Domain-Oriented Design Patterns For Service Processes

Maik Herfurth
Hilti Befestigungstechnik AG
Global Information Technology
Buchs SG, Switzerland
email: maik.herfurth@hilti.com

Thomas Schuster
Pforzheim University of Applied Sciences
Pforzheim, Germany
email: thomas.schuster@hs-pforzheim.de

Abstract — **Business processes in the domain of service procurement have the potential to increase efficiency and reduce cost. These business processes are getting more complex since service providers and service consumers collaborate and interact in network structures. A precise description, modeling, analysis and execution of service procurement business processes for the implementation of process-oriented information systems is required to unlock these potentials and optimize network collaboration. In this article, we propose a pattern based approach for business process models to improve this collaboration systematically. The approach is based on patterns of service phases and service modules to increase efficiency and reduce cost.**

Keywords – service e-procurement; business process design pattern; service phase patterns; service module patterns

I. INTRODUCTION

The service sector is a fast-growing sector in all industrial nations and therefore, it has gained significant importance in all national economies [1]. Today, the strategic impact of services overruns products. With shifting the focus towards services and moving to a more service centric perception away from a product centric view, a new paradigm *service dominant logic* is postulated [2]. New business models are arising with cross-company network structures where service providers and service consumers act in service networks, so called *service chains* [3]. More and more companies outsource different areas and reduce the degree of company-internal value-add. More services are sourced externally and the meaning of service procurement is increasing exponentially. Service procurement becomes decisive for success and competition. A business process, which defines the control flows of service procurement, is called *service process*, a process object, which represents data flow is called *service object*. Within the network structures of service chains, the complexity of service processes is raising. New requirements of service-oriented procurement result from the definition of services because of the specific characteristics like immateriality and integrality, which determine the specific characteristics of transactions between service providers and service consumers. The use of modern information technologies like service-oriented architecture (SOA) for the electronic service processes of service procurement sounds quite promising. The efficiency and performance of service processes can be improved and cost can be reduced. Due to the increasing competition and cost pressure in the domain of service procurement, service processes leverage the improvement potential and come to the fore of companies. A systematic and structured approach for modeling and analyzing service processes based on patterns leads to a harmonization and integration of service processes and advantage transparency and structure. We develop two new patterns, namely, service phase patterns and service module patterns. Their application is presented based on examples and their advantages are outlined. The remainder of this article is structured as follows: in Section 2, we look at the current challenges of service procurement and motivate our approach. We look into design patterns and introduce Petri nets as a formal modeling language in Section 3. Service procurement design patterns are proposed in Section 4. In Section 5 and Section 6, we introduce two different design pattern types for service processes and motivate the advantages. Finally, Section 7 of this article concludes with findings and outlook on future work.

II. CHALLENGES AND MOTIVATION

Today's service procurement processes of small and medium-sized companies are often characterized by heterogeneous and product-oriented business processes [3]. In contrast to products, services require interaction that is more personal and are more difficult to describe and to measure. Therefore, the procurement of services turns out to be more complex (1) due to process descriptions, (2) due to data descriptions and (3) due to process iterations, (4) due to unknown result of a service after a service request and (5) due to the individuality of services. A high amount of manual process tasks and therefore, missing automation can be observed [5]. Cross-company process structures are heterogeneous and the process and data flow design are influencing each other: an information asymmetry results out of different proprietary data formats and inconsistent data. The electronic procurement of services has still not reached a high level of maturity [6]. In summary, the following challenges can be observed:

- complex collaborative internal and cross-company business process models lead to high opacity, iterations and adjustment cost
- heterogeneous business processes, long process flows and use of different media lead to non-seamless processes

 ISBN: 978-1-61208-534-0

- heterogeneous data structures, different data formats and descriptions lead to non-integration and non-harmonization of data
- heterogeneous information technology (IT) landscapes with different interfaces lead to missing integration
- low maturity level of service process automation leads to long throughput times, redundancy of tasks and source of errors

Existing business process modeling methods for modeling, analysis and implementation of service processes aren`t matured enough and only cover partially the domain specific needs for service e-procurement. New methods for the harmonization, integration and standardization are needed:

