Design Pattern-based Modeling of Collaborative Service Chains

Maik Herfurth
Hilti Befestigungstechnik AG
Buchs SG, Switzerland
e-mail: maik.herfurth@hilti.com

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

Abstract—**Increased market competition between service providers has promoted differentiation, currently often based on offerings of so-called hybrid services. Hybrid services are complex since service providers and service consumers collaborate and interact in network structures – also called service chains. Service chains also include collaborative, cross-company service processes amongst different participants. The effectiveness of these processes constitutes a major competitive factor since they possess potential to increase efficiency and reduce cost. Along with hybrid services, e-procurement becomes strategically important. While basic service processes are typically company internal, service chains include cross-company administration, especially for service e-procurement. If enough domain specific data is available, information systems can support service operations and e-procurement in an integrated mode. Hence, a precise description, modeling and analysis of service processes is required for the implementation of process-oriented information systems is required to unlock these potentials and optimize network collaboration. To improve the transition from planning to implementation of collaborative service chains with incorporated e-procurement, we suggest an integrated and formalized modeling approach. Our modeling approach is based on Petri nets and includes a pattern-based tactic to develop business process models. To improve this collaboration systematically we distinguish hierarchically between service phase patterns and service module patterns. Finally, our approach will be demonstrated by case studies taken from the domain of industrial service procurement.**

Keywords—service processes; service objects; service e-procurement; hybrid services; integrated modeling approach; modeling language; design patterns; service phase pattern; service module pattern

I. INTRODUCTION

This article is an extended version of the recently published paper "*Integrated Modeling Approach iServMod for Modeling, Analysis and Execution of Collaborative Service Processes in Service Chains*" [1]. The service sector is a fast-growing sector in all industrial nations and therefore, it has gained significant importance for all national economies [2]. Today, the strategic impact of services outruns products. With shifting the towards services and moving away from a product centric view, a new paradigm known as *service dominant logic* is postulated [3]. Current surveys highlight that services increasingly create value offerings to customers and thus constitute an integral element of many products [4].

This includes industrial services as one example of hybrid services. Industrial services contribute a significant share of companies total spending and ensure required operational levels and availability of systems and facilities. Therefore, industrial service e-procurement is gaining importance and an integrated perspective of goods and services is required [5],[6]. New business models are arising with cross-company network structures where service providers and service consumers act in service networks. These cross-company value chains with incorporated flows of goods, cash and information are called *service chains* [7]-[9]. Progressively companies outsource different areas and reduce the degree of company-internal value-add. An important example of service chains are industrial maintenance services. They are typically delivered through third-party service providers which guarantee availability and reliability of industrial facilities and infrastructures.

With more services sourced externally, the meaning of service procurement is increasing exponentially. For the procurement of industrial services, several service providers and service consumers must interact in multilateral service chains. Capital goods producers, goods and service consumers, as well as specialized service providers interact to produce, operate and administrate these hybrid services. Hence, the ability to integrate and share product and service offerings of external business partners turns out to be a major competitive factor. In consequence, service providers take focus on supporting consumer processes or even on offering to provide larger parts of value creation processes. Since service consumers request different service types from different service providers, new types of flexible collaborations are emerging.

In this case, service e-procurement constitutes an important segment of e-business activities. It compasses extensive use of *Information and Communication Technology* (*ICT*) to improve productivity and business processes. Electronic processes support business interactions reducing interfaces, process and throughput times and enhance coordination of activities, procedures and integration of resources.

Additionally, e-business standards can help to support a shared process understanding and increase process transparency amongst business partners by harmonization and structuring of exchanged business data. As follows, e-business standards facilitate enhanced interoperability.

