have certain desired properties. In this paper, we propose to verify Web services composition using an event driven approach. We assume Web services that are coordinated by a composition process expressed in WSBPEL and we use Event Calculus to specify the properties and requirements to be monitored.*

16. Yuan Yuan, Zhongjie Li, Wei Sun. *A Graph-Search Based Approach to BPEL4WS Test Generation.* ICSEA, 14 pages, 2006.

    Abstract: *Business Process Execution Language for Web Services (BPEL4WS) is a kind of concurrent programming languages with several special features that raise special challenges for verification and testing. This paper proposes a graph-search based approach to BPEL test case generation, which effectively deals with BPEL concurrency semantics. This approach defines an extension of CFG (Control Flow Graph) - BPEL Flow Graph (BFG) - to represent a BPEL program in a graphical model. Then concurrent test paths can be generated by traversing the BFG model, and test data for each path can be generated using a constraint solving method. Finally test paths and data are combined into complete test cases.*

17. Johannes Maria Zaha, Alistair P. Barros, Marlon Dumas, Arthur H. M. ter Hofstede. *Let's Dance: A Language for Service Behavior Modeling.* OTM Conferences (1), pages 145–162, 2006.

    Abstract: *In Service-Oriented Architectures (SOAs), software systems are decomposed into independent units, namely services, that interact with one another through message exchanges. To promote reuse and evolvability, these interactions are explicitly described right from the early phases of the development lifecycle. Up to now, emphasis has been placed on capturing structural aspects of service interactions. Gradually though, the description of behavioral dependencies between service interactions is gaining increasing attention as a means to push forward the SOA vision. This paper deals with the description of these behavioral dependencies during the analysis and design phases. The paper outlines a set of requirements that a language for modeling service interactions at this level should fulfill, and proposes a language whose design is driven by these requirements.*

Technical Reports and Manuals:

1. Siddharth Bajaj, Don Box, Dave Chappell, Francisco Curbera, Glen Daniels, Phillip Hallam-Baker, Maryann Hondo, Chris Kaler, Dave Langworthy, Anthony Nadalin, Nataraj Nagaratnam, Hemma Prafullchandra, Claus von Riegen, Daniel Roth, Jeffrey Schlimmer, Chris Sharp, John Shewchuk, Asir Vedamuthu, Ümit Yal cınalp, David Orchard . *Web Services Policy Framework (WS- Policy).* IBM, March 2006.

   Abstract: *The Web Services Policy Framework (WS-Policy) provides a general purpose model and corresponding syntax to describe the policies of a Web Service. WS-Policy defines a base set of constructs that can*

be used and extended by other Web services specifications to describe a broad range of service requirements and capabilities.

4 Publications in 2005

Articles in Refereed Journals:

1. Groenmo, R., Jaeger, M.C.. *Model-driven semantic Web service composition.* Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific, To Appear, December 2005.

   Abstract: *As the number of available Web services increases there is a growing demand to realise complex business processes by combining and reusing available Web services. The reuse and combination of services results in a composition of Web services that may also involve services provided in the Internet. With semantically described Web services, an automated matchmaking of capabilities can help identify suitable services. To address the need for semantically defined Web services, OWL-S and WSML have been proposed as competing semantic Web service languages. We show how the proposed semantic Web service languages can be utilized within a model-driven methodology for building composite Web services. In addition we combine the semantic-based discovery with the support for processing QoS requirements to apply a ranking or a selection of the candidates. The methodology describes a process which guides the developer through four phases, starting with the initial modelling, and ending with a new composite service that can be deployed and published to be consumed by other users.*

2. Ma, K.J.. *Web services: what's real and what's not?.* IT Professional, Vol. 7, Num. 2, pages 14–21, March 2005.

   Abstract: *The idea of abstracted, well-defined, and ubiquitously invokable services replacing proprietary interprocess communications has been a goal of system designers for a long time. The rise of Web services has led to a lot of misconceptions about how they can and cannot support the holy grail of a service-oriented architecture (SOA). This article seeks to put Web services in perspective, explaining their current capabilities and what industry can expect from them in the near term. It gives an overview of how technologies such as the Extensible Markup Language (XML), XML schemas, Extensible Stylesheet Language Transformations (XSLT), the Simple Object Access Protocol (SOAP), the Web Services Description Language (WSDL), and universal description, discovery, and integration (UDDI) fit into the equation for an SOA.*

3. Wil M. P. van der Aalst. *Business alignment: using process mining as a tool for Delta analysis and conformance testing.* Requir. Eng., Vol. 10, Num. 3, pages 198–211, 2005.

   Abstract: *Increasingly, business processes are being controlled and/or monitored by information systems. As a result, many business processes leave their "footprints" in transactional information systems, i.e., business events are recorded in so-called event logs. Process mining aims at improving this by providing techniques and tools for discovering process, control, data, organizational, and social structures from event logs, i.e., the basic idea of process mining is to diagnose business processes by mining event logs for knowledge. In this paper we focus on the potential use of process mining for measuring business alignment, i.e., comparing the real behavior of an information system or its users with the intended or expected behavior. We identify two ways to create and/or maintain the fit between business processes and supporting information systems: Delta analysis and conformance testing. Delta analysis compares the discovered model (i.e., an abstraction derived from the actual process) with some predefined processes model (e.g., the workflow model or reference model used to configure the system). Conformance testing attempts to quantify the "fit" between the event log and some predefined processes model. In this paper, we show that Delta analysis and conformance testing can be used to analyze business alignment as long as the actual events are logged and users have some control over the process.*

4. Schahram Dustdar, Wolfgang Schreiner. *A survey on web services composition.* IJWGS, Vol. 1, Num. 1, pages 1–30, 2005.

   Abstract: *Due to the web services' heterogeneous nature, which stems from the definition of several XML-based standards to overcome platform and language dependence, web services have become an emerging and promising technology to design and build complex inter-enterprise business applications out of single web-based software components. To establish the existence of a global component market, in order to enforce extensive software reuse, service composition experienced increasing interest in doing a lot of research effort. This paper discusses the urgent need for service composition, the required technologies to perform service* *composition. It also presents several different composition strategies, based on some currently existing composition platforms and frameworks, re-presenting first implementations of state-of the-art technologies, and gives an outlook to essential future research work.*

5. Papazoglou, M.P., van den Heuvel, W.-J.. *Web services management: a survey.* Internet Computing, IEEE, Vol. 9, Num. 6, pages 58–64, November/December 2005.

   Abstract: *Solutions based on service-oriented architectures are promising in that they leverage common services and enable collaborative business processes that cross organizational boundaries. However, because Web services applications can span multiple hosts, operating systems, languages, and enterprises, it's problematic to measure, control, and manage application availability and performance. In addition to discussing the relationship of Web services management to traditional distributed systems management, this survey explores various Web services management approaches and their underlying architectural concepts.*

6. James Pasley. *How BPEL and SOA Are Changing Web Services Development.* IEEE Internet Computing, Vol. 9, Num. 3, pages 60–67, 2005.

   Abstract: *As the use of Web services grows, organizations are increasingly choosing the Business Process Execution Language for modeling business processes within the Web services architecture. In addition to orchestrating organizations' Web services, BPEL's strengths include asynchronous message handling, reliability, and recovery. By developing Web services with BPEL in mind, organizations can implement aspects of the service-oriented architecture that might previously have been difficult to achieve.*

7. Tao Yu, Kwei-Jay Lin. *Service selection algorithms for Web services with end-to-end QoS constraints.* Inf. Syst. E-Business Management, Vol. 3, Num. 2, pages 103–126, 2005.

   Abstract: *Web services are new forms of Internet software that can be universally deployed and invoked using standard protocols. Services from different providers can be integrated into a composite service regardless of their locations, platforms, and/or execution speeds to implement complex business processes and transactions. In this paper, we study the end-to-end QoS issues of composite services by utilizing a QoS broker that is responsible for selecting and coordinating the individual service component. We design the service selection algorithms used by QoS brokers to construct the optimal composite service. The objective of the algorithms is to maximize the user-defined utility function value while meeting the end-to-end delay constraint. We propose two solution approaches to the service selection problem: the combinatorial approach, by modeling the problem as the Multiple Choice Knapsack Problem (MCKP), and the graph approach, by modeling the problem as the constrained shortest path problem in the graph theory. We study efficient solutions for each approach.*

Articles in Refereed Conferences:

1. Huang, H., Tsai, W.-T., Paul, R.. *Automated model checking and testing for composite Web services.* pages 300–307, May 2005.

   Abstract: *Web services form a new distributed computing paradigm. Collaborative verification and validation are important when Web services from different vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web services by model checking the process model of OWL-S (Web ontology language for Web services) and to validate them by the test cases automatically generated in the model checking process. We extend the BLAST, a model checker that handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

2. Rocco De Nicola, Gianluigi Ferrari, Ugo Montanari, Rosario Pugliese, Emilio Tuosto. *A Basic Calculus for Modelling Service Level Agreements.* International Conference on Coordination Models and Languages, LNCS, Vol. 3454, pages 33–48, Springer Verlag, April 2005.

   Abstract: *The definition of suitable abstractions and models for identifying, understanding and managing Quality of Service (QoS) constraints is a challenging issue of the Service Oriented Computing paradigm. In*

this paper we introduce a process calculus where QoS attributes are first class objects. We identify a minimal set of primitives that allow capturing in an abstract way the ability to control and coordinate services in presence of QoS constraints.