- best practice based definition for understanding business processes and data
- harmonization and integration of business processes
- harmonization and integration of data
- integration of information systems

These challenges can be addressed by new domain-oriented design patterns. In this paper, we present a new domain-oriented design pattern approach based on the formal modeling language Petri nets. The Petri net based design patterns build up best practice knowledge and incorporate an integrated modeling approach for process and data structures. Design patterns provide an immediate benefit (1) by reducing design and integration efforts, (2) by encouraging best practices, (3) by assisting in analysis, (4) by exposing inefficiencies, (5) by removing redundancies, (6) by consolidating interfaces and (7) by encouraging modularity and transparent substitution [7].

III. BUSINESS PROCESSS DESIGN PATTERN

A *pattern* is a discernible regularity and the elements of a pattern repeat in a predictable manner. Patterns are an abstraction of a concrete problem observation, which was recognized due to its frequent appearance in a certain domain [8]. Hence, patterns result from experiences and behavioral observation. They represent identical modes of thought, design fashions, behaviors or courses of action, which can be repeated and reproduced. Software design patterns are introduced in the domain of software engineering. They are general solutions to solve a problem in a given context. Thus, a design pattern provides a reusable blueprint that may speed up the development of software [9] and is considered as a solution template for high quality software [10]. Software design patterns always represent solutions to common design problems in a given context [11]. Design patterns at architectural level provide solution templates at component level (e.g., as the pipes and filters pattern does). Object-oriented design patterns, on the other hand, typically show relationships and interactions between classes or objects, they are distinguished into creational patterns, structural patterns, behavioral patterns and concurrency patterns.

Design patterns can create substantial improvements of software quality and reduce costs associated to development and maintenance. Nowadays, design patterns are widely used since they capture and promote best practices in software design. Many catalogues for design patterns are known today, like patterns for software engineering from Gamma et al. [12] and patterns for the enterprise integration scenarios of software applications from Hohpe and Woolf [13].

Similarly, business process design patterns describe best practices for process models in a certain domain. These patterns are also based on empirical knowledge about process activity execution. Thus, business process design patterns are formalizing common structures of activities of process and data flows [14]. While a concrete pattern is bound to a specific modelling language, its abstraction is language independent and can be transferred to other business process modeling languages as well [12]. Specific patterns for the Petri net modelling language are outlined in the next subsection.

Barros et al. [15] define service bilateral and multilateral interaction patterns, which allow emerging web services functionalities like choreography and orchestration. Additionally, domain-oriented design patterns offer a flexible mechanism with clear boundaries in terms of well-defined and highly encapsulated parts being aligned with constraints of the considered domain [16].

A. Petri net based process pattern

Petri nets are a formal modeling language to model, analyze, simulate and execute distributed, discrete systems. Petri nets are bipartite graphs. Also, Petri nets can be used to model different levels of detail. Petri nets offer the modeling of static and dynamic elements, limited capacities of places and anonymous tokens to capture process objects. Petri nets are graphically represented by *tokens* (process objects), *places* (conditions), *transitions* (process tasks) and *directed arcs* (arrows). Places are containers for tokens and describe pre- or post-conditions for transitions. Places represent local conditions and describe static process components. Transitions describe dynamic process components and represent local state transitions [17]. As a formal and platform-independent modeling approach, *high-level Petri nets* allow for modeling with a precise description of individualized tokens. Hence, this can be used for the formalization of domain-specific process objects [18]. As a well-established modeling language, Petri nets have also been proposed to model process and data structures as mechanism for analysis and software-based execution of business processes. Especially, high-level Petri nets may be used as input for transformation to executable business processes.

Van der Aalst and ter Hofstede [19] define fundamental *Workflow patterns* based on Petri nets to formalize requirements of workflow languages and information systems. These patterns are further distinguished *into exception handling patterns, control flow patterns, data flow patterns* and *resource patterns*. Further existing examples for Petri net process patterns are given by *TimeNET* [20] (a software tool to model, analyze and control manufacturing systems based on colored Petri nets), *EXSPECT* [21] (a repository of tool for standardized business processes in logistics and production) or *CIMOSA* [22] (the modeling and analysis of cross-company value chains). To formally model supply chains as business processes, Liu et al. [23] are using Petri nets to define basic patterns of supply chains. Schuster [8] proposes resource assignment patterns and defines high-level resource nets.

