Gesellschaft für Informatik (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into the fields of
- Seminar
- Proceedings
- Dissertations
- Thematics

current topics are dealt with from the fields of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure the high level of the contributions.

The volumes are published in German or English.

Information: http://www.gi-ev.de/service/publikationen/lni/

ISSN 1617-5468
ISBN 978-3-88579-241-3

This volume contains papers from the BPSC 2009 Conference on Business Process and Services Computing, the ISM 2009 International Workshop on Intelligent Service Management and YRW-MBP 2009 Young Researchers Workshop on Modeling and Management of Business Processes held in Leipzig Germany March 23-24, 2009. The papers discuss research and industry experiences with relation to process-centric service-oriented paradigm as it applies to the development and integration of enterprise and e-business information systems.

Business Process, Services Computing and Intelligent Service Management 2009

GI-Edition

Lecture Notes in Informatics

Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)

Business Process, Services Computing and Intelligent Service Management

BPSC 2009
ISM 2009
YRW-MBP 2009

March 23 – 25, 2009, Leipzig, Germany

147

Proceedings

Witold Abramowicz, Leszek Maciaszek
Ryszard Kowalczyk, Andreas Speck (Eds.)

Business Process, Services Computing and Intelligent Service Management

March 23 – 25, 2009,
Leipzig, Germany

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-147

ISBN 978-3-88579-241-3
ISSN 1617-5468

Volume Editors
Witold Abramowicz
Poznan University of Economics, Department of Information Systems,
61-875 Poznan, Poland
Email: Witold@Abramowicz.pl
Leszek Maciaszek
Macquarie University, Department of Computing
Sydney, NSW 2109, Australia
Email: leszek@ics.mq.edu.au
Ryszard Kowalczyk
Swinburne University of Technology, Centre for Information Technology Research
Hawthorn, VIC 3122, Australia
Email: rkowalczyk@it.swin.edu.au
Andreas Speck
Christian-Albrechts-University Kiel, Department of Computer Science
24098 Kiel, Germany
Email: Andreas.Speck@email.uni-kiel.de

printed by Köllen Druck+Verlag GmbH, Bonn

2nd International Conference on Business Process and Services Computing BPSC 2009

Introduction

The papers published in this volume were presented at the 2nd International Conference on Business Process and Services Computing (BPSC 2009) held in Leipzig Germany on 23 – 25 March 2009. The book includes papers selected for presentation in the rigorous review process conducted by the BPSC Program Committee.

The paper selection process considered the mission of BPSC conferences to become a prime international forum to discuss and publish research findings and IT industry experiences with relation to process-centric service-oriented paradigm as it applies to the development and integration of enterprise and e-business information systems. By looking at the business process as a first-class citizen in the IT world and by using the potential of services computing for creation of adaptive process-centric business solutions, BPSC conferences identify most hopeful trends and propose new directions for consideration by researchers and practitioners involved in large-scale software development and integration.

The Business Process Management (BPM) is based on the premise that applications (of business processes) can be evolving independently from process management, very much like they have been evolving independently from data management. The technology of web services and Service Oriented Architecture (SOA) are at the forefront of enabling a desired degree of process independence. The related technology stack includes document management and workflow solutions as well as enterprise integration and e-business interoperability solutions. The related research trends include integration of SOA with Event-Driven Architecture (EDA) and AI-inspired ideas of autonomic computing, multi-agent systems or SWS (Semantic Web Services).

Services within SOA are units of processing logic that collaborate to deliver enterprise logic as a combined effect of business process logic and application control logic. In other words, services apply to both kinds of logic and create a connectivity layer that enables independence of processes and applications. Services can ensure that processes and applications evolve gracefully together (very much like the aspect code and the base code in aspect-oriented programming) and the "crosscutting concerns" are well-documented and tractable. The SOA paradigm redefines the concept of an application as a distributed set of implementation-independent services executing as an orchestrated sequence of messaging and event processing. The confluence of SOA and BPM is resulting in a new process-centric paradigm that holds great promise for enterprise and B2B computing.

Moreover, one of the major BPM tasks is the constant need to adapt implemented processes to the changing business needs. As the degree of automation in BPM is currently rather limited, a great potential lies in the attempts to automate BPM a little further by the use of Semantic Web services and technologies. Founded on ontologies, Semantic Web provides methods and tools for the machine-understandable representation of collective knowledge and business processes in which such knowledge resides. Semantic Web Services (SWS) make use of Semantic Web technology to support the automated discovery, substitution, composition, and execution of SOA-based applications. Semantic Web goes as far as expecting that intelligent software agents can use semantic descriptions of Web services and resources to automate their use to accomplish user goals. Current research shows that combining the worlds of BPM and SWS may be very fruitful.

Acknowledgements

Our gratitude goes first of all to the Organizing Committee of BPSC 2009 – Karol Wieloch of Poznan University of Economics, Poland and Martin Matzner of University of Leipzig, Germany. Between two of them, they have taken care of coordinating the work of PC members, communicating with the authors, enabling electronic paper submissions and consequent reviewing tasks, registering of participants, as well as formatting this volume for publication.