3. Bernd-Holger Schlingloff, Axel Martens, Karsten Schmidt. *Modeling and Model Checking Web Services.* Vol. 126, pages 3–26, Elsevier, March 2005.

   Abstract: *We give an overview on web services and the web service technology stack. We then show how to build Petri net models of web services formulated in the specification language BPEL4WS. We define an abstract correctness criterion for these models and study the automated verification according to this criterion. Finally, we relate correctness of web service models to the model checking problem for alternating temporal logics.*

4. Luciano Baresi, Sam Guinea. *Towards Dynamic Monitoring of WS-BPEL Processes.* ICSOC, pages 269–282, 2005.

   Abstract: *The intrinsic flexibility and dynamism of service-centric applications preclude their pre-release validation and demand for suitable probes to monitor their behavior at run-time. Probes must be suitably activated and deactivated according to the context in which the application is executed, but also according to the confidence we get on its quality. The paper supports the idea that significant data may come from very different sources and probes must be able to accommodate all of them. The paper presents: (1) an approach to specify monitoring directives, called monitoring rules, and weave them dynamically into the process they belong to; (2) a proxy-based solution to support the dynamic selection and execution of monitoring rules at run-time; (3) a user-oriented language to integrate data acquisition and analysis into monitoring rules.*

5. Alistair P. Barros, Marlon Dumas, Arthur H. M. ter Hofstede. *Service Interaction Patterns.* Business Process Management, pages 302–318, 2005.

   Abstract: *With increased sophistication and standardization of modeling languages and execution platforms supporting business process management (BPM) across traditional boundaries, has come the need for consolidated insights into their exploitation from a business perspective. Key technology developments in BPM bear this out, with several web services-related initiatives investing significant effort in the collection of compelling use cases to heighten the exploitation of BPM in multi-party collaborative environments. In this setting, we present a collection of patterns of service interactions which allow emerging web services functionality, especially that pertaining to choreography and orchestration, to be benchmarked against abstracted forms of representative scenarios. Beyond bilateral interactions, these patterns cover multilateral, competing, atomic and causally related interactions. Issues related to the implementation of these patterns using established and emerging web services standards, most notably BPEL, are discussed.*

6. Aysu Betin-Can, Tevfik Bultan, Xiang Fu. *Design for verification for asynchronously communicating Web services.* WWW, pages 750–759, 2005.

   Abstract: *We present a design for verification approach to developing reliable web services. We focus on composite web services which consist of asynchronously communicating peers. Our goal is to automatically verify properties of interactions among such peers. We propose a design pattern that eases the development of such web services and enables a modular, assume-guarantee style verification strategy. In the proposed design pattern, each peer is associated with a behavioral interface description which specifies how that peer will interact with other peers. Using these peer interfaces we automatically generate BPEL specifications to publish for interoperability. Assuming that the participating peers behave according to their interfaces, we verify safety and liveness properties about the global behavior of the composite web service during behavior verification. During interface verification, we check that each peer implementation conforms to its interface. Using the modularity in the proposed design pattern, we are able to perform the interface verification of each peer and the behavior verification as separate steps. Our experiments show that, using this modular approach, one can automatically and efficiently verify web service implementations.*

7. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani. *An approach for QoS-aware service composition based on genetic algorithms.* GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation, pages 1069–1075,

ACM, 2005.

Abstract: *Web services are rapidly changing the landscape of software engineering. One of the most interesting challenges introduced by web services is represented by Quality Of Service (QoS)–aware composition and late–binding. This allows to bind, at run–time, a service–oriented system with a set of services that, among those providing the required features, meet some non–functional constraints, and optimize criteria such as the overall cost or response time. In other words, QoS–aware composition can be modeled as an optimization problem. We propose to adopt Genetic Algorithms to this aim. Genetic Algorithms, while being slower than integer programming, represent a more scalable choice, and are more suitable to handle generic QoS attributes. The paper describes our approach and its applicability, advantages and weaknesses, discussing results of some numerical simulations.*

8. Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, Maria Luisa Villani. *QoS-Aware Replanning of Composite Web Services.* ICWS '05: Proceedings of the IEEE International Conference on Web Services, pages 121–129, IEEE Computer Society, 2005.

   Abstract: *Run-time service discovery and late-binding constitute some ofthe most challenging issues of service-oriented software engineering.For late-binding to be effective in the case of composite services,a QoS-aware composition mechanism is needed. This meansdetermining the set of services that, once composed, not only willperform the required functionality, but also will best contribute toachieve the level of QoS promised in Service Level Agreements (SLAs). However, QoS-aware composition relies on estimated QoS values and workflow execution paths previously obtained using a monitoring mechanism. At run-time, the actual QoS valuesmay deviate from the estimations, or the execution path may not bethe one foreseen. These changes could increase the risk of breakingSLAs and obtaining a poor QoS. Such a risk could be avoided byreplanning the service bindings of the workflow slice still to be executed. This paper proposes an approach to trigger and perform compositeservice replanning during execution. An evaluation has been performedsimulating execution and replanning on a set of composite serviceworkflows.*

9. Jorge Cardoso. *About the Data-Flow Complexity of Web Processes.* 6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility, pages 67–74, 2005.

   Abstract: *Abstract. Organizations are increasingly faced with the challenge of managing e-commerce and e-business applications involving Web services and Web processes. In some cases, Web processes design can be highly complex, due, for example, to the vast number of services carried out in global markets. High complexity in a process has several undesirable drawbacks, it may result in bad understandability, more errors, defects, and exceptions leading processes to need more time to develop, test, and maintain. Therefore, excessive complexity should be avoided. Flexibility and complexity are guiding principles in the design of business processes. In most business processes, flexibility and complexity are related inversely. Processes with a high complexity tend be less flexible, since it is more complicated to make changes to the process. In our previous work we have defined a metric to measure the control-flow complexity of Web processes. The major goal of this paper is to study the issues and establish the requirements for the development of a metric to analyze the data-flow complexity of Web processes.*

10. Samuele Carpineti, Cosimo Laneve, Paolo Milazzo. *BoPi - A Distributed Machine for Experimenting Web Services Technologies.* ACSD, pages 202–211, 2005.

    Abstract: *BoPi is a programming language with a runtime support that allows the distribution and the execution of programs over the network. The language is a process calculus with XML values and datatypes, and with a pattern matching mechanism for deconstructing values. The compiler gives a typesafe bytecode in the form of an XML document, that may be deployed on the network. What comes out is a simple, statically typed, and formally defined core BPEL language with a basic query mechanism supplied by patterns.*

11. Rocco De Nicola, Daniele Gorla, Rosario Pugliese. *Basic Observables for a Calculus for Global Computing.* ICALP, pages 1226–1238, 2005.

    Abstract: *We introduce a foundational language for modelling applications over global computers whose interconnection structure can be explicitly manipulated. Together with process distribution, mobility, remote operations and asynchronous communication through distributed data spaces, the language provides constructs for explicitly modelling inter-node connections and for dynamically establishing and removing them.*

For the proposed language, we define natural notions of extensional observations and study their closure under operational reductions and/or language contexts to obtain barbed congruence and may testing equivalence. For such equivalences, we provide alternative characterizations in terms of a labelled bisimulation and a trace equivalence that can be used for actual proofs.

12. Casey K. Fung, Patrick C. K. Hung, Guijun Wang, Richard C. Linger, Gwendolyn H. Walton. *A Study of Service Composition with QoS Management.* ICWS, pages 717–724, 2005.
Abstract: *Quality of service (QoS) management in compositions of services requires careful consideration of QoS characteristics of the services and effective QoS management in their execution. A Web service is a software system that supports interoperable application-to-application interaction over the Internet. Web services are based on a set of XML standards such as simple object access protocol (SOAP). The interactions of SOAP messages between Web services form the theoretical model of SOAP message exchange patterns (MEP). Web Services Business Process Execution Language (WSBPEL) defines an interoperable integration model that facilitates automated process integration in intra- and inter-corporate environments. A service-level agreement (SLA) is a formal contract between a Web services requestor and provider guaranteeing quantifiable issues at defined levels only through mutual concessions. Based on a prior research work on message detail record (MDR), this paper further proposes a SOAP message tracking model for supporting QoS end-to-end management in the context of WSBPEL and SLA. This paper motivates the study of QoS management in a Web service composition framework with the evolution of a distributed toolkit in an industrial setting.*

13. Dan Hirsch, Emilio Tuosto. *SHReQ: Coordinating Application Level QoS.* SEFM, pages 425–434, 2005.
Abstract: *We present SHReQ, a formal framework for specifying systems that handle abstract high-level QoS requirements which are becoming more and more important for service oriented computing. SHReQ combines Synchronised Hyperedge Replacement (SHR) with constraint-semirings. SHR is a (hyper)graph rewriting mechanism for modelling evolution of systems. The novelty of the approach relies on the synchronisation mechanism which is based on constraint-semirings, algebraic structures that provide both the mathematics for multi-criteria QoS and the synchronisation policies underlying the SHR mechanism.*

14. Hai Huang, Wei-Tek Tsai, Raymond Paul, Yinong Chen. *Automated Model Checking and Testing for Composite Web Services..* 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2005), pages 300–307, IEEE Computer Society, 2005.
Abstract: *Web Services form a new distributed computing paradigm. Collaborative verification and validation are important when Web Services from differen vendors are integrated together to carry out a coherent task. This paper presents a new approach to verify Web Services by model checking the proces model of OWL-S (Web Ontology Language for Web Services) and to validate them by the test cases automatically generated in the model checking process. We extend the BLAST, a model checker tha handles control flow model naturally, to handle the concurrency in OWL-S. We also propose enhancement in OWL-S and PDDL (Planning Domain Definition Language) to facilitate the automated test case generation. Experiments on realistic examples are provided to illustrate the process.*

15. Michael C. Jaeger, Gero Mühl, Sebastian Golze. *QoS-Aware Composition of Web Services: A Look at Selection Algorithms.* ICWS, pages 807–808, 2005.
Abstract: *When a composition of Web services is designed, available services are put together to form a defined flow of executions. In a discovery process, a trader proposes available Web services as potential candidates. In a succeeding selection, for each task a trader chooses one candidate to form the optimal composition due to selection criteria. This paper discusses how the selection can consider different quality-of-service (QoS) categories to determine the most suitable candidates for the composition. If more than one category is used for optimisation, a multi-dimensional optimisation problem arises. This mentions similarities to similar combinatorial problems. Then, possible solutions are proposed and their performance is evaluated.*

16. Dimka Karastoyanova, Alejandro Houspanossian, Mariano Cilia, Frank Leymann, Alejandro P. Buchmann. *Extending BPEL for Run Time Adaptability.* EDOC, pages 15–26, 2005.