Electronic business processes of service chains are in focus throughout planning (modeling level) and operation (execution level). A business process, which defines the control flows of service procurement, is called *service process*. A process object, which represents flow of business data is called *service object*. Within the network structures of service chains, the complexity of service processes is raising. New requirements on service-oriented procurement result from the service definition. Characteristics that add further requirements are *immateriality* and *integrality*. Both determine the specific characteristics of transactions between service providers and service consumers. The use of modern information architectures such as service-oriented architecture (SOA) for the electronic service processes and service e-procurement is promising improvements. Nevertheless, there is still a lack of a precise modeling, analysis and benchmarking approach for these service processes. The efficiency and performance of service processes still has to be improved and cost has to 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.

This article is focusing on the development of a systematic and structured approach for an integrated analysis and modeling of service processes in service chains. The approach is based on patterns leading to a harmonization and integration of service processes. In consequence, this leverages transparency and structure of service chains. We suggest a formalized modeling method for collaborative service processes in service chains for further improvement. We developed two new patterns, namely, *Service Phase Patterns* and *Service Module Patterns*. Their application is presented based on examples and their advantages are outlined. Our concentration within that is on service e-procurement. The overall research approach is based and evaluated on case studies of a research and standardization project [10].

The remainder of this article is structured as follows: in Section II, the current challenges of service procurement are outlined and the pattern-based modeling approach of *integrated Service Modeling* (*iServMod*) is motivated. An overview of the state of the art in service modeling is shown in Section III. In Section IV, the modeling approach of *iServMod* is presented. Design patterns are proposed and we introduce Petri nets as a formal modeling language. In Section V, the modeling of service objects is demonstrated and Section VI introduces two different design pattern types for service processes. In Section VII, we present the *Service Phase Pattern* (*SPP*) and continue with the pattern of *Service Module Pattern* (*SMP*) in Section VIII. The modeling of Service nets *eSN* is shown in Section IX, and in Section X, the modeling of high-level Service nets *hSN* is described in detail. Based on the modeling of *eSN* and *hSN*, Section XI introduces the *Evolutionary Procedure Model* (*EPM*) for the pattern-based integrated modeling approach. Finally, the use case driven application of *iServMod* in Section XII and final remarks, findings and an outlook on future work in Section XIII conclude this article.

II. Challenges and Motivation

Service chain collaborations achieve economies of scale, economies of scope and lower transaction costs. These collaborations are confronted with several challenges: missing harmonization, integration and standardization of cross-company service processes. Therefore, the creation of new collaborations often suffers from low quality of business interactions caused by integration and transaction costs, manual exception handling, offline communication (media breaks) and long lead times resulting in less transparency and low quality of processes and data. Today's service procurement processes of small and medium-sized companies are mostly defined by heterogeneous and product-oriented business processes [11]. Also, a high amount of manual process tasks and therefore, missing automation can be observed [12].

In contrast to products, services typically require personal interaction and are more difficult to describe and to measure. Therefore, the procurement of services turns out to be particularly complex due to (1) process descriptions, (2) data descriptions, (3) process iterations, (4) unknown result of a service after a service request, and (5) individuality of services. 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. Thus, the electronic procurement of services has still not reached a high level of maturity [13]. Inefficiencies result from internal and especially cross-company handling und coordination of transactions and non-harmonized and non-integrated electronic service processes. Service processes must support procedural rules and service logic of required interactions as well as communication between service providers and service consumers [5].

In turn industrial service e-procurement is still a source of high costs because underlying service processes are error prone. Errors and failures occur foremost through the absence of coherent e-business standards and reference frameworks. Together, both could offer meta-models for processes, standardized data objects and interaction patterns for the service logic. Summarized, we observe the following challenges:

- complex collaborative internal and cross-company service process models lead to high opacity, iterations and adjustment costs

- heterogeneous service processes, long running processes and use of different media lead to error-prone process execution
- 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 lead to long throughput times, redundancy of tasks and source of errors

As stated above, the aforementioned shortcomings result from missing standards in document exchange and lack of information harmonization. In addition, service processes for administrative order processing in service chains did not draw much attention in the past. However, especially these processes require many resources and incorporate long process and throughput times. Existing business process modeling methods for modeling, analysis and implementation of service processes aren`t mature enough and only cover partly the domain-specific needs for service e-procurement. Thus, new methods for the harmonization, integration and standardization have to be established and need to include:

- best practice based definition for improved understanding of service processes and data
- harmonization and integration of service processes and service data
- integration of information systems with support of these service processes and service data

These challenges can be addressed by a formal modeling approach based on domain-oriented design patterns. In this article, we present a new domain-oriented analysis and modeling approach based on the formal modeling language Petri nets. This Petri net based modeling language incorporates design patterns that build upon best practice knowledge as well as 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 [14].