The most responsible work was of course placed on the PC members who had to decide which papers were worthy of presentation at the conferences. We would like to thank them for marvelous work done.

Leszek Maciaszek
BPSC 2009 General Chair

Witold Abramowicz
BPSC 2009 Program Chair

Program Committee

The reviewing process was carried out by the BPSC 2009 Program Committee members. The final decision of acceptance/rejection was strictly based on the reviews. The PC members who contributed reviews were:

Rainer Alt	University of Leipzig, Germany
Giuseppe Berio	Université de Bretagne Sud, France
Miriam Capretz	University of Western Ontario, Canada
Schahram Dustdar	Vienna University of Technology, Austria
Bogdan Franczyk	University of Leipzig, Germany
Cesar Gonzalez-Perez	IEGPS, Spanish National Research Council, Spain
Paweł Kalczyński	J. California State University, Fullerton, USA
Marek Kowalkiewicz	SAP Australia Pty Ltd, Australia
Lech Madeyski	Wroclaw University of Technology, Poland
Mariusz Momotko	General Electric Money Bank, Poland
Mitsunori Ogihara	University of Rochester, USA
Eric Paquet	National Research Council, Canada
Václav Repa	Prague University of Economics, Czech Republic
Jürgen Sauer	University of Oldenburg, Germany
Janice C. Sipior	Villanova University, USA
Yoichi Takayama	Macquarie University, Australia
Ioan Toma	University of Insbruck, Austria
Krzysztof Wecel	Poznań University of Economics, Poland
Mathias Weske	University of Potsdam, Germany

Organizing Committee

Martin Matzner	University of Leipzig, Germany
Karol Wieloch	Poznań University of Economics, Poland

Table of Contents

BUSINESS PROCESS AND SERVICES COMPUTING

- Andrea Delgado, Francisco Ruiz, Ignacio García-Rodríguez de Guzmán, Mario Piattini
  Towards a service-oriented and model-driven framework with business processes as first-class citizens 19
- Denis Gagné, André Trudel
  A Formal Temporal Semantics for Microsoft Project based on Allen's Interval Algebra 32
- Stefan Jablonski, Bernhard Volz, Sebastian Dornstauder
  Evolution of Business Process Models and Languages 46
- Marwane El Kharbili, Elke Pulvermüller
  A Semantic Framework for Compliance Management in Business Process Management 60
- Ganna Monakova, Oliver Kopp, Frank Leymann, Simon Moser, Klaus Schäfers
  Verifying Business Rules Using an SMT Solver for BPEL Processes 81
- Emilian Pascalau, Adrian Giurca, Gerd Wagner
  Validating Auction Business Processes using Agent-based Simulations 95
- Artem Polyvyanyy, Sergey Smirnov, Mathias Weske
  On Application of Structural Decomposition for Process Model Abstraction 110
- Amir Afrasiabi Rad, Morad Benyoucef, Craig E. Kuziemsky
  On Modeling Web Service based Processes for Healthcare 123
- Yoichi Takayama, Ernie Ghiglione, Scott Wilson, James Dalziel
  Human Activities in distributed BPM 139

INTELLIGENT SERVICE MANAGEMENT

- Roman Belter, Rolf Kluge, Thomas Hering, Holger Müller
  A conceptual information model for service management dimensions 155
- Houssam Haitof, Hans-Dieter Wehle, Michael Gerndt
  FinGrid Accounting and Billing 167
- Christina Klüver, Jürgen Klüver, Rainer Unland
  A Medical Diagnosis System based on MAS Technology and Neural Networks 179

Towards a Service-Oriented and Model-Driven framework with business processes as first-class citizens

Andrea Delgado[1], Francisco Ruiz[2], Ignacio García-Rodríguez de Guzmán[2]
Mario Piattini[2]

[1]Computer Science Institute, Engineering Faculty, University of the Republica
Julio Herrera y Reissig 565, 5to. Piso
CP 11300, Montevideo,Uruguay
[2]Alarcos Group, Technology and IS Department, University of Castilla-La Mancha
Paseo de la Universidad No.4
CP 13071, Ciudad Real, España
adelgado@fing.edu.uy
{francisco.ruizg, ignacio.grodriguez, mario.piattini}@uclm.es

Abstract: One challenge that organizations face nowadays is to agilely react to changes in their business, adapting their business processes and technologies to new possibilities. To do so, organizations must be capable of separating the definition of their business processes from their technical implementation, which most are currently streaky. Applying the Service Oriented Computing (SOC) and Business Process Management (BPM) paradigms in conjunction, is an important but not trivial, step to take, involving different visions of business and technological challenges. The Model Driven Development (MDD) paradigm is also applied to serve as a bridge between business process models and technical models of the software to implement them. In this paper, the further work done on a service oriented methodology defined years ago is presented, considering business processes as the centre of software development. From business process models, software services are derived in a straightforward way, which will be automated by model transformations using the OMG service profile.

