Christoph Bussler
Armin Haller et al. (Eds.)

LNCS 3812

# Business Process Management Workshops

BPM 2005 International Workshops
BPI, BPD, ENEI, BPRM, WSCOBPM, BPS
Nancy, France, September 2005
Revised Selected Papers

Springer

Christoph Bussler Armin Haller et al. (Eds.)

# Business Process Management Workshops

BPM 2005 International Workshops
BPI, BPD, ENEI, BPRM, WSCOBPM, BPS
Nancy, France, September 5, 2005
Revised Selected Papers



Springer

Volume Editors

Christoph Bussler
Armin Haller
National University of Ireland, Galway
Digital Enterprise Research Institute (DERI)
IDA Business Park, Lower Dangan, Galway, Ireland
E-mail: {christoph.bussler,armin.haller}@deri.org

Library of Congress Control Number: 2006920907

CR Subject Classification (1998): H.3.5, H.4.1, H.5.3, K.4.3, K.4.4, K.6, J.1

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-32595-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-32595-6 Springer Berlin Heidelberg New York



Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11678564 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3812

*Commenced Publication in 1973*
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

# Preface

Six parallel Business Process Management workshops were held on September 5th, 2005, in conjunction with the Third International Conference on Business Process Management (BPM 2005) in Nancy, France. This was the first time BPM had associated workshops, and the workshop program was a great success.

The topics of the workshops ranged from fundamental process modeling primitives to the recently emerged field of Web service choreography and orchestration, represented in the "Workshop on Business Processes and Services" and the "Workshop on Web Service Choreography and Orchestration;" a topic which intersects the research fields of business process management and Web services. Another strong focus was on business process design and business process intelligence, emerging areas that have gained increasing importance in supporting business process reengineering to derive superior process designs. These topics were covered in the respective workshops.

A widely discussed topic in several sessions was business process interoperability in the workshop on "Enterprise and Networked Enterprises Interoperability." Finally, the last workshop in these proceedings called "Business Process Reference Models" aimed at discussing different views on reference models in order to come to a common understanding of the terms involved.

We would like to thank the workshop organizers for their efforts in the workshop preparation, the organization of the review process, the exciting workshop programs and their management onsite and for their cooperation in the post-publication process.

Further, we thank all the authors for their submissions to the workshops, the Program Committees for their hard work during a brief reviewing period, the invited speakers and presenters for their very interesting presentations and the audience for their interest, questions and discussions.

Christoph Bussler
Armin Haller

# Organization

## Workshop Organization Committee

**Christoph Bussler**
Workshop Chair
Digital Enterprise Research Institute (DERI), Ireland

**Armin Haller**
Publications Chair
Digital Enterprise Research Institute (DERI), Ireland

## BPS 2005

Marlon Dumas, Queensland University of Technology, Australia
Schahram Dustdar, Vienna University of Technology, Austria
Frank Leymann, University of Stuttgart, Germany

## WSCOBPM 2005

Christoph Bussler, Digital Enterprise Research Institute (DERI), National University of Ireland

Alistair Duke, Next Generation Web Research, British Telecommunications plc (BT)

Dumitru Roman, Digital Enterprise Research Institute (DERI), Leopold-Franzens Universität Innsbruck

Michael Stollberg, Digital Enterprise Research Institute (DERI), Leopold-Franzens Universität Innsbruck

## BPI 2005

Malu Castellanos, Hewlett-Packard Labs, USA
Ton Weijters, U. of Eindhoven, The Netherlands

## ENEI 2005

Boudjlida, Nacer, UHP Nancy 1, LORIA, France
Panetto, Hervé, UHP Nancy 1, CRAN, France

## BPD 2005

Tom Davenport, School of Executive Education, Babson College

Hajo A. Reijers, Department of Technology Management, Eindhoven University of Technology

Michael Rosemann, Faculty of Information Technology, Queensland University of Technology

## BPRM 2005

Ekkart Kindler, Paderborn, Germany
Markus Nüttgens, Hamburg, Germany

# Table of Contents

## Workshop on Business Processes and Services



## Workshop on Web Service Choreography and Orchestration for Business Process Management

## Workshop on Business Process Intelligence



## Workshop on Enterprise and Networked Enterprises Interoperability



### Session: Enterprise Modelling Language for Enterprise Interoperability

**Session: Interoperability Applications and Case Studies**



## Workshop on Business Process Design: Past, Present, Future



## Workshop on Business Process Reference Models

# Preface
## (BPS 2005)

Service-oriented computing (SOC) is emerging as a promising paradigm for integrating software applications within and across organizational boundaries. In this paradigm, independently developed and operated applications are exposed as (Web) services which are then interconnected using a stack of Web-based standards including SOAP, WSDL, UDDI, WS-Security, etc. While the technology for developing basic services and interconnecting them on a point-to-point basis has attained a certain level of maturity and adoption, there are still many open challenges when it comes to managing interactions with complex services or managing interactions involving large numbers of services.