A. *Research objectives*

Within this article, we follow paradigms defined in design science. Thus, knowledge can be gained by creation and evaluation of artifacts in the form of models, methods and systems. In contrast to empirical research, the goal is not necessarily to evaluate the validity of research results with respect to their truth, but rather the usefulness of the built artifacts as a tool to solve certain problems [15]. In this spirit, we will impose requirements driven by analysis of service and service e-procurement literature, interviews with domain experts as wells as hypotheses. The requirements analysis will disclose the decisions for the design concept of our planning approach. In contrast to an approach driven by theory, the basis for the design has not necessarily to be formulated as hypothesis. Hence, the planning method will be constructed, implemented and tested in a real environment.

In this article, we propose a model-based approach for the following reasons: information and knowledge must be captured before it can be part of sound analysis and utilization. Informal, semi-formal as well as formal models offer an abstract possibility to represent information and knowledge. Furthermore, graphical representations such as class diagrams, data-flow diagrams, state-transition diagrams or Petri nets ease understanding and exchange between stakeholders, both for the expert and the non-expert. Overall, this facilitates the communication between persons of different domains. In addition, formal languages allow description of certain phenomenon uniquely and precisely, but with a high level of abstraction. In addition, they can be evaluated and verified or be used to automate certain tasks. The goal of this article is the definition of a modeling method, which improves the quality of service chains by a domain-specific modeling approach, linked collaborative, cross-company service processes, hierarchical modeling structures, and precise modeling of processes and data.

B. *Planning and modeling requirements*

Due to these challenges, the modeling of service processes seeks for an adequate and precise integrated modeling approach as well as a precise system design for information systems. So far, no adequate modeling approach based on a modeling language that focusses the domain-specific context of service e-procurement is existing. Furthermore, a modeling approach for system design should be based on a formal modeling language to enable the following advantages [16]:

- adequate concept for the representation of domain-specific description of data and control flow,
- formal semantics of electronic business processes due to a formalized syntax,
- uniqueness of syntax and graphical descriptions for an easy understanding,
- expressiveness for a precise system modeling,
- mathematical foundation for the evaluation and sound proof of system design,
- analysis of information systems for properties like deadlocks, performance or the correctness of information systems,
- interoperability and vendor independence of the modeling language to support different modeling and analysis tools, and consideration of
- static and dynamic elements in service processes to describe the control flow and data flow.

III. State of the Art

Scientific literature reveals several approaches for service procurement with different emphasis and granularity. *FlexNet Architect* [17] offers reusable modules for the scenario-based modeling of hybrid value creation. For planning and modeling of hybrid value creation networks, the cooperation definition, actors, areas and information flows can be modeled. The *HyproDesign* [18] modeling language was developed for modeling customer-specific configurations and calculations of hybrid bundles of services and is based on a meta-model to describe variants and configurations. Single modules are described as semantic models via *Entity Relationship diagrams* [19]. Winkelmann and Luczak [20] propose a Petri net-based approach for the cooperative supply of industrial services by using *colored Petri nets* (*CPN*) [21]. Becker and Neumann [22] define central components like processes and activities, technical objects, contacts and service offers based on data models for the order transaction of technical services. Che et al. [23] are using *XML nets* [24] for modeling, execution and monitoring of cross-company business processes. Mevius and Pibernik [25] propose XML nets for the support of business processes for the *Supply Chain Process Management* (*SCPM*). Each of these approaches considers certain aspects of the description of services and service processes. However, none of them represents a comprehensive model for the description of service objects and service processes for industrial services based on a formalized approach. For the modeling of collaborative service processes and service objects in service transactions of service e-procurement, none of these approaches takes domain-specific characteristics of service processes for service e-procurement into account.