1 Introduction

Service Oriented Computing (SOC) involves the integration of technologies and concepts from various disciplines of the area of computation [PTDL07], and the Service Oriented Architecture (SOA) is a key element of its realization. In [KBS05] SOA is an architecture style of reusable software-based services, with well-defined public interfaces, where suppliers and consumers of services interact in a decoupled way for conducting business processes. A service provides business logic and data, a service contract, restrictions for the consumer, an interfaces that exposes functionality. A repository for storing service contracts and a service bus for connecting those involved are also defined. This vision is related to Business Process Management (BPM) which deals with efforts to optimize or adapt the organizational needs of business processes [BPMI] and BPM Systems (BPMS) as support tools that allow, among others, modeling and implementation of these processes in sequences of invocations to services (orchestration, choreography) [KBS05] [SF03].

The joint application of the SOC and BPM paradigms is the ideal way to model business processes implementing them independently of technology. An important feature is that it allows the logic of business processes to be clearly separated from the core business logic located in lower-services of finer granularity. Using a BPMS for the definition and control of these processes avoids codifying information and business rules directly in the software, thus facilitating the modification, re-configuration and optimization of the processes through graphical tools to define process flows, i.e., using the Business Process Modeling Notation (BPMN) [BPMN]. Processes could be added or modified with few adjustments, the business logic will be implemented only once in a service, increasing the reuse of knowledge and reducing inconsistencies and redundancies.

While progress has been made in the conceptual and technological aspects of SOC and SOA, BPM and business processes, there is still a lack of methodologies and guides for their joint application. The methodology presented here was initially proposed in 2005 [DGP06][De06][De07] in 2005, as an extension to a base process [BP00][BP06] which is an adaptation of RUP[RUP]. In this paper a modified methodology is presented for its application along with any software development process, with special focus on the business process models to derive software services. At this stage, this derivation is only methodological and conceptual, but it would be automated applying the Model Driven Development (MDD) paradigm, by defining models, metamodels and transformations between them, in a framework which integrates the three paradigms.

The rest of the paper is structured as follows: section 2 will discuss the related work, section 3 will present the business process-driven methodology, including the first proposal and the further work done, and section 4 will illustrate the use of the methodology through a case study based on the generic "Grant Loan" business process of a bank. Finally, in section 5 some conclusions and future work will be presented.

2 Related work

There are many guidelines, techniques and recommendations, and some methodologies that prescribe key aspects of the development process for service oriented software or business process implementation, but very few combine them. [EA04][Er05][KBS05] provides insight of the main aspects for SOA development, but in a general way, enumerating the features and elements of service orientation, and discussing specific technologies to implement it, i.e., Web Services. The SOMA plug-in [SOMA] of RUP in Rational Method Composer (RMC) [RMC], is contemporaneous with the first proposal of the methodology presented here, and similarly, it is focused on disciplines of Business Modeling, Design and Implementation. It defines more elements (activities, products, roles) making it more complex. A comparison is presented in [DGRP08].

Some of the most relevant works in the area include the following: [PJ06] defines phases, activities and artifacts for the development of services associated with business processes. It differs from ours in that although it defines guides for a service oriented development, it is also focused on the implementation of services as Web Services, defining technical aspects that cannot be applied with other technologies. Other works also include the MDD paradigm and its realization the Model Driven Architecture (MDA) [MDA03], as in [DML06] in which the main focus is the development of service oriented web systems defining models, metamodels and transformations between them to obtain a service composition model which expresses the interaction of services to perform business processes. Although they plan to, the method does not start with the modeling of business processes, as the methodology presented here does.

In [ZHD07] patterns are defined to guide the definition, transformation and implementation of technical processes using software services from business processes in which they call process-driven service oriented architecture. The macroflow pattern represents long-running business processes and the microflow pattern short-running technical processes, starting top-down from business process and bottom-up from software systems, joining them in the middle. Differently, our proposal does not classify types of processes, and does not use patterns to link models, as they are successively refined starting from business process models. In [RBM06] models and metamodels for services are defined to relate them to business processes and the underlying architecture, focusing the derivation of services to three architectures: brokerless, centralized and decentralized broker, providing a technical focus, which is not treated in our proposal.

Other approaches aims to relate business processes and software services as in [QDV05] which uses models expressed in ISDL to relate conceptual models and to asses conformance between them, which in our work is not made formally. In [LKT04] UML [UML] software artifacts like use cases, activity and collaboration diagrams are obtained automatically from business processes expressed in BPMN [BPMN], which is complementary to ours since we derive from BPMN business processes the needed services. In [HZ05] business processes are related to technical processes which use existing services, defining types of realization to identify the quality of the transformation between them. Although our proposal also takes into account existing services, it doesn't relate or constraint the business process because of services.