    Abstract: *The existing Web Service Flow (WS-flow) technologies enable both static and dynamic binding of participating Web services (WSs) on the process model level. Adaptability on per-instance basis is not sufficiently supported and therefore must be addressed to improve process flexibility upon changes in the environment. Ad-hoc process instance changes can be enabled by swapping participating WS instances, by modifying port Types of the partners to be invoked, and by changing process logic. In this work we address the problem of dynamic binding of WSs to WS-flow instances at run time, i.e. the ability to exchange a WS instance participating in a WS-flow instance with an alternative one. The problem is additionally complicated by the fact that the execution of a process depends on its deployment. We describe the "find and bind" mechanism, and we show its representation as a BPEL extension. We discuss the benefits that could be gained and the disadvantages it brings in. The mechanism extends and improves the existing process technologies. It facilitates a precisely controlled policy-based selection of WSs at run time and also provides for process instance repair, while maintaining simplicity. We also discuss a prototypical implementation of the presented functionality.*

17. Khaled Mahbub, George Spanoudakis. *Run-time Monitoring of Requirements for Systems Composed of Web-Services: Initial Implementation and Evaluation Experience.* ICWS, pages 257–265, 2005.

    Abstract: *This paper describes a framework supporting the runtime monitoring of requirements for systems implemented as compositions of Web-services specified in BPEL. The requirements that can be monitored are specified in event calculus. The paper presents an overview of the framework and describes the architecture and implementation of a tool that we have developed to operationalise it. It also presents the results of a preliminary experimental evaluation of the framework.*

18. Mike P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann, Bernd J. Krämer. *05462 Service-Oriented Computing: A Research Roadmap.* Service Oriented Computing, 2005.

    Abstract: *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.*

19. Florian Rosenberg, Schahram Dustdar. *Business Rules Integration in BPEL - A Service-Oriented Approach.* CEC, pages 476–479, 2005.

    Abstract: *Business rules change quite often. These changes cannot be handled efficiently by representing business rules embedded in the source code of the business logic. Efficient handling of rules that govern ones business is one factor for success. That is where business rules engines play an important role. The service-oriented computing paradigm is becoming more and more popular. Services offered by different providers, are composed to new services by using Web service composition languages such as BPEL. Such process-based composition languages lack the ability to use business rules managed by different business rules engines in the composition process. In this paper, we propose an approach on how to use and integrate business rules in a service-oriented way into BPEL.*

20. Wei-Tek Tsai, Yinong Chen, Raymond A. Paul. *Specification-Based Verification and Validation of Web Services and Service-Oriented Operating Systems.* WORDS, pages 139–147, 2005.

    Abstract: *Service-Oriented Architecture (SOA) and Web Services (WS) have received significant attention recently. Even though WS are based on open standards and support software interoperability, but the trustworthy issues of WS has actually limited the growth of WS applications as organizations do not trust those WS developed by other vendors and at the same time they do not have access to the source code. This paper addressed this issue by proposing several solutions including specification-based verification and validation, collaborative testing, and group testing. The key concept is that it is possible to provide a comprehensive evaluation of WS even if their source code is not available.*

Publications in Refereed Workshops:

1. S. Nakajima. *Model-Checking Behavioral Specification of BPEL Applications.* Proceedings of the International Workshop on Web Languages and Formal Methods, WLFM 2005, 2005.

   Abstract: *To provide a framework to compose lots of specialised services flexibly, BPEL is proposed to describe Web service flows. Since the Web service flow description is basically a distributed collaboration, writing correct programs in BPEL is not easy. Verifying BPEL program prior to its execution is essential. This paper proposes a method to extract the behavioral specification from a BPEL appliation program and to analyze it by using the SPIN model checker. With the adequate abstraction method and support for DPE, the method can analyze all the four example cases in the BPEL standard document.*

Technical Reports and Manuals:

1. Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, Ming Xu. *Web Services Agreement Specification (WS-Agreement).* Grid Resource Allocation Agreement Protocol, September 2005.

   Abstract: *This document describes Web Services Agreement Specification (WS-Agreement), a Web Services protocol for establishing agreement between two parties, such as between a service provider and consumer, using an extensible XML language for specifying the nature of the agreement, and agreement templates to facilitate discovery of compatible agreement parties. The specification consists of three parts which may be used in a composable manner: a schema for specifying an agreement, a schema for specifying an agreement template, and a set of port types and operations for managing agreement life-cycle, including creation, expiration, and monitoring of agreement states.*

2. The OASIS Group. *Quality Model for Web Services.* The Oasis Group, September 2005.

   Abstract: *The purpose of this document is to provide a model for Web services quality management and quality factors in the process of developing and using Web services. We define the consistent and systematic conceptual model of Web services quality, which may be used by intimate associates, i.e. stakeholders, developers, service providers, and customers of Web services.*

3. The OMG Group. *UML$^{\text{TM}}$ Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms.* Num. ptc/2005-05-02, The OMG Group, May 2005.

4. I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich . *The Open Grid Services Architecture, Version 1.0.* ANL, IBM, Fujitsu, NeSC, CCLRC-RAL, UVa, Hitachi, Intel, HP, 2005.

   Abstract: *Successful realization of the Open Grid Services Architecture (OGSA) vision of a broadly applicable and adopted framework for distributed system integration, virtualization, and management requires the definition of a core set of interfaces, behaviors, resource models, and bindings. This document, produced by the OGSA working group within the Global Grid Forum (GGF), provides a first version of this OGSA definition. The document focuses on requirements and the scope of important capabilities required*

to support Grid systems and applications in both e-science and e-business. The capabilities described are Execution Management, Data, Resource Management, Security, Self-Management, and Information. The description of the capabilities is at a high-level and includes, to some extent, the interrelationships between the capabilities.

5. The Oasis Group. *Summary of Quality Model for Web Services.* The Oasis Group, 2005.

5 Publications in 2004

Articles in Refereed Journals:

1. Neal Leavitt. *Are Web Services Finally Ready to Deliver?.* IEEE Computer, Vol. 37, Num. 11, pages 14–18, November 2004.

   Abstract: *Web services, in brief, are a framework of software technologies designed to support interoperable machine-to-machine interaction over a network. Companies on different systems can use Web services to exchange information online with business partners, customers, and suppliers. Various standards organizations and industry consortia are developing Web services specifications without a unifying authority. Organizations such as the World Wide Web Consortium (W3C), the Organization for the Advancement of Structured Information Standards (OASIS), the Liberty Alliance Project, and the Web Services Interoperability Organization (WS-I) have developed or reviewed numerous standards. A primary goal of Web services is to unlock a new generation of e-commerce applications.*

2. Xiulan Yu, Long Zhang, Ying Li, Ying Chen. *WSCE: a flexible Web Service Composition Environment.* Web Services, 2004. Proceedings. IEEE International Conference on, To Appear, July 2004.

   Abstract: *In this paper, we propose the concepts of virtual partner and inspector into the Web services composition. Virtual partner, as an IT level concept, is a Web service (pseudo Web service) using the same interface with the actual partner but different binding message. A virtual partner can be invoked directly by a business process described by BPEL, so that the BPEL programmer can test both application's functionality and non functionality performance early in the development cycle to avoid any problems in the final runtime, or test the selection of their partners in business level design. The IT virtual partners provide developers with a range of the techniques which let them explore every aspect of their program. Inspector is proposed when using the third-party process engine. An inspector itself is also a Web service. The programmer can register any required output information in it. The IT virtual partner and the inspector concepts have been integrated in our WSCE, a flexible Web Services Composition Environment for a business process. WSCE is a prototype of autonomic modeling and simulation environment. With the help of a third-party BPEL engine, it provides programmer with concepts and tools to facilitate business process programming.*

3. Liangzhao Zeng, Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., Chang, H.. *QoS-aware middleware for Web services composition.* Software Engineering, IEEE Transactions on, Vol. 30, Num. 5, pages 311–327, May 2004.