IV. Integrated Modeling Approach for Service Modeling *iServMod*

To meet the challenges and requirements described before, an integrated modeling approach based on a formal modeling language considering the domain-specific context of service procurement will be presented in this article. The integrated modeling approach *integrated Service Modeling* (*iServMod*) is based on Petri nets [26]. We use domain-specific extensions to Petri nets which will serve as a basis for a detailed and precise modeling of service processes (design time) to integrate service processes on the execution level (run time). In a first step, the modeling of *service objects* as static components of data schemas is presented. *Service nets* as dynamic components of business processes are developed: *Service nets* (*eSN*) are defined on basis of place/transition nets. In a second step, *high-level service nets* (*hSN*) based on XML nets are developed. This formal modeling approach will promote the following advantages:

- *increased transparency in service chains*: service processes lack transparency due to individual internal service processes of service providers and service providers.
- *precise modeling of collaborative service processes and data flow*: the precise modeling of service processes and service objects serves as a basis for high quality documentation and analysis. With an adequate modeling approach, internal service processes can be modeled separately and put together in service process models.
- *analysis of service processes*: application of analysis methods for the quantitative and qualitative evaluation of service process models serve as a basis for benchmarking.
- *integration of domain-specific context*: integration of service e-procurement context for its integration into information systems.
- *support of modeling and execution layers*: the modeling and the execution of service processes rounds up the comprehensive analysis of service process models.

Such a modeling approach results in improved efficiency (performance) and productivity of the service processes due to (1) the reduction of process costs, (2) reduction of process times, (3) reduction of process throughput time, (4) the improvement of process quantity, (5) improvement of process transparency, and (6) the increase of process flexibility.

A. Modeling concepts

Based on modeling concepts, service processes and corresponding service objects can gradually be modeled in a top-down approach – thus in detail and on a higher level of abstraction. This allows stepwise transformation of service processes into different formalization stages and enables hierarchical scaling and modularization. For the modeling of different formalized service processes, the *screen model* [27] serves as a modeling concept for Petri nets (see Figure 1).

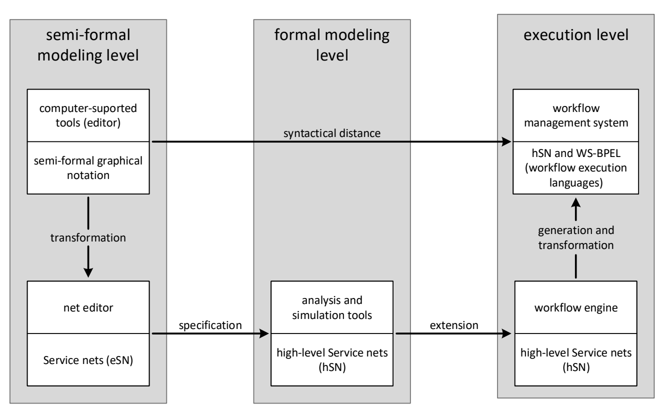

Figure 1. Screen model for gradual transformation of abstraction levels

This modeling concept supports four different modeling language types, beginning with informal modeling languages up to programming modeling language types for an automated execution (such as XML nets [24]). The screen model defines a gradual conversion of different formalization level of Petri net variants. Therefore, modeling languages used in our approach can be classified into four different groups:

- application-oriented, informal modeling languages support the documentation and visualization of business processes. Petri nets can be used to describe the implicit domain knowledge. The process models are described colloquially.
- application-neutral, semi-formal modeling languages provide more structure and a higher abstraction level due to a partial formalization of the business process models. Petri nets can be used to describe more detailed process models.
- formal, platform-independent modeling languages enable to model precise models by their power of description logic. Process models can be analyzed for the validation and plausibility checks. High-level Petri nets with individual tokens enable a precise and individual description of process and data concepts.
- machine-readable programming languages support the automated execution of business processes on the execution level. Service processes with integrated choreography and orchestration of web services can be automated. Web Service nets as extension of XML nets can be used and be transformed into executable code in WS-BPEL to call web services [28],[29].