3 Business process-driven framework

Most of the current software development processes used are based on the philosophy of interaction and change, given the fact that the requirements of systems are usually unstable and it should be possible to incorporate them as they arise. These models classified as "heavy" or "agile", typically indicate that frequent releases are used to obtain feedback from users, and the construction of the system in iterative way is based on these incremental releases. A methodology for development-oriented services does not need to be a completely new methodology, but it could be built over the process or approach used in the organization, adding specific activities and artifacts for the development of services, and therefore any process could serve as a base.

The first proposal of the methodology [DGP06][De06][De07] made in 2005, was defined over a base process [BP00][BP06] adapted from RUP [RUP], and later RUP aspects like use cases or models were generalized to suit the process model of COMPETISOFT [CO06]. The core set defined emphasizes the Disciplines of Business Modeling, Design and Implementation to identify and model business processes, identify and derive the services making up these processes, and to design and build them.[EA04][Er05][KBS05] were references for general aspects and the Business Modeling Discipline was based on [RUP]. The core methodology was validated with cases studies in an academic context [DGP06][De07] and improved adding activities and deliverables in other Disciplines -which are not presented here- such as: testing, quality assurance, configuration management, deployment and management of services.

The core Disciplines, Activities, Deliverables and Roles presented here differ from the first proposal in many ways. First of all, they are independent of the RUP and its elements, and secondly, they are focused on modeling business processes and sub-processes, indicating how to derive the services from them. In our current work, the transformations between models are being defined, including the use of metamodels to make this derivation as automatic as possible.

3.1 Elements of the methodology

As we see in the service orientation paradigm, a key aspect is focused on business process modeling, taking them as first-class citizens from which to derive the required software services. The methodology proposed here defines two main activities in the Business Modeling Discipline to identify and model business processes. Besides, it defines five activities in the Design Discipline, which are key factors to identify, categorize, reuse, specify, and define the needed services, and their orchestration or choreography. The Implementation Discipline indicates the development of services as designed. Each activity, its input and output deliverables, and associated roles are clearly defined, as well as associated responsibilities. The roles defined are Software Architect, Analyst of business and requirements, and Developer specialized in technology.

Business Modeling Discipline

The purpose of the Business Modeling Discipline is to ensure that developers and others stakeholders have a common understanding of the Organization and derive the requirements for the software system, linking them to the identified business processes. The goal is to obtain a map of the organization and its processes to gain a better understanding of the business and requirements for the software system. The roles involved are Analyst and Architect, who have meetings with the client to identify them. The main deliverables are the Assessment of the target Organization with the identified key aspects and the Business Processes document with the specified business processes.

Assess the target Organization (MN1)

This activity aims to involve the project team with the organization for which the development is being carried out, in issues such as: the area of business, operation, employees, etc. of the organization, its current business processes, tools, skills of

people, customers, competition, technological challenges, problems and areas of improvement, clearly identifying those stakeholders involved in the business modeling effort. The role responsible is Analyst but Architect also participates, as both roles are involved in the business process model effort from which to derive the requirements for the application. The input of this activity is information about the business obtained from the stakeholders while the output is the assessment of the target organization which details its key aspects.

Identify Business Processes (MN2)

This activity has the objective of understanding as well as describing the business processes in the organization, mainly those related to the application being developed. Business processes are modeled with the selected notation, i.e. BPMN [BPMN], or Activity Diagrams of UML [UML], among others. The description must include: actors involved, control flow including sequence of activities, flow decisions and business rules. It is recommended to use process patterns (workflow patterns [VTKB03]) to help in various modeling aspects of business processes. The boundaries of business are clearly stated, indicating who and what interacts with the organization, specifying processes both in natural language and graphically. The role responsible is Analyst although Architect takes part too. As input, this activity has the Assessment of the Target Organization document, and minutes of meetings with customers. The output will be the document of the specified Business Processes.

Design Discipline

The purpose of the Design Discipline is that of identifying and cataloging services to perform the defined business processes , specify their interfaces and define their operations, the components that will implement them, reusing existing services in the organization. In addition, it has the target of defining the interaction of services in orchestrations or choreographies, to carry out the modeled business processes, including a BPMS when possible, for modeling, implementation, and monitoring of processes, among others. The roles involved are Architect, Analyst and Developer, being the Architect responsible for the whole discipline. The deliverables are the Services document, with specified services, and the Services Catalogue, to include the new services and search for existing ones.

Identify and categorize services (D1)

This activity is aimed at identifying the services needed to perform the modeled business processes, classifying them by type of service. Services represent features related to business concepts, and their categorization into a conceptual hierarchy helps in guiding the development and avoiding their proliferation in an indiscriminate manner, which is known as "services syndrome". Services are derived from business process models and their sub-processes and required functionalities, which have to be provided by services mapped to software subsystems. The role responsible is the Architect while Analyst and Developer participates as well. This activity has as input the document of Software Architecture, the Business Processes document, Requirements document and Services document. As output, it has the Services document including the identification and categorization of the informed services.