IV. SERVICE PROCUREMENT DESIGN PATTERN

In the domain of service procurement, service suppliers and service consumers collaborate with each other. The collaboration itself can be considered as an instance of a service procurement process model. An instance of a service process model is described as a choreography of specific service phases, which comprise internal and cross-company service processes and service modules. The order of exchanged messages is predefined. Choreography is used to define a cross-company service process out of several independently orchestrated service processes (see Fig. 1). The interaction between several partners for the procurement of services based on the exchanged data is described [24]. The order of the exchanged data is pre-defined. Only valid orders of data between partners are allowed to be defined.

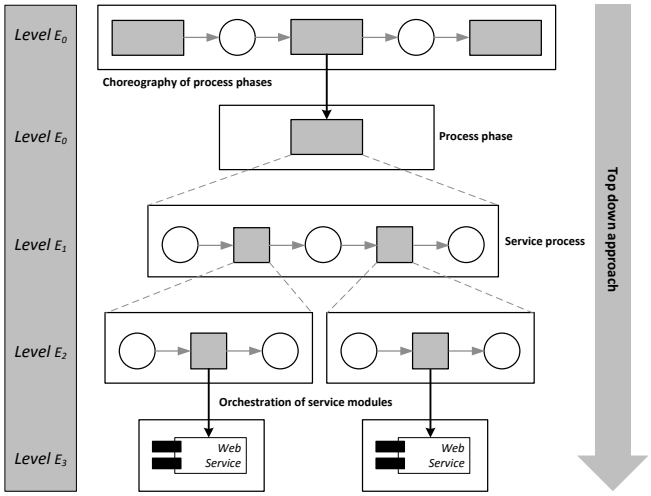


Figure 1. Choreography of service phases and orchestration of service modules on different abstraction levels

Based on our review of scientific literature [25] and empirical case studies [26], we derive new patterns for service e-procurement, which represent best practice of service procurement processes. We introduce the design patterns *Service Phase Patterns* (*SPP*) and *Service Module Patterns* (*SMP*) also supporting software architectures based on service-oriented architecture. These patterns define hierarchic structures and provide a structured concept for the modeling and implementation of service procurement process models. SPP and SMP ensure and precisely describe the order of message exchange and interaction in bilateral and multilateral service chains and constitute required process interfaces.

A sequence of SPP includes data flow, complex service processes and web services. SPP consist of SMP and are linked by internal and cross-company process interfaces. The choreography of SPP serves as a connector between orchestrations of SMP and their internal service processes. A concrete sequence of SMP is pre-defined. The combination of SPP and SMP results into global, cross-company service processes, which define the interaction of internal service processes accordantly.

The pattern-based application using SPP and SMP leads to a top-down approach from specific process phases down to detailed service process descriptions, executed by web services. The pattern-based approach enables a coordinated realization of service processes in information systems at the execution level. In a first modeling and description approach of service e-procurement process descriptions, we use Petri nets as modeling language to describe domain-specific service phases and service modules. Based on this definition, we further develop these patterns based on XML nets [27], a high-level Petri net variant.

V. DESIGN PATTERNS FOR CHOREOGRAPHY OF PROCESS FLOW AND DATA FLOW

Service procurement processes (between service providers and service consumers) are characterized by highly collaborative service processes. The collaboration is defined by specific process and data flows based on specific process interfaces. It can be observed that typical recurrent service procurement process models are characterized by a specific order of data flow and by specific service procurement types. These recurrent orders of process and data flow defining service procurement types can be described by patterns.

A. Service Phase Patterns (SPP)

SPP choreograph service procurement phases and therefore, the data flow in order to represent and manage cross-company interaction. The logic of process flow instances is determined as well. SPP are characterized by capsulated service procurement processes on a higher abstraction level. SPP configure different *service procurement types*. A service procurement type pre-defines a service process model, which represents a specific process flow occurrence for service procurement. The following service procurement types are defined:

- A planned need of a service is required and a frame contract does not exist.
- A non-planned need of service is required and a frame contract doesn't exist
- A planned need of a service is required and a frame contract exists.
- A non-planned need of service is required and a frame contract exists.