The hierarchical and modular modeling of service processes is supported by the *layer model* [28]. The layer model supports the modeling of service processes on four different abstraction levels. In addition, modelers are guided from specific process phases to detailed service process descriptions within a top-down approach to describe a coordinated realization of service processes on the execution level. Thus, the overall structure of service processes will be improved by the hierarchical modeling layers. Software architectures based on the concept of service-oriented architectures may also be developed by means of such a hierarchical structure.

Service processes can be gradually refined. On a higher and abstract modeling level, a choreography is composed by several service process phases. The choreography resides on the highest level of abstraction and defines the logic of service processes. It is data driven and determines the order of lower level process phases. While the capsulated service processes are modeled by process phases on higher abstraction levels, distributed capsulated service processes are represented by process modules on a detailed level. Process modules incorporate complex electronic processes. These processes are implemented based on web service interfaces and thus can be orchestrated. Overall, from choreography to electronic processes on the lowest modeling layer, the logic consequence is the implementation of a service-oriented architecture. While process phases consist of process modules, they are linked by internal and cross-company process interfaces – which can be implemented by web services. Process modules encapsulate service processes and partial service processes.

Within service chains, a choreography of process phases serves as a link between distinct orchestrations of process modules and service processes. A combination of these coordination patterns process phases and process modules leads to (1) a choreography of process phases driven by the data exchange and (2) to the orchestration of their internal service processes. This results in a global, cross-company service process. In the domain of service e-procurement, service providers and service consumers collaborate for the execution of a procurement transaction. The choreography is being used to define the overall service process out of several orchestrated service processes with supervision of different process participants. Within our approach, the interaction of partners based on exchanged data can be pre-described and the order of is pre-defined. This results in valid orders of exchanged data. This approach defines the basis for agreements and contracts to define the necessary process interfaces in complex Business-to-Business (B2B) scenarios for the domain of service e-procurement.

B. Petri Net based modeling of service processes

Petri nets are a formal modeling language to describe, analyze, simulate and execute distributed, discrete systems. A Petri net is a based on a mathematical definition. Petri nets are bipartite graphs (consisting of the node types place and transition). They allow to capture static and dynamic characteristics of systems (by the concept of tokens). With different extensions to basic Petri nets, the modeling of different levels of formalization (precision) can also be accomplished. As a result, Petri nets may even be used for application-oriented, informal and semi-formal modeling. Besides the modeling of static and dynamic aspects, Petri nets offer to model limited capacities of places and anonymous tokens for modeling process objects.

With Petri nets allow to model typical process patterns such as sequence, iteration, alternative, concurrency, synchronization and further complex patterns as well as their combination. Dynamic properties like liveness, reachability, and soundness can be formally analyzed [30]. Petri nets are graphically represented by *tokens* (process objects), *places* (conditions), *transitions* (nodes for events) and *directed arcs* (arrows). Places are containers for tokens and pre- or post-conditions for transitions. Places represent local conditions and are static process components. Transitions are dynamic components and represent local state transitions [31]. For a formal, platform-independent and machine-readable modeling approach and an automated execution of business processes (execution level), *high-level Petri nets* allow a precise description of individualized tokens as well as the definition and formalization of further domain-specific process elements [32].

C. 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 form and define a static structure, which was recognized due to its identical re-appearing [33]. Thus, patterns represent best practice solutions to common problems and are a result of experiences and behavioral observation. They represent identical modes of thought, design fashions, behaviors or courses of action, which can be repeated and reproduced. Patterns can be observed in a lot of domains, maybe at first recognized in the domain of architecture. *Software design patterns* are introduced in the domain of software engineering. They are a general solution to solve a problem in programming. A design pattern provides a reusable architectural outline that may speed up the development of many computer programs [34]. It is considered as a recurrent solution template for software architecture and software development [35].