   Abstract: *The paradigmatic shift from a Web of manual interactions to a Web of programmatic interactions driven by Web services is creating unprecedented opportunities for the formation of online business-to-business (B2B) collaborations. In particular, the creation of value-added services by composition of existing ones is gaining a significant momentum. Since many available Web services provide overlapping or identical functionality, albeit with different quality of service (QoS), a choice needs to be made to determine which services are to participate in a given composite service. This paper presents a middleware platform which addresses the issue of selecting Web services for the purpose of their composition in a way that maximizes user satisfaction expressed as utility functions over QoS attributes, while satisfying the constraints set by the user and by the structure of the composite service. Two selection approaches are described and compared: one based on local (task-level) selection of services and the other based on global allocation of tasks to services using integer programming.*

4. Andrew W. Cooke, Alasdair J. G. Gray, Werner Nutt, James Magowan, Manfred Oevers, Paul Taylor, Roney Cordenonsi, Rob Byrom, Linda Cornwall, Abdeslem Djaoui, Laurence Field, Steve Fisher, Steve Hicks, Jason Leake, Robin Middleton, Antony J. Wilson, Xiaomei Zhu, Norbert Podhorszki, Brian A. Coghlan, Stuart Kenny, David O'Callaghan, John Ryan. *The Relational Grid Monitoring Architecture: Mediating Information about the Grid.* J. Grid Comput., Vol. 2, Num. 4, pages 323–339, 2004.

   Abstract: *We have developed and implemented the Relational Grid Monitoring Architecture (R-GMA) as part of the DataGrid project, to provide a flexible information and monitoring service for use by other* *middleware components and applications. R-GMA presents users with a virtual database and mediates queries posed at this database: users pose queries against a global schema and R-GMA takes responsibility for locating relevant sources and returning an answer. R-GMAs architecture and mechanisms are general and can be used wherever there is a need for publishing and querying information in a distributed environment. We discuss the requirements, design and implementation of R-GMA as deployed on the DataGrid testbed. We also describe some of the ways in which R-GMA is being used.*

5. Mariya Koshkina, Franck van Breugel. *Modelling and verifying web service orchestration by means of the concurrency workbench.* SIGSOFT Softw. Eng. Notes, Vol. 29, Num. 5, pages 1–10, ACM, 2004.

   Abstract: *Verification techniques like model checking, preorder checking and equivalence checking are shown to be relevant to web service orchestration. The Concurrency Workbench of the New Century (CWB) is a verification tool that supports these verification techniques. By means of the Process Algebra Compiler (PAC), the CWB is modified to support the BPE-calculus. The BPE-calculus is a small language, based on BPEL4WS, to express web service orchestration. Both the syntax and the semantics of the BPE-calculus are formally defined. These are subsequently used as input for the PAC. As output, the PAC produces modules that are incorporated into the CWB so that it supports the BPE-calculus and, hence, provides a verification tool for web service orchestration.*

6. Milanovic, N., Malek, M.. *Current solutions for Web service composition.* Internet Computing, IEEE, Vol. 8, Num. 6, pages 51–59, November/December 2004.

   Abstract: *Web service composition lets developers create applications on top of service-oriented computing's native description, discovery, and communication capabilities. Such applications are rapidly deployable and offer developers reuse possibilities and users seamless access to a variety of complex services. There are many existing approaches to service composition, ranging from abstract methods to those aiming to be industry standards. The authors describe four key issues for Web service composition.*

7. Marco Pistore, Marco Roveri, Paolo Busetta. *Requirements-Driven Verification of Web Services.* Electr. Notes Theor. Comput. Sci., Vol. 105, pages 95–108, 2004.

   Abstract: *We propose a requirements-driven approach to the design and verification of Web services. The proposed methodology starts from a requirements model, which defines a business domain at a "strategic" level, describing the participating actors, their mutual dependencies, goals, requirements, and expectations. This business requirements model is then refined into a business process model. In this refinement, definitions of the processes carried out by the actors of the domain are added to the model in the form of BPEL4WS code. We show how to exploit model checking techniques for the verification of the specification, both at the requirements and at the process level. At the requirements level, model checking is used to validate the specification against a set of queries specified by the designer; at the process level, it is used to verify if the BPEL4WS processes satisfy the constraints described in the requirements model.*

Articles in Refereed Conferences:

1. Foster, H., Uchitel, S., Magee, J., Kramer, J.. *Compatibility verification for Web service choreography.* pages 738–741, July 2004.

   Abstract: *In this paper we discuss a model-based approach to verifying process interactions for coordinated Web service compositions. The approach uses finite state machine representations of Web service orchestrations and assigns semantics to the distributed process interactions. The move towards implementing Web service compositions by multiple interested parties as a form of distributed system architecture motivates the need for supporting compatibility verification of activities and transactions in all the processes. The described approach is supported by a suite of cooperating tools for specification, formal modeling and providing verification results from orchestrated Web service interactions.*

2. Rohit Aggarwal, Kunal Verma, John A. Miller, William Milnor. *Constraint Driven Web Service Composition in METEOR-S.* IEEE SCC, pages 23–30, 2004.

   Abstract: *Creating Web processes using Web service technology gives us the opportunity for selecting new*

services which best suit our need at the moment. Doing this automatically would require us to quantify our criteria for selection. In addition, there are challenging issues of correctness and optimality. We present a Constraint Driven Web Service Composition tool in METEOR-S, which allows the process designers to bind Web Services to an abstract process, based on business and process constraints and generate an executable process. Our approach is to reduce much of the service composition problem to a constraint satisfaction problem. It uses a multi-phase approach for constraint analysis. This work was done as part of the METEOR-S framework, which aims to support the complete lifecycle of semantic Web processes.

3. Jesus Arias-Fisteus, Luis Sanchez Fernandez, Carlos Delgado Kloos. *Formal Verification of BPEL4WS Business Collaborations.* E-Commerce and Web Technologies, 5th International Conference, EC-Web 2004, Proceedings, Lecture Notes in Computer Science, Vol. 3182, pages 76–85, Springer Verlag, 2004.

   Abstract: *Web services are a very appropriate communication mechanism to perform distributed business processes among several organisations. These processes should be reliable, because a failure in them can cause high economic losses. To increase their reliability at design time, we have developed VERBUS, a framework for the formal verification of business processes. VERBUS can automatically translate business process definitions to specifications verifiable in several available tools. It is based on a modular and extensible architecture: new process definition languages and verification tools can be added easily to the framework. The prototype of VERBUS presented in this work can verify BPEL4WS process specifications, by translating them to Promela. The Promela specifications are verified with the well known model checker Spin. In this paper we describe the general architecture of VERBUS and how BPEL4WS specifications are translated and verified. The explanation is completed by describing what types of properties can be verified and providing an overview of the implementation.*

4. Tevfik Bultan, Xiang Fu, Jianwen Su. *Tools for Automated Verification of Web Services.* ATVA, pages 8–10, 2004.

5. Vikas Deora, Jianhua Shao, Gareth Shercliff, Patrick J. Stockreisser, W. A. Gray, N. J. Fiddian. *Incorporating QoS Specifications in Service Discovery.* WISE Workshops, pages 252–263, 2004.

   Abstract: *In this paper, we extend the current approaches to service discovery in a service oriented computing environment, such as Web Services and Grid, by allowing service providers and consumers to express their promises and requirements for quality of service (QoS). More specifically, we allow service providers to advertise their services in an extended DAML-S that supports quality specifications, and we allow service consumers to request services by stating required quality levels. We propose a model here for incorporating QoS specifications and requirements in service discovery, and describe how matchmaking between advertised and requested services based on functional as well as quality requirements is supported in our model.*

6. Roozbeh Farahbod, Uwe Glässer, Mona Vajihollahi. *Specification and Validation of the Business Process Execution Language for Web Services.* Abstract State Machines, pages 78–94, 2004.

   Abstract: *We formally define an abstract executable semantics for the Business Process Execution Language for Web Services in terms of a distributed ASM. The goal of this work is to support the design and standardization of the language. There is a need for formalism. It will allow us to not only reason about the current specification and related issues, but also uncover issues that would otherwise go unnoticed. Empirical deduction is not sufficient. - Issue 42, OASIS WSBPEL TC. The language definition assumes an infrastructure for running Web services on some asynchronous communication architecture. A business process is built on top of a collection of Web services performing continuous interactions with the outside world by sending and receiving messages over a communication network. The underlying execution model is characterized by its concurrent and reactive behavior making it particularly difficult to predict dynamic system properties with a sufficient degree of detail and precision under all circumstances.*

7. Xiang Fu, Tevfik Bultan, Jianwen Su. *Analysis of Interacting BPEL Web Services.* WWW'04: Proceedings of the 13th International Conference on World Wide Web, pages 621–630, ACM Press, 2004.