Based on a specific service procurement type, SPP can be configured to choreograph the data flow and process flow (see Fig. 2).

 ISBN: 978-1-61208-534-0

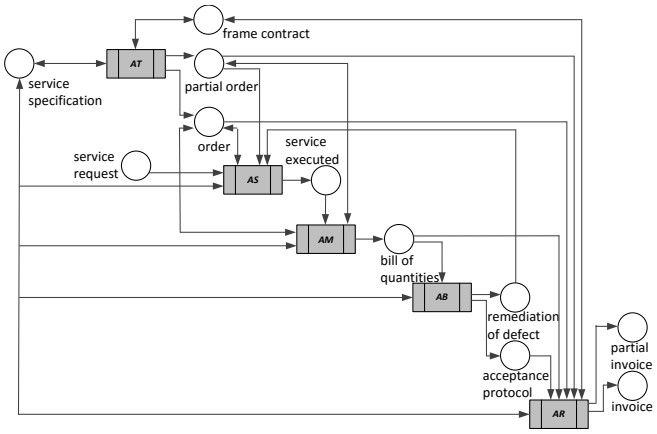


Figure 2. Service procurement type as choreography of SPP modeled as Petri net

B. Definition and modeling of SPP with Petri nets

SPP are transition-bounded service processes and are represented in a Petri net as a single transition, which can be extended to sub nets. *Service places* are defined by a set of service object-specific places, which are classified into *service object places SO, static and dynamic service interface places SI* and *service document places SD* (see Fig. 3).

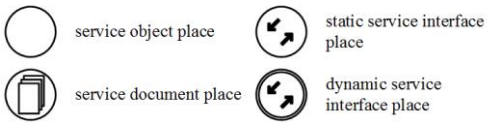


Figure 3. Service object specified places *SO*, *SD* and *SI*

SPP represent a self-contained set of cross-company collaborative service processes. SPP are connected by cross company interfaces defined by service object-specific interface places *SI* and *SD*. SPP are represented graphically by a rectangle, which includes the service phase name (see Fig. 4).

Figure 4. SPP modeled as Petri net

Domain-specific concepts for a formal modeling approach of service processes in the context of service e-procurement are also formalized based on high-level Petri nets. The transfer of the presented formalized concepts further enables communication and information context. SPP are further developed into formalized patterns based on high-level Petri nets with individualized and distinguishable tokens representing the service e-procurement-specific data transfer in information systems. Specific interfaces in collaborative and cross-company service processes represented by service process phases are identified, formalized and defined as patterns based on high-level Petri nets. The set of service object-specific places *SPS* are typified as object containers for service processes and defined as an XML net. *SPS* represent the complex data flow based on XML service objects to define the data and document exchange in collaborative cross-company service processes.

The set of typified service object specific places *SPS* is further distinguished into the set of service object places *SSO*, service interface places *SSI* and service document places *SSD*. The domain specific stereo types of SPP are proposed. SPP based on XML nets represent coarsened structures of capsulated service processes and SMP. SPP are defined based on specific, typified input and output places. They represent process patterns. SPP are defined based on the process and data flow of collaborative service e-procurement processes and consider the specific phases. Fig. 5 shows the example of the pattern of the service phase *Accounting AC*.

SPP are formally defined as the set *TSP* based on single transitions with dedicated service object-specific places. The sets of input places S_{SO}^{IN} and output places S_{SO}^{OUT} are assigned and consist of the sets of service object places *SSO*, service interface places *SSI* and service document places *SSD*. Each service phase $t_{sp_j}^{i}$ is defined by its internal structure, which enables the composition of service phases.