Specify services (D2)

This activity aims to specify the services, defining for each one its contract with the interfaces it provides and its operations and parameters. Each operation in the interface must specify: a) name of the method, b) required parameters and for each one, its name, type and description, c) return value, indicating name, type and description d) list of exceptions, e) brief description of the provided functionality f) pre-conditions for the successful execution, g) post-conditions that have to be valid after execution. The role responsible is Architect and Analyst is also involved, due to the fact that the specification can be carried out by both roles. It has as input and output the Services document, as input the identification and categorization of services, and as output the specification of services.

Investigate existing services (D3)

The purpose of this activity is to find services that are already implemented in the organization and can be reused in the application under development. It could be necessary to implement intermediary services for reusing other services that are not completely suited to the application, which is preferable than implement them again from scratch. To keep track of the services, their contracts and associated software systems, the Catalogue of services for its registration is defined. The role responsible is Architect and Analyst is also involved, as it is related to business modeling. The inputs of this activity are the Catalogue of services where existing services are listed and the Services document with the defined services, while the output is the Services document with existing services to reuse and the updated Catalogue of services.

Assign services to components (D4)

This activity aims to define the components to be implemented for providing the specified services. To assign services to components the question of whom (which component) will be provided with the services defined by the specified interfaces must be answered. The existing services also count if there are components providing the required functionality, or if an intermediary service could be developed to provide it using existing services. The role responsible is Architect and Developer also participates, as it involves mapping from design to implementation. The inputs are the Design and the Services documents, with information of software and services. As output it has the Development and Services documents, with the components defined for the services.

Define services interaction (D5)

The target of this activity is to define the sequence of interaction between services needed to perform the identified business processes. It will be an orchestration of services if it is internal to the organization or choreography if it is a collaborative process with communication with other organizations or entities. The invocation sequence of services is shown for each process or sub-process in a sequence diagram describing the interaction between the services involved. In addition, it is recommended to use a BPMS to define, implement, enact and manage the invocation sequences in languages such as BPEL[WSBPEL] or XPDL[XPDL]. The role responsible is the Architect but Analyst and Developer also participate because interaction of services needs business and technical knowledge. The inputs are Business Process, Requirements and Services documents, with information of needed services interaction to perform business

processes. As output, it has the Services document with the defined interaction and the Implementation document with the associated details.

Implementation Discipline

The Implementation Discipline aims to develop components providers of services, according to the assignment of services to components previously performed, the binding to them for invocations at the application level or between services, using the defined strategy.

Implement services (I1)

This activity has the objective of implementing the described services , taking into account the type of service, the designed interfaces, the interaction with other services (with or without a repository of services, binding in development or execution time). The only participating role is Developer who is in charge of the implementation of services. The inputs are the Services document, with the correspondence between design and implementation, Design and Implementation documents with associated details. Its output is the service implemented in the specified component.

4 Case study – the "Grant Loan" business process

The "Grant Loan" business process of a bank specifies the procedure defined to carry out the steps for the granting of loans to its customers. The bank is the target Organization in which the software project is taking part. The business process starts when a customer goes to the bank to request a loan, filling out the documentation. The request is sent to the Loan Authorization section, where the personnel study the documentation assessing the client's loan history, requested amount , and the client's credit information provided by the Credit Information Centre. The loan is approved or rejected and the resolution is sent back to the Customer Service area, which informs the client. If approved the client has to sign a loan contract and withdraw the money. In either case, the resolution is registered in the client's loan history. There are three actors involved: Client, Bank and Credit Information Centre, and inside the Bank there are also two different sections. It is worth mentioning that although the Bank owns the "Grant Loan" business process, it is a collaborative process since it has communications with other organizations.

4.1 Using the methodology

The first activities to be performed are the two defined in the Business Modeling Discipline, so the Analysts and the Architect have meetings with stakeholders and identify, among other things, the main business processes to develop. In this case study, it is the "Grant Loan" business process, which is shown in figure 1 using BPMN.