Design patterns [36] represent solutions to common design problems in a given context and improve software quality substantially. This can also help to reduce development costs. *Creational patterns* are used for the creation of objects independent of concrete implementation. *Structural patterns* support the design of software by providing templates for relations between classes. *Behavioral patterns* model the complex behavior of software and are used to increase flexibility. Nowadays, design patterns are widely used since they capture and promote best practices in software design like patterns for software engineering from Gamma et al. [34] and patterns for the enterprise integration scenarios of software applications from Hohpe and Woolf [37].

Business process design patterns describe business process models in a certain domain being harmonized with best practices. These patterns are also based on empirical knowledge how process activities should be executed. Business process design patterns are formalizing common structures of activities of process and data flows [38]. They are characterized by situations in courses of business and problems of the realization of modeling languages and implementation solutions [34]. Barros et al. [39] define bilateral and multilateral service interaction patterns, which allow emerging mechanisms such as choreography and orchestration to be benchmarked. Domain-oriented design patterns offer a flexible architecture with clear boundaries in terms of well-defined and highly encapsulated parts being aligned with the natural constraints of the considered domain [40].

D. Petri net based process patterns

As a well-established modeling language, Petri nets enable an integrated modeling of process and data structures. Additionally, analysis of software-based execution of business processes can easily be implemented with Petri nets. Van der Aalst and ter Hofstede [41] define fundamental *Workflow patterns* based on Petri nets to formalize requirements of workflow management 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 *TimeNET* [42], a software tool to model, analyze and control manufacturing systems based on colored Petri nets, *EXSPECT* [43], a repository of tool for standardized business processes in logistics and production or *CIMOSA* [44], the modeling and analysis of cross-company value chains. To formally model supply chains as business processes with an integrated sense and respond capability, Liu et al. [45] are using Petri nets to define basic patterns of supply chains. Schuster [33] proposes resource assignment patterns and defines higher-level resource nets for improved support of these patterns.

V. Modeling of Service Objects

Service objects describe object-oriented components, which are being handled within service processes. Service objects literally are data objects, which describe central *service* master data and transaction data. These data either characterizes a service consumer or service provider, materials, business documents, a service specification, a service order, a service invoice or any other service-relevant document needed for service transactions of service e-procurement. Hence, service objects represent input, intermediate and output objects of service processes. They convey economically relevant data, which is transformed, created or simply needed as execution support in electronic service processes (also known as service workflows).

As graphical modeling representation, service objects can be modeled with the *XML schema model* (*XSM*) [24]. *XSM* can be utilized as modeling construct to describe object structures in XML nets. *XSM* serves as a formal object description method to describe complex object structures in conjunction with organizational processes. Data structures in XML nets are associated with places to integrate them in the process flow. In this fashion, places combine structured service objects with a common schema (typification of places). The modeling constructs of XML schema models are element types represented by type classes and dependencies represented by association types. As an example, the service object *industrial service description* is shown in Figure 2. The example reveals a complex data structure, which can be applied in XML nets to describe typical objects in industrial service processes.