Abstract: *This paper presents a set of tools and techniques for analyzing interactions of composite web services which are specified in BPEL and communicate through asynchronous XML messages. We model the interactions of composite web services as conversations, the global sequence of messages ex- changed by the web services. As opposed to earlier work, our tool-set handles rich data manipulation via XPath expressions. This allows us to verify designs at a more detailed level and check properties about message content. We present a framework where BPEL specifications of web ser- vices are translated to an intermediate representation, followed by the translation of the intermediate representation to a verification language. As an intermediate representation we use guarded automata augmented with unbounded queues for incoming messages, where the guards are expressed as XPath expressions. As the target verification language we use Promela, input language of the model checker SPIN. Since SPIN model checker is a finite-state verification tool we can only achieve partial verification by fixing the sizes of the input queues in the translation. We propose the concept of synchronizability to address this problem. We show that if a composite web service is synchronizable, then its conversation set remains same when asynchronous communication is replaced with synchronous communication. We give a set of sufficient conditions that guarantee synchronizability and that can be checked statically. Based on our synchronizability results, we show that a large class of composite web services with unbounded input queues can be completely verified using a finite state model checker such as SPIN.*

8. Xiang Fu, Tevfik Bultan, Jianwen Su. *Realizability of Conversation Protocols With Message Contents.* ICWS, 2004.

   Abstract: *A conversation protocol is a top-down specification framework which specifies desired global behaviors of a web service composition. In our earlier work [A formalism for specification and verification of reactive electronic services] we studied the problem of realizability, i.e., given a conversation protocol, can a web service composition be synthesized to generate behaviors as specified by the protocol. Several sufficient realizability conditions were proposed in [A formalism for specification and verification of reactive electronic services] to ensure realizability. Conversation protocols studied in [A formalism for specification and verification of reactive electronic services], however, are essentially abstract control flows without data semantics. This paper extends the work in [A formalism for specification and verification of reactive electronic services] and achieves more accurate analysis by considering data semantics. To overcome the state-space explosion caused by the data content, we propose a symbolic analysis technique for each realizability condition. In addition, we show that the analysis of the autonomy condition can be done using an iterative refinement approach.*

9. Xiang Fu, Tevfik Bultan, Jianwen Su. *Model checking XML manipulating software.* ISSTA, pages 252–262, 2004.

   Abstract: *The use of XML as the de facto data exchange standard has allowed integration of heterogeneous web based software systems regardless of implementation platforms and programming languages. On the other hand, the rich tree-structured data representation, and the expressive XML query languages (such as XPath) make formal specification and verification of software systems that manipulate XML data a challenge. In this paper, we present our initial efforts in automated verification of XML data manipulation operations using the SPIN model checker. We present algorithms for translating (bounded) XML data and XPath expressions to Promela, the input language of SPIN. The techniques presented in this paper constitute the basis of our Web Service Analysis Tool (WSAT) which verifies LTL properties of composite web services.*

10. Gang Huang, Meng Wang, Liya Ma, Ling Lan, Tiancheng Liu, Hong Mei. *Towards Architecture Model based Deployment for Dynamic Grid Services.* IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East'04), 2004.

    Abstract: *The deployment of grid services should make the services, including those to be deployed and those already deployed, operate with desired functionalities and qualities. The critical challenge in the deployment is that many technical and nontechnical factors have to be taken into account, such as performance, reliability, utilization, operating cost, incomes, and so on. Since the factors change continuously, some deployed services may have to be redeployed for guaranteeing their functionalities and qualities. This position paper presents an approach to the deployment and redeployment of grid services based on software architecture models. In this approach, all services in a grid consist in a software architecture, which represents the services, their relationships and other factors in a global, understandable and easy-to-use way.*

To demonstrate the approach, a visual tool for deploying services onto a set of popular grid infrastructures, including J2EE application servers and BPEL engines, with the help of software architectures is developed.

11. Michael C. Jaeger, Gregor Rojec-Goldmann, Gero Mühl. *QoS Aggregation for Web Service Composition using Workflow Patterns.* EDOC, pages 149–159, 2004.

    Abstract: *Contributions in the field of Web services have identified that (a) finding matches between semantic descriptions of advertised and requested services and (b) nonfunctional characteristics - the quality of service (QoS) - are the most crucial criteria for composition of Web services. A mechanism is introduced that determines the QoS of a Web service composition by aggregating the QoS dimensions of the individual services. This allows to verify whether a set of services selected for composition satisfies the QoS requirements for the whole composition. The aggregation performed builds upon abstract composition patterns, which represent basic structural elements of a composition, like sequence, loop, or parallel execution. This work focusses on workflow management environments. We define composition patterns that are derived from Van der Aalst's et al. comprehensive collection of workflow patterns. The resulting aggregation schema supports the same structural elements as found in workflows. Furthermore, the aggregation of several QoS dimensions is discussed.*

12. Jinghai Rao, Xiaomeng Su. *A Survey of Automated Web Service Composition Methods.* SWSWPC, pages 43–54, 2004.

    Abstract: *In todays Web, Web services are created and updated on the fly. Its already beyond the human ability to analysis them and generate the composition plan manually. A number of approaches have been proposed to tackle that problem. Most of them are inspired by the researches in cross-enterprise workflow and AI planning. This paper gives an overview of recent research efforts of automatic Web service composition both from the workflow and AI planning research community.*

13. Gwen Salaün, Lucas Bordeaux, Marco Schaerf. *Describing and Reasoning on Web Services using Process Algebra.* ICWS, 2004.

    Abstract: *We argue that essential facets of web services, and especially those useful to understand their interaction, can be described using process-algebraic notations. Web service description and execution languages such as BPEL are essentially process description languages; they are based on primitives for behaviour description and message exchange which can also be found in more abstract process algebras. One legitimate question is therefore whether the formal approach and the sophisticated tools introduced for process algebra can be used to improve the effectiveness and the reliability of web service development. Our investigations suggest a positive answer, and we claim that process algebras provide a very complete and satisfactory assistance to the whole process of web service development. We show on a case study that readily available tools based on process algebra are effective at verifying that web services conform their requirements and respect properties. We advocate their use both at the design stage and for reverse engineering issues. More prospectively, we discuss how they can be helpful to tackle choreography issues.*

14. James Skene, D. Davide Lamanna, Wolfgang Emmerich. *Precise Service Level Agreements.* ICSE, pages 179–188, 2004.

    Abstract: *SLAng is an XML language for defining service level agreements, the part of a contract between the client and provider of an Internet service that describes the quality attributes that the service is required to possess. We define the semantics of SLAng precisely by modelling the syntax of the language in UML, then relating the language model to a model that describes the structure and behaviour of services. The presence of SLAng elements imposes behavioural constraints on service elements, and the precise definition of these constraints using OCL constitutes the semantic description of the language. We use the semantics to define a notion of SLA compatibility, and an extension to UML that enables the modelling of service situations as a precursor to analysis, implementation and provisioning activities.*

15. Andreas Wombacher, Peter Fankhauser, Erich J. Neuhold. *Transforming BPEL into Annotated Deterministic Finite State Automata for Service Discovery.* ICWS, pages 316–323, 2004.

    Abstract: *Web services advocate loosely coupled systems, although current loosely coupled applications are*

limited to stateless services. The reason for this limitation is the lack of a method supporting matchmaking of state dependent services exemplarily specified in BPEL. In particular, the sender's requirement that the receiver must support all possible messages sent at a certain state are not captured by models currently used for service discovery. Annotated deterministic finite state automata provide this expressiveness. In this paper the transformation of a local process specification given in BPEL to annotated deterministic finite state automata is presented.

16. Tao Yu, Kwei-Jay Lin. *Service Selection Algorithms for Web Services with End-to-End QoS Constraints.* CEC, pages 129–136, 2004.

    Abstract: *Web services are new forms of Internet software that can be universally deployed and invoked using standard protocol. Services from different providers can be integrated to provide composite services. In this paper, we study the end-to-end QoS issues of composite service by utilizing a QoS broker that is responsible for coordinating the individual service component to meet the quality constraint. We design the service selection algorithms used by QoS brokers to meet end-to-end QoS constraints. The objective of the algorithms is to maximize the user-defined utility while meeting the end-to-end delay constraint. We model the problem as the Multiple Choice Knapsack Problem (MCKP) and provide efficient solutions. The algorithms are tested for their performance.*

6 Publications in 2003

Articles in Refereed Journals:

1. W. van der Aalst. *Don't go with the flow: Web services composition standards exposed.* IEEE Intelligent Systems, To Appear, Jan/Feb 2003.

   Abstract: *The recently released Business Process Execution Language for Web Services (BPEL4WS) is said to combine the best of other standards for web services composition such as WSFL from IBM and XLANG of Microsoft. BPEL4WS allows for a mixture of block structured and graph structured process models thus making the language expressive at the price of being complex. Although BPEL4WS is not such a bad proposal by itself, it is remarkable how much attention this standard receives while the more fundamental issues and problems such as semantics, expressiveness, and adequacy do not get the attention they deserve. Having a standard is a very good idea. However, there are too many of them and most of them die before becoming mature. A simple indicator of this development is the increasing length of acronyms: PDL, XPDL, BPSS, EDOC, BPML, WSDL, WSCI, ebXML, and BPEL4WS are just some of the acronyms referring to various standards in the domain. Another problem is that these languages typically have no clearly defined semantics. The only way to overcome these problems is to critically evaluate the so-called standards for web services composition, i.e., Don't go with the flow!*

2. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski, Alistair P. Barros. *Workflow Patterns.* Distributed and Parallel Databases, Vol. 14, Num. 1, pages 5–51, 2003.

   Abstract: *Differences in features supported by the various contemporary commercial workflow management systems point to different insights of suitability and different levels of expressive power. The challenge, which we undertake in this paper, is to systematically address workflow requirements, from basic to complex. Many of the more complex requirements identified, recur quite frequently in the analysis phases of workflow projects, however their implementation is uncertain in current products. Requirements for workflow languages are indicated through workflow patterns. In this context, patterns address business requirements in an imperative workflow style expression, but are removed from specific workflow languages. The paper describes a number of workflow patterns addressing what we believe identify comprehensive workflow functionality. These patterns provide the basis for an in-depth comparison of a number of commercially availablework flow management systems. As such, this paper can be seen as the academic response to evaluations made by prestigious consulting companies. Typically, these evaluations hardly consider the workflow modeling language and routing capabilities, and focus more on the purely technical and commercial aspects.*

3. Gian Luigi Ferrari, Stefania Gnesi, Ugo Montanari, Marco Pistore. *A model-checking verification environment for mobile processes.* ACM Trans. Softw. Eng. Methodol., Vol. 12, Num. 4, pages 440–473, 2003.