There exist strong links between business process management (BPM) and SOC. On the one hand, BPM may rely on SOC as a paradigm for managing resources (especially software ones), describing process steps, or capturing the interactions between a process and its environment. On the other hand, a service may serve as an entry point to an underlying business process, thereby inducing an inherent relation between the service model and the process model. Also, services may engage in interactions with other services in the context of collaborative business processes.

The First International Workshop on Business Processes and Services (BPS 2005) was organized with the aim of bringing together researchers and practitioners in the areas of BPM and SOC in order to further the fundamental understanding of the relations between business processes and services. The workshop's call for papers attracted nine submissions, of which the Program Committee selected four as full papers. In addition, two speakers presented their latest research and perspectives at the workshop: Wil van der Aalst on the topic of *interaction patterns* (organized jointly with the Workshop on Web Services Choreography and Orchestration for BPM), and Daniela Grigori on the topic of *service protocol adaptation*. Finally, a panel discussion on *intelligent processes and services for the adaptive enterprise* was held in conjunction with the Workshop on Business Process Intelligence. The panel was moderated by Malu Castellanos (HP Labs) and was composed of Wil van der Aalst (Eindhoven University of Technology), Boualem Benatallah (University of New South Wales), Frank Leymann (Stuttgart University), and Manfred Reichert (University of Twente).

The workshop was held during the preamble to the Third International Conference on Business Process Management (BPM 2005). We thank the officers and organizers of BPM 2005 for their support, as well as the members of the BPS 2005 Program Committee for their help in coming up with an exciting program.

September 2005

Marlon Dumas
Schahram Dustdar
Frank Leymann

# Organization

## Workshop Organizing Committee

Marlon Dumas, Queensland University of Technology, Australia
Schahram Dustdar, Vienna University of Technology, Austria
Frank Leymann, University of Stuttgart, Germany

## Program Committee

Karim Baina, ENSIAS, Morocco
Alistair Barros, SAP Research, Australia
Malu Castellanos, HP Labs, USA
Sara Comai, Politecnico di Milano, Italy
David Edmond, Queensland University of Technology, Australia
Marie-Christine Fauvet, University of Grenoble, France
Jose Fiadeiro, University of Leicester, UK
Dimitrios Georgakopoulos, Telcordia, USA
Avigdor Gal, Technion, Israel
Volker Gruhn, University of Leipzig, Germany
Manfred Hauswirth, EPFL, Switzerland
Rania Khalaf, IBM Research, USA
Heiko Ludwig, IBM Research, USA
Jeff Nickerson, Stevens Institute of Technology, USA
Mike Papazoglou, Tilburg University, The Netherlands
Cesare Pautasso, ETH Zürich, Switzerland
Manfred Reichert, University of Twente, The Netherlands
Hajo Reijers, Eindhoven University of Technology, The Netherlands
Shazia Sadiq, University of Queensland, Australia
Jianwen Su, UCSB, USA
Samir Tata, INT Evry, France
Mathias Weske, University of Potsdam, Germany
Andreas Wombacher, University of Twente, The Netherlands
Jian Yang, Macquarie University, Australia

# Guided Interaction: A Language and Method for Incremental Revelation of Software Interfaces for Ad Hoc Interaction

Phillipa Oaks and Arthur H.M. ter Hofstede

School of Information Systems - Faculty of Information Technology,
Queensland University of Technology,
GPO Box 2434, Brisbane, QLD 4001, Australia
{p.oaks, a.terhofstede}@qut.edu.au

**Abstract.** At present, most of the interest in web services is focussed on pre-planned B2B interaction. Clients interact with services using advance knowledge of the the data and sequence requirements of the service and pre-programmed calls to their interfaces. This type of interaction cannot be used for ad hoc interaction between services and their clients such as mobile devices moving in and around rich dynamic environments because they may not have the necessary knowledge in advance.

For unplanned ad hoc interaction an interaction mechanism is required that does not require clients to have advance knowledge of programmatic service interfaces and interaction sequences. The mechanism must ensure clients with different resources and diverse competencies can successfully interact with newly discovered services by providing assistance such as disambiguation of terminology, alternative types of inputs, and context sensitive error reporting when necessary.

This paper introduces a service interaction mechanism called *guided interaction*. Guided interaction is designed to enable clients without prior knowledge of programmatic interfaces to be assisted to a successful outcome. The mechanism is grounded in core computing primitives and based on a dialogue model. Guided interaction has two parts, the first part is a language for the exchange of information between services and their clients. The second part is a language for services to create interaction plans that allow them to gather the data they require from clients in a flexible way with the provision of assistance when necessary. An interpreter uses the plan to generate and interpret messages in the exchange language and to manage the path of the dialogue.