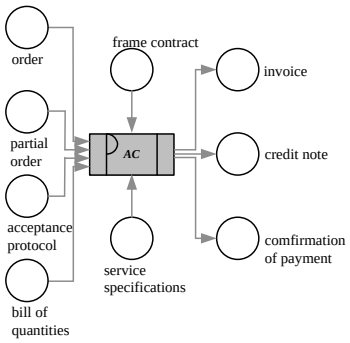


Figure 5. SPP example *Accounting AC* modeled as XML net

In case of a composition of two service phases $t_{sp_a}^{i}$ and $t_{sp_b}^{i}$, input and output places are melted together. The set of *TSP* is defined as single transitions of transition bounded sub XML nets *XN'=(S',T',F')* and service process modules $t_{SM}^{i} \in TSM$. The syntactical compatibility is a requirement for the composition of SPP.

C. Advantages of SPP

SPP for service processes of service procurement process models enable the following advantages:

- SPP choreograph service procurement phases and data flow and therefore, they define recurring process flow and data flow orders based on best practice in service procurement.
- SPP are defined patterns for the service procurement phases *specification*, *request*, *quotation*, *order*, *execution*, *measurement*, *acceptance* and *accounting*.
- SPP configure best practice service procurement types.
- SPP enable a pre-defined data flow. The order of exchanged data is prescribed for the definition of domain-specific standard for the interaction and data exchange for

partners [28]. The definition of specific data and process sequences provides the basis for required process interfaces in complex business-to-business (B2B) scenarios.

VI. DESIGN PATTERNS FOR ORCHESTRATION OF PROCESS FLOW AND DATA FLOW

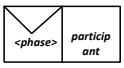
Collaborative service processes describe the interaction of service suppliers and service consumers on a detailed business process level. The pre-defined order of recurring service phases can be further structured into detailed service modules. These patterns of detailed service modules describe a recurring process and data flow characterized by specific process interfaces.

A. Service Module Patterns (SMP)

SMP are characterized by capsulated electronic service processes. SMP define orchestrations of their internal capsulated service processes. The process and data flow is orchestrated. The activities of these service processes are executed by web services for the horizontal and vertical integration of different information systems. One of the main characteristics of SMP is collaboration: the collaborative service process of a process participant is further capsulated into service modules.

B. Definition and modeling of SMP

Collaborative SMP are transition-bounded service processes and are represented in a Petri net as a single transition, which can be extended to sub nets. SMP define self-contained collaborative service processes of one collaboration participant (service supplier or service consumer). SMP are represented graphically by a rectangle, which includes the specific service phase name as well as the participant of the service process (see Fig. 6). The set of *SI* and *SD* are input and output places of service modules.



service module pattern

Figure 6. SMP modeled as Petri net

SMP are defined based on high-level Petri nets. SMP represent coarsened collaborative service processes of one process participant. The collaborative service process consists of several SMP representing all process participants and therefore, the entire service process of a SPP. SMP are connected via a set of input and output places *SPS* to model bidirectional interaction and communication patterns like *sending* and *receiving*. The internal structure of a service module is built by a coarsened service net and consists of a set of internal input and output places (S_{SM}^{IN}, S_{SM}^{OUT}) and internal transitions. The set of input and output places is defined as an internal *module interface* of a service module. The internal structure of a service module fulfills the requirements of a workflow net and soundness criteria [29]. A service module interface $S_{sm_1}^{IN/OUT}$ of one service module t_{sm}^{1} can be melted with the service module interface $S_{sm_2}^{IN/OUT}$ of another service module t_{sm}^{2}. The set of service modules *TSM* is defined as single transition of a transition-bounded sub XML net *XN'=(S',T',F')* as part of an XML net and dedicated to one service process phase of the set of service process phases *TSP*. The syntactical compatibility of service process modules enables the composition of service process modules. Syntactically compatible service process modules have completely overlapping process interfaces. The composition of service process modules causes the melting of the common set of interface places. Based on a specific SPP, a capsulated service process can be further detailed into SMP to orchestrate the data flow and process flow (see Fig. 7).