   Abstract: *This article presents a semantic-based environment for reasoning about the behavior of mobile systems. The verification environment, called HAL, exploits a novel automata-like model that allows finite-state verification of systems specified in the π-calculus. The HAL system is able to interface with several efficient toolkits (e.g. model-checkers) to determine whether or not certain properties hold for a given specification. We report experimental results on some case studies.*

4. Xavier Franch, Juan Pablo Carvallo. *Using Quality Models in Software Package Selection.* IEEE Software, Vol. 20, Num. 1, pages 34–41, 2003.

   Abstract: *The growing importance of commercial off-the-shelf software packages requires adapting some software engineering practices, such as requirements elicitation and testing, to this emergent framework. Also, some specific new activities arise, among which selection of software packages plays a prominent role. All the methodologies that have been proposed recently for choosing software packages compare user requirements with the packages' capabilities. There are different types of requirements, such as managerial, political, and, of course, quality requirements. Quality requirements are often difficult to check. This is partly due to their nature, but there is another reason that can be mitigated, namely the lack of structured and widespread descriptions of package domains (that is, categories of software packages such as ERP systems, graphical or data structure libraries, and so on). This absence hampers the accurate description of software* *packages and the precise statement of quality requirements, and consequently overall package selection and confidence in the result of the process. Our methodology for building structured quality models helps solve this drawback.*

5. Alexander Keller, Heiko Ludwig. *The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services.* J. Network Syst. Manage., Vol. 11, Num. 1, 2003.
   Abstract: *We describe a novel framework for specifying and monitoring Service Level Agreements (SLA) for Web Services. SLA monitoring and enforcement becomes increasingly important in a Web Service environment where enterprises rely on services that may be subscribed dynamically and on demand. For economic and practical reasons, we want an automated provisioning process for both the service itself as well as the SLA management system. It measures and monitors the QoS parameters, checks the agreed-upon service levels, and reports violations to the authorized parties involved in the SLA management process. The Web Service Level Agreement (WSLA) framework, our approach to these issues, is targeted at defining and monitoring SLAs for Web Services. Although WSLA has been designed for a Web Services environment, it is applicable as well to any inter-domain management scenario such as business process and service management or the management of networks, systems and applications in general. The WSLA framework consists of a flexible and extensible language based on XML Schema and a runtime architecture comprising several SLA monitoring services, which may be outsourced to third parties to ensure a maximum of accuracy. WSLA enables service customers and providers to unambiguously define a wide variety of SLAs, specify the SLA parameters and the way how they are measured, and relate them to managed resource instrumentations. Upon receipt of an SLA specification, the WSLA monitoring services are automatically configured to enforce the SLA. An implementation of the WSLA framework, the SLA Compliance Monitor, is publicly available as part of the IBM Web Services Toolkit.*

6. Shuping Ran. *A model for web services discovery with QoS.* SIGecom Exch., Vol. 4, Num. 1, pages 1–10, ACM, 2003.
   Abstract: *Web services technology has generated a lot interest, but its adoption rate has been slow. This paper discusses issues related to this slow take up and argues that quality of services is one of the contributing factors. The paper proposes a new Web services discovery model in which the functional and non-functional requirements (i.e. quality of services) are taken into account for the service discovery. The proposed model should give Web services consumers some confidence about the quality of service of the discovered Web services.*

Articles in Refereed Conferences:

1. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Mathias Weske. *Business Process Management: A Survey .* International Conference on Business Process Management (BPM), Lecture Notes in Computer Science, Vol. 2678, Springer Verlag, 2003.
   Abstract: *Business Process Management (BPM) includes methods, techniques, and tools to support the design, enactment, management, and analysis of operational business processes. It can be considered as an extension of classical Workflow Management (WFM) systems and approaches. Although the practical relevance of BPM is undisputed, a clear definition of BPM and related acronyms such as BAM, BPA, and STP are missing. Moreover, a clear scientific foundation is missing. In this paper, we try to demystify the acronyms in this domain, describe the state-of-the-art technology, and argue that BPM could benefit from formal methods/languages (cf. Petri nets, process algebras, etc.).*

2. Vikas Deora, Jianhua Shao, W. A. Gray, N. J. Fiddian. *A Quality of Service Management Framework Based on User Expectations.* ICSOC, pages 104–114, 2003.
   Abstract: *The ability to gauge the quality of a service is critical if we are to achieve the service oriented computing paradigm. Many techniques have been proposed and most of them attempt to calculate the quality of a service by collecting quality ratings from the users of the service, then combining them in one way or another. We argue that collecting quality ratings alone from the users is not sufficient for deriving a reliable or accurate quality measure for a service. This is because different users often have different expectations on the quality of a service and their ratings tend to be closely related to their expectations, i.e. how their*

expectations are met. In this paper, we propose a quality of service management framework based on user expectations. That is, we collect expectations as well as ratings from the users of a service, then calculate the quality of the service only at the time a request for the service is made and only using the ratings that have similar expectations. We give examples to show that our approach can result in a more accurate and meaningful measure for quality of service.

3. Howard Foster, Sebastián Uchitel, Jeff Magee, Jeff Kramer. *Model-based Verification of Web Service Compositions..* 18th IEEE International Conference on Automated Software Engineering (ASE 2003) , pages 152–163, IEEE Computer Society, 2003.

   Abstract: *In this paper we discuss a model-based approach to verifying web service compositions for web service implementations. The approach supports verification against specification models and assigns semantics to the behavior of implementation models so as to confirm expected results for both the designer and implementer. Specifications of the design are modeled in UML, in the form of Message Sequence Charts (MSCs), and mechanically compiled into the Finite State Process notation (FSP) to concisely describe and reason about the concurrent programs. Implementations are mechanically translated to FSP to allow a trace equivalence verification process to be performed. By providing early design verification, the implementation, testing and deployment of web service compositions can be eased through the understanding of the differences, limitations and undesirable traces allowed by the composition. The approach is supported by a suite of cooperating tools for specification, formal modeling and trace animation of the composition workflow.*

4. Rania Khalaf, Frank Leymann. *On Web Services Aggregation.* TES, pages 1–13, 2003.

   Abstract: *The Web services framework is enabling applications from different providers to be offered as services that can be used and composed in a loosely-coupled manner. Subsequently, the aggregation of services to form composite applications and maximize reuse is key. While choreography has received the most attention, services often need to be aggregated in a much less constrained manner. As a number of different mechanisms emerge to create these aggregations, their relation to each other and to prior work is useful when deciding how to create an aggregation, as well as in extending the models themselves and proposing new ones. In this paper, we discuss Web services aggregation by presenting a first-step classification based on the approaches taken by the different proposed aggregation techniques. Finally, a number of models are presented that are created from combinations of the above.*

5. Octavio Martín-Díaz, Antonio Ruiz Cortés, Amador Durán, David Benavides, Miguel Toro. *Automating the Procurement of Web Services.* ICSOC, pages 91–103, 2003.

   Abstract: *As government agencies and business become more dependent on web services, software solutions to automate their procurement gain importance. Current approaches for automating the procurement of web services suffer from an important drawback: neither uncertainty measures nor non-linear, and complex relations among parameters can be used by providers to specify quality-of-service in offers. In this paper, we look deeply into the roots of this drawback and present a proposal which overcomes it. The key point to achieve this improvement has been using the constraint programming as a formal basis, since it endows the model with a very powerful expressiveness. A XML-based implementation is presented along with some experimental results and comparisons with other approaches.*

6. Bart Orriëns, Jian Yang, Mike P. Papazoglou. *Model Driven Service Composition.* ICSOC, pages 75–90, 2003.

   Abstract: *The current standards for web service composition, e.g. BPEL. neither cater for dynamic service composition nor for dynamic business configuration. Our firm belief is that business processes can be built dynamically by composing web services in a model driven fashion where the design process is controlled and governed by a series of business rules. In this paper we examine the functional requirements of service composition and introduce a phased approach to the development of service compositions that spans abstract definition, scheduling, construction and execution. Subsequently, we analyze the information requirements for developing service compositions by identifying the basic elements in a web service composition and the business rules that are used to govern the development of service compositions.*

7. Vladimir Tosic, Bernard Pagurek, Kruti Patel. *WSOL - A Language for the Formal Specification of Classes of Service for Web Services.* ICWS, pages 375–381, 2003.