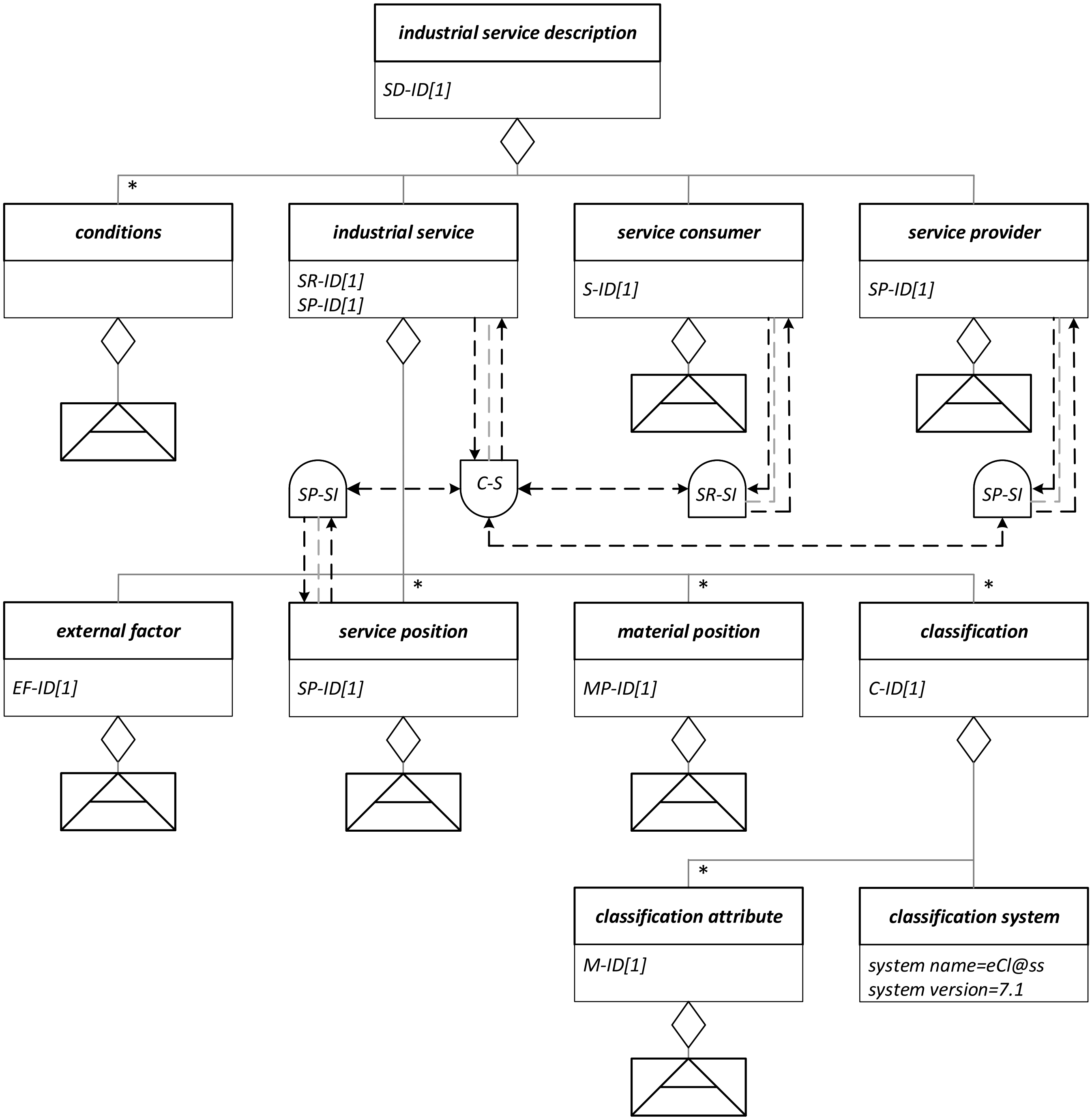

Figure 2. Service object industrial *service description*

VI. SERVICE PROCESS DESIGN PATTERNS

In the domain of service procurement, service providers and service consumers collaborate in a concrete instance of a service procurement process model. An instance of a service process model again is defined as choreography of specific service phases and comprises 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 Figure 3). The interaction between several partners for the procurement of services based on the exchanged data is described [46]. Only valid orders of data between partners may be defined.

Based on our review of scientific literature [47] and empirical case studies [49], 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*) which support the development of software architectures based on service-oriented architecture. These patterns define hierarchic structures and provide a structured concept for the modeling and implementation phase 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.

Service 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.