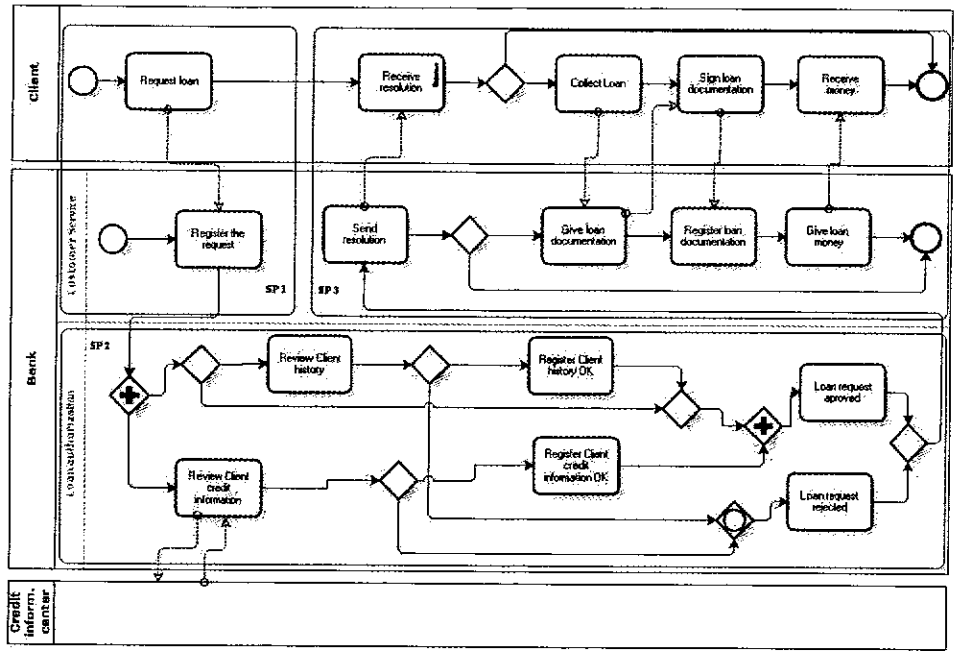

Figure 1: business process model of "Grant Loan" in BPMN

The notation and tool to be used in modeling the business processes have to be selected accordingly with the Organization's capabilities, including the economic ones; the important thing is to express the business processes graphically to help the understanding and further analysis for their implementation. Other important thing to keep in mind while carrying this selection out is the scope that the business process modeling effort would have. If it is mainly for better communication of ideas between the stakeholders, including business persons, it would be preferable to use a simple and business oriented notation like BPMN. Moreover, this notation could be transformed both into BPEL and XPDL, so business processes could also be deployed into a compatible business process engine, which supports the enactment of processes.

In the model presented in figure 1, it can be seen that there are three red rectangles which contain subparts of the process. These rectangles correspond to the following sub-processes: SP1-"Loan Request", SP2-"Loan Authorization" and SP3-"Loan Delivery", identified in the business process, which make the general process more manageable and simpler, facilitating the derivation of services from it. If uses cases are used, the identified sub-processes correspond to the system use cases performing the business process. To identify the needed services as defined in the Identify and categorize services (D1) activity, the functional requirements of each sub-process have to be identified and specified, and further refined, accordingly to the design activities. In the case study, the high level services identified from the "Grant Loan" business process are shown in table 1, together with the sub-process and a description.

Table 1: Definition of services from requirements of sub-process

Service	Sub-process	Description
RegisterLoanRequest	SP1–"Loan Request"	It models the interaction between client and Customer Service when the loan is requested.
ReviewClientHistory (defined for reuse)	SP2–"Loan Authorization"	It abstracts the management of the client's history.
ReviewClientCreditInfo (defined for reuse)	SP2–"Loan Authorization"	It abstracts the interaction with the Credit Information Center.
RegisterClientInfo (defined for reuse)	SP2–"Loan Authorization"	It registers Client's information.
RegisterLoanGranted	SP3-"Loan Delivery"	It models the interaction between client and Customer Service when the loan is approved.

As we can see in table 1, for the SP2-"Loan Authorization" sub-process there are three services identified and defined for reuse: ReviewClientHistory, ReviewClientCreditInfo and RegisterClientInfo. This is carried out since the probabilities that other applications need to obtain the client's history of loans in the bank as well as the client's credit information in the Credit Information Centre, or to register client's information, are high. Also, it is advisable to design services for reuse in the organization, since the Catalogue of services is searched every time to find services to be reused in new applications. Once a service is identified and categorized, its functional contract is specified, as indicated in the Specify services (D2) activity. The services are iterative and incrementally identified, defined, categorized and specified, performing the two first activities once and again as the projects evolve and the team gains knowledge of the organization and its business processes. Once the main business processes have been identified and specified, the Investigate existing services (D3) activity is performed, searching in the Catalogue of services for services or system functionalities that bring the desired behavior, or for a composition that could bring it. After that, the components to provide the functionalities of the services are designed, following the guides included in the Assign services to components (D4) activity.

Finally, the interaction of services into orchestrations or choreographies is defined, as the Define services interaction (D5) activity indicates, that is, taking a business process and its identified sub-process, the flow of interaction of services has to be defined, following the business process model flow. This interaction could be presented by sub-processes to make the whole process more understandable, joining them to make up the general process. This interaction between defined services is shown in a UML sequence diagram, as it clearly describes the interchange of messages between the participants. In figure 2 the sequence diagram in UML for the SP2-"Loan Authorization" sub-process is shown, including the interaction with internal defined services and external services from other organizations.