Abstract: *We are developing a language, Web Service Offerings Language (WSOL), for the formal specification of various constraints, management statements, and classes of service for Web Services. WSOL is an XML (Extensible Markup Language) notation compatible with WSDL (Web Services Description Language). A service offering in WSOL is a formal description of one class of service of a Web Service. It contains formal representation of various constraints: functional (pre-, post-, and future- conditions), Quality of Service (QoS, a.k.a. non-functional, extra- functional), and access rights. It also contains management statements, such as statements about prices, monetary penalties, and management responsibilities. One Web Service can be associated with multiple service offerings. For easier specification of similar service offerings, WSOL enables specification of constraint groups (CGs) and constraint group templates (CGTs). We have also developed a format for representation of dynamic relationships between service offerings. WSOL service offerings are simple contracts and SLAs (Service Level Agreements) between Web Services. Describing a Web Service in WSOL, in addition to WSDL, enables monitoring, metering, and management of Web Services. The Web Service, its consumer, or one or more designated third parties (usually SOAP message intermediaries) can meter QoS metrics and evaluate constraints in WSOL service offerings. Further, manipulation of service offerings can be used for dynamic adaptation and management of Web Service compositions. In addition, WSOL supports selection of a more appropriate Web Service and service offering for particular circumstances. The main distinctive characteristics of WSOL, compared to recent related works, are its expressive power, features that reduce run-time overhead, and orientation towards management applications.*

Books and Monographs:

1. Mariya Koshkina. *Verification of Business Processes for Web Services.* Ms. Thesis, York University - Department of Computer Science, October 2003.

   Abstract: *The Business Process Execution Language for Web Services (BPEL4WS or simply BPEL) is a recently developed language, which is used to specify interactions between web services. Among its features it allows specification of concurrent behavior. Erroneous specification can lead to such problems as deadlock. In our research we focus on the concurrency mechanism in BPEL. Our main goal is to analyze processes in order to detect possible deadlocks. To achieve this we introduce a process algebra called the BPE-calculus. It is a small language which captures all the BPEL features relevant to the analysis. This process algebra is modelled using a labeled transition system. An existing verification tool called the Concurrency Workbench is customized to use our BPE-calculus. This tool allows us to verify many properties of BPE-calculus processes specified in a logic called the mu-calculus, including deadlock freedom.*

Publications in Refereed Workshops:

1. M. Tian, A. Gramm, T. Naumowicz, H. Ritter, J. Schiller. *A Concept for QoS Integration in Web Services.* Web Information Systems Engineering Workshops, pages 149–155, December 2003.

   Abstract: *With the growing popularity of Web services, a general QoS support for Web services plays an important role for the success of this emerging technology. Unfortunately, current Web service environments do not offer comprehensive QoS support. In this paper, we present an approach that does not only enable the QoS integration in Web services, but also the selection of appropriate services based on QoS requirements regarding server and network performance. Furthermore, we present how application requirements regarding communication QoS are mapped onto the underlying QoS aware network at runtime, as well as how users can obtain real-time information about server performance in order to monitor the accomplishment of assured services, giving the user an instant QoS feedback.*

Technical Reports and Manuals:

1. Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, Ivana Trickovic, Sanjiva

Weerawarana. *Business Process Execution Language for Web Services, version 1.1.* BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems, 2003.

Abstract: *This document defines a notation for specifying business process behavior based on Web Services. This notation is called Business Process Execution Language for Web Services (abbreviated to BPEL4WS in the rest of this document). Processes in BPEL4WS export and import functionality by using Web Service interfaces exclusively. Business processes can be described in two ways. Executable business processes model actual behavior of a participant in a business interaction. Business protocols, in contrast, use process descriptions that specify the mutually visible message exchange behavior of each of the parties involved in the protocol, without revealing their internal behavior. The process descriptions for business protocols are called abstract processes. BPEL4WS is meant to be used to model the behavior of both executable and abstract processes. BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions. BPEL4WS defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.*

2. Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, Richard Franck . *Web Service Level Agreement (WSLA) Language Specification.* IBM Corporation, 2003.

   Abstract: *This document describes the specification language for service level agreements for Web Services, the Web Service Level Agreement (WSLA) language. WSLAs are agreements between a service provider and a customer and as such define the obligations of the parties involved. Primarily, this is the obligation of a service provider to perform a service according to agreed-upon guarantees for IT-level service parameters (such as availability, response time and throughput) for Web Services.*

   An SLA also specifies the measures to be taken in case of deviation and failure to meet the asserted service guarantees, for example, a notification of the service customer. The assertions of the service provider are based on a detailed definition of service parameters including the algorithms – how basic metrics should be measured in systems and how they are aggregated into composite metrics and SLA parameters. In addition, a WSLA can express the operations of monitoring and managing the service. This may include third parties (such as Management Service Providers) that contribute to the measurement of metrics, supervision of guarantees or even the management of deviations of service guarantees. These multi-party constellations necessitate the definition of the interactions among the parties supervising the WSLA.

   However, a WSLA only covers the agreed common view of a service between the parties involved. To actually act as a participant in a WSLA, parties have various degrees of freedom to define an implementation policy for a service and its supervision. Typically, the obligations of a WSLA must be translated into system-level configuration information, which can be proprietary to each party involved.

7 Publications in 2002

Articles in Refereed Journals:

1. Menasce, D.A.. *QoS Issues in Web Services.* Internet Computing, IEEE, Vol. 6, Num. 6, pages 72–75, November/December 2002.

   Abstract: *Quality of service (QoS) is a combination of several qualities or properties of a service, such as: availability is the percentage of time that a service is operating; security properties include the existence and type of authentication mechanisms the service offers, confidentiality and data integrity of messages exchanged, nonrepudiation of requests or messages, and resilience to denial-of-service attacks; response time is the time a service takes to respond to various types of requests; Response time is a function of load intensity, which can be measured in terms of arrival rates (such as requests per second) or number of concurrent requests. QoS takes into account not only the average response time, but also the percentile of the response time; and throughput is the rate at which a service can process requests. QoS measures can include the maximum throughput or a function that describes how throughput varies with load intensity. The QoS measure is observed by Web services users. These users are not human beings but programs that send requests for services to Web service providers. QoS issues in Web services have to be evaluated from the perspective of the providers of Web services and from the perspective of the users of these services.*

Articles in Refereed Conferences:

1. Massimo Mecella, Francesco Parisi Presicce an Barbara Pernici. *Modeling E -service Orchestration through Petri Nets .* Technologies for E-Services, Lecture Notes in Computer Science, Vol. 2444, pages 109–134, Springer Verlag, July 2002.

   Abstract: *B2B interaction requires new forms of coordination between participating organizations. Indeed, the main requirement is that the autonomy of each participating partner is preserved during the interaction, guaranteeing at the same time that the overall goals of the common process are reached. Mechanisms for regulating distributed workflow evolution when the workflow is composed of the invocation of E-services of different organizations are needed. The E-service orchestration model proposed in this paper provides a mechanism for supporting control of process evolution in terms both of control and data flows, and for distributing and assigning process responsibilities.*

2. Srini Narayanan, Sheila A. McIlraith. *Simulation, verification and automated composition of web services.* WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 77–88, ACM, 2002.

   Abstract: *Web services – Web-accessible programs and devices - are a key application area for the Semantic Web. With the proliferation of Web services and the evolution towards the Semantic Web comes the opportunity to automate various Web services tasks. Our objective is to enable markup and automated reasoning technology to describe, simulate, compose, test, and verify compositions of Web services. We take as our starting point the DAML-S DAML+OIL ontology for describing the capabilities of Web services. We define the semantics for a relevant subset of DAML-S in terms of a first-order logical language. With the semantics in hand, we encode our service descriptions in a Petri Net formalism and provide decision procedures for Web service simulation, verification and composition. We also provide an analysis of the complexity of these tasks under different restrictions to the DAML-S composite services we can describe. Finally, we present an implementation of our analysis techniques. This implementation takes as input a DAML-S description of a Web service, automatically generates a Petri Net and performs the desired analysis. Such a tool has broad applicability both as a back end to existing manual Web service composition tools, and as a stand-alone tool for Web service developers.*

3. Vladimir Tosic, Kruti Patel, Bernard Pagurek. *WSOL - Web Service Offerings Language.* CAiSE '02/ WES '02: Revised Papers from the International Workshop on Web Services, E-Business, and the Semantic Web, pages 57–67, Springer-Verlag, 2002.

   Abstract: *WSOL (Web Service Offerings Language) is an XML (Extensible Markup Language) notation*

compatible with the WSDL (Web Services Description Language) standard. While WSDL is used for describing operations provided by Web Services, WSOL enables formal specification of multiple classes of service for one Web Service. A service offering is a formal representation of one class of service for a Web Service. As classes of service for Web Services are determined by combinations of various constraints, WSOL enables formal specification of functional constraints, some QoS (a.k.a., non-functional) constraints, simple access rights (for differentiation of service), price, and relationships with other service offerings of the same Web Service. Describing a Web Service in WSOL, in addition to WSDL, enables selection of a more appropriate Web Service and service offering for particular circumstances. Further, it supports dynamic adaptation and management of Web Service compositions using manipulation of service offerings.