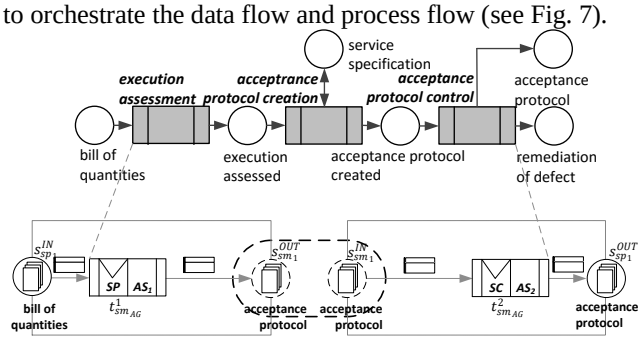


Figure 7. SMPs modeled as XML net

C. Advantages of SMP

SMP for service procurement processes enable the following advantages:

- SMP orchestrate the process flow and data flow and therefore, they define recurring collaborative service processes based on best practice in service procurement.
- SMP define patterns for the detailed service procurement processes *specification*, *request*, *quotation*, *order*, *execution*, *measurement*, *acceptance* and *accounting*.
- SMP enable a modularization concept for modeling and implementing collaborative service processes. The defined activities can be further modeled and implemented by web services.

VII. CONCLUSION & OUTLOOK

We presented two new patterns SPP and SMP for the choreography of service phases and the orchestration of service modules in collaborative cross-company business process models. The design patterns are intended to describe recurring service process sequences based on observed best practices. SPP and SMP support the modeling and implementation of electronic service processes. SPP and SMP are defined based on a formal Petri net modeling approach for service e-procurement business process models.

Both the formal modeling language of Petri nets, as well as the service procurement domain-specific patterns lead to improved domain understanding, and support simulation based analysis as well as process implementation. The definition of SPP and SMP enables

- an *integrated*, *formalized modeling approach* of service processes and service objects.
- the modeling of *hierarchic service processes and modularization* of collaborative service processes.
- the definition and modeling of *service process interfaces*.

 ISBN: 978-1-61208-534-0

- a step-wise transformation of modeling to different *formalization levels*.
- the support of *distributed business processes* based on service oriented architecture (SOA).
- the validation of service process models.

As next steps, we further need to verify that these patterns meet the set of design specification and its intended purpose. We will also have to conduct further experiments to ensure the usability, the level of details and completeness of the defined patterns. We will evaluate our pattern approach by analyzing further service e-procurement use cases in order to validate the correctness and use of purpose of the pattern approach. We are planning to integrate the new defined patterns into a modeling approach for service processes, so called *Service nets*. The presented patterns SPP and SMP will be a part of the definition of Service nets. This modeling approach will serve as modeling support for collaborative service processes and distributed hierarchic service process models. The integrated modeling of service processes and service objects, the formalization of different levels of abstraction (hierarchy) and the modularization of service processes (interface design) will be addressed herein. The domain-specific extension of Petri nets is only based on a syntactical level without changing the semantic characteristics of Petri nets. Our pattern approach is based and developed on use cases of service e-procurement of industrial services. E-procurement of other service domains will also be analyzed and validated. The pattern approach can be transferred to further service process types different than procurement. Analog service process types are repair orders, return orders, warranty service orders and further ones. The pattern-based modeling approach will also further be used for simulation experiences and benchmarking of collaborative service processes of different service supplier and service consumer combinations.

REFERENCES

[1] A. Linhart, J. Manderscheid and M. Roeglinger, "Roadmap to Flexible Service Processes - A Project Portfolio Selection and Scheduling Approach", ECIS 2015, Paper 125, pp. 1-16, 2015.

[2] R. F. Lusch and R. Nambisan, "SERVICE INNOVATION: A SERVICE-DOMINANT LOGIC PERSPECTIVE", MIS Quarterly volume 39 (1), pp. 155-175, 2015.

[3] O. Kleine and R. Schneider (editors), "Service Chain Management in the industry – an approach for planning of cooperative industrial services", VDM Dr. Müller, 2010.

[4] L. R. Smeltzer and J. A. Ogden, "Purchasing Professionals' Perceived Differences between Purchasing materials and Purchasing Services", Journal of Supply Chain Management, volume 38 (19), pp. 54-70, 2002.

[5] M. Herfurth and P. Weiß, "Conceptual Design of Service Procurement for collaborative Service Networks", Collaborative Networks for a Sustainable World: 11th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2010, France, pp. 435-442, 2010.