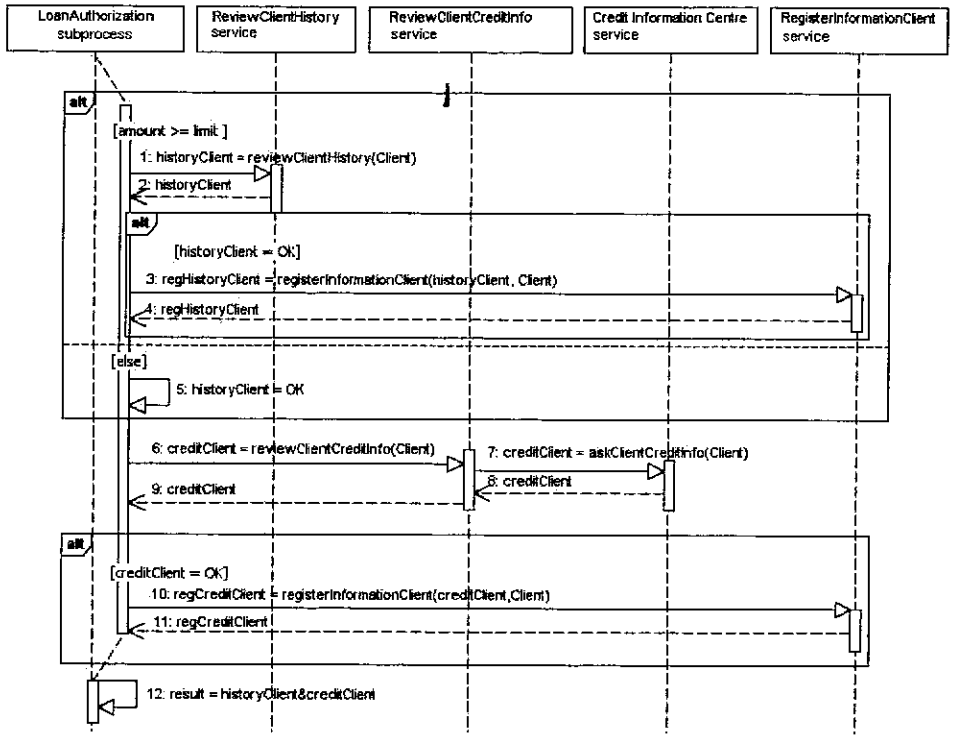

Figure 2: UML sequence diagram for SP2-"Loan Authorization" sub-process

As it can be seen in figure 2, the two invocations in the SP2-"Loan Authorization" sub-process to the services performing it, are concurrent (as it is stated in the business process model), and the result it returns is made up of the logical AND of the results from its invocations. The sequence diagram also helps us clarify operations in the contract of each service, or refine them if it shows that something is missing or misunderstood. So, it could be carried out when defining the service's contract, although the whole interaction of services is still unknown. The diagram shows how and with which other services each one interacts, and clearly identifies external services to be negotiated with partner organizations.

As service functionality has to be implemented, classes, subsystems and/or components associated with each service have to be defined too. In the example, the subsystems defined could be "ClientManagement", "LoanManagement", "AccountsManagement", assigning the services identified to them, i.e. the RegisterLoanRequest and RegisterLoanGranted services are assigned to the "LoanManagement" subsystem, and the "ReviewClientData" service -not shown here- is assigned to the "ClientManagement" one. Inside each subsystem the classes to implement the operations defined are designed, for now, in a conceptual way. We are working in defining transformations from BPMN to OMG service profile [UPMS07] to support the automatic derivation, when possible, of services classes from business processes.

5 Conclusions and future work

In this work, we have presented a service oriented methodology for the derivation of software services from business process models, which is included in a framework under development for the improvement of business processes. The methodology is composed of a core set of Disciplines, Activities, Deliverables and Roles, to guide the development of this type of software, taking into account the main characteristics defined by the paradigm.

Business processes are first-citizen classes since one of the most important activities prescribed is the business process modeling in the selected notation, applying process patterns to reuse known solutions to modeling problems. As part of the methodology, the derivation of services from business processes and sub-processes is stated following the guides of the defined activities. To illustrate its use, a case study is presented based on the generic "Grant Loan" of a bank, which explains step by step how to identify and obtain the needed services from business process models, and how to define the interaction of services to perform the identified business processes.

In this way, software services are derived from business process models in a straightforward way, so the next step to complete the methodological aspect of the framework we are working on, is to define transformations between models and metamodels from BPMN and the OMG service profile, to automate the defined derivations as much as possible. We will also investigate for other notations. We think that the methodology proposed here is simple but powerful enough to be easily applied in various environments complementing almost any software development process which needs to incorporate guides for service oriented development based on business processes.

References

[BPMI] Business Process Management Initiative,http://www.bpmi.org/

[BPMN] Business Process Modeling Notation (BPMN) Spec.v.1.1, Object Management Group (OMG),<http://www.bpmn.org/Documents/BPMN%201-1%20Specification.pdf>, 2008.

[CO06] COMPETISOFT – Process Improvement to increase competitively of Small and Medium Iberoamerican Software Industry, CYTED, http://alarcos.infcr.uclm.es/Competisoft/

[De06] Delgado, A.: Metodología para desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture), LatinAmerican Informatics Conference (CLEI'06), Santiago de Chile, Chile, 2006.