## 1 Introduction

Ad hoc is defined in Wordnet[1] as "unplanned" or "without apparent forethought or prompting or planning". In an ad hoc interaction environment, a software client could find, using for example a discovery mechanism, a software service

[1] wordnet.princeton.edu/

C. Bussler et al. (Eds.): BPM 2005 Workshops, LNCS 3812, pp. 3–17, 2006.

that provides the capability [1] it (the client) requires. Depending on the mechanism used to find services, the client may have little or no knowledge about the inputs the service requires, the dependencies between the data inputs, or the order of invocation of its operations, or the type and formatting information associated with these inputs.

Automated ad hoc interaction between web-based applications is a desirable goal. Applications that can automatically locate and interact with software services without a priori knowledge of their interfaces will be able to achieve many tasks that are beyond human resources at present. The Internet and the world wide web have now made many different types of information accessible on demand. The numbers of providers and the types of information available mean that the current interaction mechanisms based on prior or discovered knowledge about software interfaces and pre-programmed one-on-one interactions with those interfaces will not scale up to provide the potential benefits of ubiquitous web accessibility. The increasingly large number of software services[2] means that their use needs to be automated to the largest possible extent in order to fully profit from the promise and potential of ubiquitous service access.

Ad hoc interaction is very different from the present situation where software applications interact in a planned manner via interfaces. A software interface provides a static view of the operation signatures provided by a software service. Operation signatures detail the names of the operations and their input and output data types.

A partial solution to enable ad hoc interaction is the idea of "standard interfaces"[3]. In this solution all providers of a particular function use the same interface[4]. The solution relies on a common agreement between heterogeneous service providers on the best interface for a particular operation in a particular domain.

The problem is that that standard interfaces limit ad hoc interaction in two ways. Firstly, it is the responsibility of the client application's programmer to know or find out in advance how to call the operations supplied by the interface. Secondly, it locks providers into performing the task in a single way. If the same task can be performed with the same inputs but formatted in a different manner or with a different set of input data the provider must supply (non-standard) interface operations for each of these variants.

Another possible approach to ad hoc interaction is the use of techniques to obtain information about the operation signatures of software services at runtime such as dynamic CORBA and Java reflection. These techniques are used by client programs to gather information at runtime such as the names of oper-

[2] When accessible over Web-based standards, such software services are usually known as "Web services".

[3] www.learnxmlws.com/book/chapters/chapter11.htm

[4] Many common software development environments help the developer in this task by reading the service interface and automatically generating the code necessary to interact with the service. However, these tools are only usable if the interface of the service is available when the client application is developed.

ations and the data types the operations expect as input and return as output. One of the difficulties of doing this on a large scale is that the information that is available at runtime via reflection is syntactic rather than semantic. Client programs would need to interpret this derived syntactic information and create semantically correct request objects or messages to send to the provider at runtime. This interpretation effort places a large computational burden on the client at runtime which is only possible because the developer has programmed the client software with the necessary discovery, interpretation and message generation logic.

In reality the number of service providers, the number of possible operations and the different contexts in which those operations will be performed means both of these approaches will not scale to solve the problems of ad hoc interaction in the heterogeneous web services environment. These approaches do not have the flexibility required for ad hoc interaction such as the ability to allow alternative or equivalent inputs, the ability to provide help in the form of disambiguation of terminology at runtime with context sensitive error reporting and some form of dialogue control for services and their clients.

The next section (2) introduces *Guided interaction* interaction as a scalable and flexible solution to the problems of ad hoc interaction between web services. Guided interaction allows the use of alternative inputs, the provision of help and context sensitive error reporting and dialogue control for both the services and their clients. Section 3 introduces a shared language for interaction. Section 4 introduces the means of generating and interpreting messages using the language and gives an illustrative example of how guided interaction can be used to direct ad hoc interactions between heterogeneous services. Section 5 reviews some related work and the paper concludes with section 6.

## 2 Guided Interaction

Guided interaction is an abstraction mechanism that hides the details of web service interfaces by providing level of indirection between clients and the web service itself. A *guide* is a type of mediator or facade which presents a "user friendly" interface to a back end service. Clients may be other services, software agents or people. A guide, representing a service provider can tell clients what the capabilities of the service are, and proactively seek the input data the service needs. Guided interaction is based on a shared language for the exchange of information and dialogue control.

The guide is responsible for asking client applications what they want the service to do and for requesting the input data the service needs. This is a reversal of the current paradigm which makes the client application responsible for knowing or finding out what the software service can do, either by interpreting static interfaces or using advanced techniques like reflection.

Guides could be implemented as an alternative means of accessing one or more of the capabilities delivered by a single provider. It is not necessary that these capabilities are implemented as web services described by Web Services