[6] P. P. Maglio and J. Spohrer, "Fundamentals of Service Science", Journal of the Academy of Marketing Science, volume 36 (1), pp. 18-20, 2008.

[7] R. J. Glushko and T. McGrath, "Designing Business Processes With Patterns", The MIT Press Cambridge, Massachusetts, 2005.

[8] T. Schuster, "Modeling, integration and analysis of resources in business processes", KIT Scientific Publishing, 2012.

[9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison–Wesley, 1995.

[10] J. Arlow and I. Neustadt, "Enterprise Patterns and MDA", Addison-Wesley, 2003.

[11] E. Gamma and K. Beck, "Eclipse to be extended. Principles, Patterns and Plug-Ins", Addison-Wesley, 2005.

[12] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 2001.

[13] G. Hohpe and B. Woolf, "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions", Addison-Wesley Longman Publishing Co. Inc., 2004.

[14] O. Barros, "Business process patterns and frameworks: Reusing knowledge in process innovation", Business Process Management Journal, volume 13 (1), pp. 47-69, 2007.

[15] A. Barros, M. Dumas, and A. ter Hofstede, "Service Interaction Patterns", Proceedings of the 3rd International Conference on Business Process Management, France, Springer, pp. 302-318, 2005.

[16] D. Port, "Derivation of domain specific desing patterns", USC-CSE Annual Reasearch and Technology Week Presentations and Binder Materials, 1998.

[17] W. Reisig, "Petri nets – an introduction", Springer, 1982.

[18] J. Desel and A. Oberweis, "Petri nets in applied informatics: introduction, basics and perspectives", Business Informatics, volume 38(4), pp. 359-367, Vieweg + Teubner, 1996.

[19] N. Russell, A.H.M. ter Hofstede, W.M.P. van der Aalst, and N. Mulyar, "Workflow Control-Flow Patterns: A Revised View", BPM Center Report BPM-06-22, BPMcenter.org, 2006.

[20] A. Zimmermann, J. Freiheit, and A. Huck, "A Petri net based design engine for manufacturing systems", Journal International Journal of Production Research, volume 39 (2), pp. 225-253, 2001.

[21] W.M.P. van der Aalst and A.W. Waltmans, "Modeling logistic systems with EXPECT", in H.G. Sol, K.M. van Hee, Dynamic Modeling of Information Systems, pp. 269-288, 1991.

[22] M. Dong and F. F. Chen, "Process modeling and analysis of manufacturing supply chain networks using object oriented Petri nets", Robotics and Computer Integrated Manufacturing, volume 17, pp. 121-129, 2001.

[23] E. Liu, A. Kumar, and W.M.P. van der Aalst, "Managing Supply Chain Events to Build Sense-and-Response Capability", in D. Straub and S. Klein, editors, Proceedings of International Conference on Information Systems (ICIS 2006), Milwaukee, Wisconsin, pp. 117-134, 2006.

[24] M. Weske, "Business Process Management: Concepts, Languages, Architectures", Springer, 2007.

[25] M. Herfurth, A. Meinhardt, J. Schumacher, and P. Weiß, "eProcurement for Industrial Maintenance Services", Proceedings of Leveraging Knowledge for Innovation in Collaborative Networks: 10th IFIP WG 5.5 Working Conference on Virtual Enterprises, Greece, pp. 363-370, 2009.

[26] P. Weiss, M. Herfurth, and J. Schumacher, "Leverage Productivity Potentials in Service-oriented Procurement Transactions: E-Standards in Service Procurement", RESER Conference, Germany, pp. 1-22, 2011.

[27] K. Lenz, "Modeling and execution of e-business processes with XML nets", Dissertation J.W. Goethe University Frankfurt am Main, 2003.

[28] A. Grosskopf, G. Decker, and M. Weske, "The Process. Business Process Modeling Using BPMN", Meghan-Kiffer Press, 2009.

[29] W.M.P. van der Aalst, "Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets, Systems Analysis - Modeling – Simulation", volume 34 (3), pp. 335-367, 1999.

 ISBN: 978-1-61208-534-0