[De07] Delgado, A.: Metodología de desarrollo para aplicaciones con enfoque Service Oriented Architecture (SOA),Master Thesis, PEDECIBA, UdelaR, Montevideo, Uruguay, 2007.

[DGP06] Delgado, A.; González, L.; Piedrabuena, F.: Desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture), Iberoamerican conference on Software Engineering and Knowledge Engineering (JIISIC'06), Puebla, México, 2006.

[DGRP08] Delgado, A.; García-Rodríguez de Guzmán, I.; Ruiz, F.; Piattini, M.: Metodologías de desarrollo para Service Oriented Architectures con Rational Unified Process, Scientific-technical Conference on WS and SOA (JSWEB'08), Sevilla, España, 2008.

[DML06] De Castro, V.; Marcos, E., López Sanz, M.: A model driven method for service composition modelling: a case study, J.WebEngineering&Technology,Vol.2,No.4, 2006.

[DP00] Delgado, A.; Pérez, B.: Modelado de proceso de software, Proyecto Taller V, Instituto de computación, Facultad de Ingeniería, Universidad de la República, 2000.

[DP06] Delgado, A.; Pérez, B.: Modelo de Desarrollo de Software OO, Experimentación en un curso de Ingeniería de Software, Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento, JIISIC'06, Puebla, México, 2006.

[EA04] Endrei, M.; Ang, J.; Arsanjani, A. et. al: Patterns: Service Oriented Architecture and Web Services, IBM Redbook, SG24-6303-00, 2004.

[Er05] Erl,T.:Service OrientedArchitecture:Concepts,Technology,andDesign,PrenticeHall,2005.

[HZ05] Henkel, M.; Zdravkovic, J.: Supporting Development and Evolution of Service-based Processes, International Conference on e-Business Engineering (ICEBE'05)

[KBS05] Krafzig, D.; Banke, K.; Slama, D.: Enterprise SOA, Service Oriented Architecture Best Practices, Prentice Hall, 2005.

[LKT04] Liew, P.; Kontogiannis, K.; Tong, T.: A Framework for Business Model Driven Development,International Workshop on Software Technology and Engineering Practice (STEP'04), 2004.

[MDA03] Model Driven Architecture (MDA) Guide version 1.0.1, Object Management Group (OMG), <http://www.omg.org/mda/>, 2003.

[PJ06] Papazoglou, M.; Jan van den Heuvel, M.: Service Oriented design and development mehodology, J.WebEngineering&Technology,Vol.2,No.4, 2006.

[PJ07] Papazoglou, M.; Jan van den Heuvel, M.: Business Process development life cycle methodology, Communications of the ACM, Vol. 50, No. 10, October 2007.

[PTDL07] Papazoglou, M.; Traverso, P.; Dustdar, S.; Leymann, F.: Service Oriented Computing: State of the Art and Research Challenge, IEEE Computer Society, 2007.

[QDV05] Quartel, D.; Dijkman, R.; van Sinderen, M.: An approach to relate business and applications services using IDSL, International Enterprise Computing Conference (EDOC'05)

[RMC] IBM - RMC, <http://www- 306.ibm.com/software/awdtools/rmc/>

[RUP] IBM - RUP, <http://www- 306.ibm.com/software/awdtools/rup/>

[RBM06] Roser, S.; Bauer, B.; Müller, J.: Model- and Architecture-Driven Development in the Context of Cross-Enterprise Business Process Engineering, International Conference on Services Computing (SCC'06), 2006.

[SF03] Smith, H.; Fingar, P.: Business Process Management: The third wave, Meghan-Kieffer Press, 2003.

[SOMA] IBM – RMC - RUPforSOMA, Plug-inV2.4,<http://www.ibm.com/developerworks/rational/downloads/06/rmc_plugin7_1 /#12>

[SV06] Stahl, T.; Volter, M. et. al.: Model-Driven Software Development, Technology, Engineering, Management, John Wiley & Sons, Ltd., 2006.

[UML] Unified Modeling Language (UML) 2, Object Management Group (OMG), <http://www.omg.org/spec/UML/2.1.2/ >, 2007.

[CBK06] UML Profile and Metamodel for Services (UPMS),Object Management Group (OMG),<http://www.omg.org/docs/ad/07-06-02.odt>,<…./ad/08-05-03.pdf>, 2007/08

[VTKB03] van der Aalst, W.; ter Hofstede, A.; Kiepuszewski, B.; Barros, A. Workflow Patterns,
Distributed and Parallel Databases, 14(3), pages 5-51, July 2003

[WSBPEL] Web Services Business Process Execution Language Version 2.0, OASIS, <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

[XPDL] XML Process Definition Language (XPDL), Workflow Management Coalition (WfMC)
<http://wfmc.org/index.php?option=com_docman&task=cat_view&gid=42&Itemid=72>

[ZHD07] Zdun, U.; Hentrich, C.; Dustdar, S.: Modeling Process-Driven and Service Oriented Architectures Using Patterns and Pattern Primitives, ACM Transaction on the Web, Vol.1, No.3, Article 14, September 2007.