Books and Monographs:

1. J. Cardoso. *Quality of Service and Semantic Composition of Workflows.* Ph.D. Thesis, University of Georgia, 2002.

   Abstract: *Workflow management systems (WfMSs) have been used to support a variety of business processes. As organizations adopt new working models, such as e-commerce, new challenges arise for workflow systems. These challenges include support for the adequate management of quality of service (QoS) and the development of new solutions to facilitate the composition of workflow applications involving Web services. The good management of QoS directly impacts the success of organizations participating in e- commerce activities by better fulfilling customer expectations and achieving customer satisfaction. To enable adequate QoS management, research is required to develop mechanisms that specify, compute, monitor, and control the QoS of the products or services to be delivered. The composition of workflows to model e-service applications differs from the design of traditional workflows due to the number of Web services available during the composition process and to their heterogeneity. Two main problems need to be solved: how to efficiently discover Web services and how to facilitate their interoperability. To enhance WfMSs with QoS management, we have developed a QoS model that allows for the description of nonfunctional aspects of workflow components, from a quality of service perspective. To automatically compute the overall QoS of a workflow, we have developed a mathematical model and implemented an algorithm (SWR algorithm). Our QoS model and mathematical model have been validated with the deployment and execution of a set of production workflows in the area of genetics. The analysis of the collected data proves that our models provide a suitable framework for estimating, predicting, and analyzing the QoS of production workflows. To support, facilitate, and assist the composition of workflows involving Web services, we present a solution based on ontologies. We have developed an algorithm that workflow systems and discovery mechanisms can use to find Web services with desired interfaces and operational metrics, and to assist designers in resolving heterogeneity issues among Web services. Our approach provides an important solution to enhance Web service discovery and interoperability.*

Publications in Refereed Workshops:

1. S. Nakajima. *Model-Checking Verification for Reliable Web Service.* OOPSLA Workshop on Object-Oriented Web Services, 2002.

   Abstract: *Model-checking is a promising technique for the verification and validation of software systems. Web service, an emerging technology in the Internet, is an autonomous server that may offer an individual service. It sometimes requires to combine more than one to meet our requirements. WSFL(Web Services Flow Language) is proposed as a language to provide means to describe Web service aggregation. We are interested in how much the software model-checking technique can be used as a basis for raising reliability of Web service, Web service flow descriptions in particular. Our experience shows that faulty flow descriptions can be identified with the proposed method. The method is also very helpful in studying an alternative semantics of the WSFL in regard to the handling of dataflows.*

Technical Reports and Manuals:

1. Akhil Sahai, Anna Durante, Vijay Machiraju. *Towards Automated SLA Management for Web Services.* Num. HPL-2001-310, HP Laboratories, July 2002.

   Abstract: *In order to automate SLA management it is essential to specify SLAs in precise and unambiguous manner as well as keep the specification flexible. While precision will help automate the process of monitoring and metric collection, flexibility will enable extending it to unforeseen service level agreement specifications.*

2. Anbazhagan Mani, Arun Nagarajan. *Understanding quality of service for Web services.* IBM, 2002.

   Abstract: *With the widespread proliferation of Web services, quality of service (QoS) will become a significant factor in distinguishing the success of service providers. QoS determines the service usability and utility, both of which influence the popularity of the service. In this article, we look at the various Web service QoS requirements, bottlenecks affecting performance of Web services, approaches of providing service quality, transactional services, and a simple method of measuring response time of your Web services using the service proxy.*

8 Publications in 2001

Articles in Refereed Journals:

1. Nicholas R. Jennings, Peyman Faratin, A. R. Lomuscio, Simon Parsons, Michael Wooldridge, Carles Sierra. *Automated Negotiation: Prospects Methods and Challenges.* Group Decision and Negotiation, To Appear, 2001.

2. Frank E. Ritter, Richard M. Young. *Embodied models as simulated users: introduction to this special issue on using cognitive models to improve interface design.* Int. J. Hum.-Comput. Stud., Vol. 55, Num. 1, pages 1–14, 2001.

   Abstract: *Cognitive models provide a means for applying what is known from psychology to the design of interfaces, thereby improving their quality and usability. Existing uses of models include predicting time and errors for users to perform tasks, acting as embedded assistants to help users perform their tasks, and serving as surrogate users. Treating the design of human-computer interfaces as a form of engineering design requires the development and application of user models. A recent trend is for models to be built within the fixed framework of a cognitive architecture, which has been extended by the addition of simulated eyes and hands, enabling the construction of embodied models. Being embodied allows models to interact directly with interfaces. The resulting models can be used to evaluate the interfaces they use, and serve as explanations of users' behavior. The papers in this Special Issue point to a new route for the future, one in which models built within embodied cognitive architectures provide information for the design of better interfaces.*

9 Publications in 2000

Articles in Refereed Conferences:

1. Wil M. P. van der Aalst, Alistair P. Barros, Arthur H. M. ter Hofstede, Bartek Kiepuszewski. *Advanced Workflow Patterns.* CoopIS, pages 18–29, 2000.

   Abstract: *Conventional workflow functionality like task sequencing, split parallelism, join synchronization and iteration have proven effective for business process automation and have widespread support in current workflow products. However, newer requirements for workflows are enconntered in practice, opening grave uncertainties about the extensions for current languages. Different concepts, although outwardly appearing to be more or less the same, are based on different paradigms, have fundamentally different semantics and different levels of applicability -more specialized for modeling or more generalized for workflow engine posit. By way of developmental insight of new requirements, we define workflow patterns which are described imperatively but independently of current workflow languages. These patterns provide the basis for an in-depth comparison of 12 workflow management systems. As such, the work reported in this paper can be seen as the academic response to evaluations made by prestigious consulting companies. Typically, these evaluations hardly consider the workflow modeling language and routing capabilities and focus more on the purely technical and commercial aspects.*

2. Wil M. P. van der Aalst. *Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques.* Business Process Management, pages 161–183, 2000.

   Abstract: *Workflow management systems facilitate the everyday operation of business processes by taking care of the logistic control of work. In contrast to traditional information systems, they attempt to support frequent changes of the workflows at hand. Therefore, the need for analysis methods to verify the correctness of workflows is becoming more prominent. In this chapter we present a method based on Petri nets. This analysis method exploits the structure of the Petri net to find potential errors in the design of the workflow. Moreover, the analysis method allows for the compositional verification of workflows.*

10 Publications in 1998

Articles in Refereed Journals:

1. George A. Papadopoulos, Farhad Arbab. *Coordination Models and Languages.* Advances in Computers, Vol. 46, pages 330–401, 1998.

   Abstract: *A new class of models, formalisms and mechanisms has recently evolved for describing concurrent and distributed computations based on the concept of "coordination". The purpose of a coordination model and associated language is to provide a means of integrating a number of possibly heterogeneous components together, by interfacing with each component in such a way that the collective set forms a single application that can execute on and take advantage of parallel and distributed systems. In this chapter we initially define and present in sufficient detail the fundamental concepts of what constitutes a coordination model or language. We then go on to classify these models and languages as either "data-driven" or "control-driven" (also called "process-" or "task-oriented"). Next, the main existing coordination models and languages are described in sufficient detail to let the reader appreciate their features and put them into perspective with respect to each other. The chapter ends with a discussion comparing the various models and some conclusions.*

Technical Reports and Manuals:

1. Svend Frolund, Jari Koistinen. *QML: A Language for Quality of Service Specification.* Num. HPL-98-10, 63 pages, HP Laboratories, 1998.

   Abstract: *To be competitive, future software system must provide not only the correct functionality, but also an adequate level of quality of service (QoS). By QoS, we refer to non-functional properties, such as reliability, performance, timing, and security. To provide an adequate level of QoS, software systems need to include capabilities such as QoS negotiation, monitoring, and adaptation. These capabilities all require the expected and the provided QoS levels to be explicitly specified. QoS can be specified statically at the time of implementation, design, or dynamically at deployment or runtime. To facilitate QoS specification, we present a general Quality of service Modeling Language (QML) for defining multi-category QoS specifications for components in distributed object systems. QML is designed to support QoS in general, encompassing QoS categories such as reliability, performance, security, and timing. QoS specification in QML facilitate the static decomposition of a software system into components with precisely specified QoS boundaries. They also facilitate dynamic QoS functions, such as negotiations, monitoring, and adaptation. QML is designed for a good fit with object-oriented distributed architectures and concepts such as interfaces and inheritance. It also allows specification at a fine-grained level for operations, operation arguments, and attributes. QML enables user- defined QoS categories, and allows specifications within those categories to be associated with component interface definitions. In addition, checks can be made dynamically to determine whether one QML specification satisfies another. This mechanism allows us to dynamically match QoS requirements and offers*

11 Publications in **1997**

Publications in Refereed Workshops:

1. Sabata, B., Chatterjee, S., Davis, M., Sydir, J.J., Lawrence, T.F.. *Taxonomy for QoS specifications.* pages 100–107, 1997.

   Abstract: *It is becoming increasingly commonplace for multiple applications with different quality of service (QoS) requirements to share the resources of a distributed system. Within this environment, the resource management algorithms must take into account the QoS desired by applications and the ability of the system resources to provide it. We present a taxonomy for specifying QoS for the different components of a distributed system, from the applications down to the resources. We specify QoS as a combination of metrics and policies. QoS metrics are used to specify performance parameters, security requirements and the relative importance of the work in the system. We define three types of QoS performance parameters: timeliness, precision, and accuracy. QoS policies capture application-specific policies that govern how an application is treated by the resource manager. Examples of such policies are management policies and the levels of service. We explore each of these components of the QoS taxonomy in detail*

12 Publications in **1993**

Articles in Refereed Conferences:

1. Laurent Fribourg, Marcos Veloso Peixoto. *Concurrent Constraint Automata.* International Logic Programming Symposium, 1993.

   Abstract: *We address the problem of the specification and the proof of properties of concurrent systems which manipulate an unbounded number of data. We propose an approach based on an extended notion of automata, called "Concurrent Constraint Automata (CCA)". A CCA is an automaton with constraints and synchronous communication. By automata with constraints, we mean a state machine whose actions and states contain parameters that take their value in the set of natural numbers. (...)*