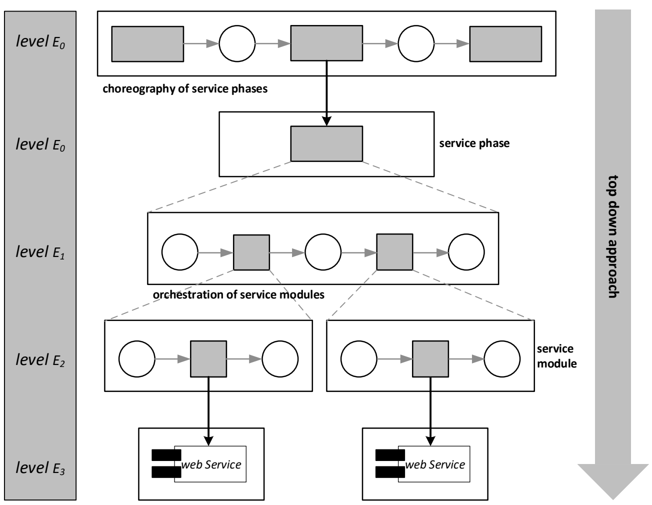

Figure 3. Choreography of service phases and orchestration of service modules on different abstraction levels

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 phase patterns and service module patterns. Based on this definition, we further develop these patterns based on XML nets [49], a high-level Petri net variant.

VII. SERVICE PHASE PATTERN (SPP) FOR THE CHOREOGRAPHY OF PROCESS FLOW AND DATA FLOW

The best practice for procurement of services is based on service procurement types. Service procurement types are characterized by a specific order of service process phases and a specific data flow to represent and manage cross-company interaction. 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 can be defined:

- A planned need of a service is required and a frame contract doesn't 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 the identified procurement types, we define a new pattern. *Service Phase Pattern* (*SPP*) are further introduced and described, the modeling support is presented, a definition is given and the composition and syntactical compatibility definition and as well as an example provided.

A. *Pattern description*

A standard pattern-based formal modeling approach based on best practice for service procurement has not been provided yet. Recurring service process phases as well as the validation of the correct order of service processes and service data for service process phases are not supported. The pattern offers the description of service procurement types by a choreography of service phases based on data flow. The valid composition of service process phases is ensured by the syntactical compatibility. The logic of process flow instances is determined as well. *SPP* are characterized by capsulated service procurement processes on a higher abstraction level.

B. *Modeling support*

This pattern can be modeled by Petri nets and high-level Petri nets and can be integrated into Petri net-based and high-level Petri net-based service processes. To support the pattern-based modeling approach, a domain-specific modeling extension for service process phases is necessary to support modelers in the design phase.

C. *Service Phase Pattern definition*

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 Figure 4).

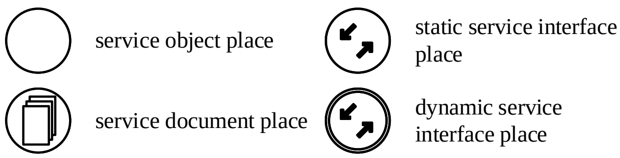

Figure 4. Service object specified places *SO*, *SD* and *SI*

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 Figure 5).

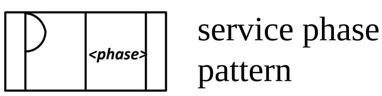

Figure 5. SPP modeled as Petri net

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 a coarsened 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 and also represent process patterns. *SPP* are defined based on the process and data flow of collaborative service e-procurement processes and consider the specific phases.

D. *Composition and syntactical compatibility*

In case of a composition of two service phases $t_{sp_a}^i$ and $t_{sp_b}^i$, input and output places are melted. 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*. The syntactical compatibility postulates that each interface of *SPP* is an output place S_{SP}^{OUT} of a *SPP* $t_{sp_i}^i$ and an input place S_{SP}^{IN} of a *SPP* $t_{sp_j}^i$. Syntactically compatible *SPP* do not necessary have completely overlapped interfaces. While the interfaces of two *SMP* during a composition are melted, the common non-empty subset of interfaces is modeled. A mandatory condition for the syntactical compatibility is the partly overlapping of interfaces. The partly overlapping of interfaces is also a sufficient condition for *SPP*.

E. *Example*

Based on a specific service procurement type, *SPP* can be configured to choreograph the data flow and process flow (see Figure 6).

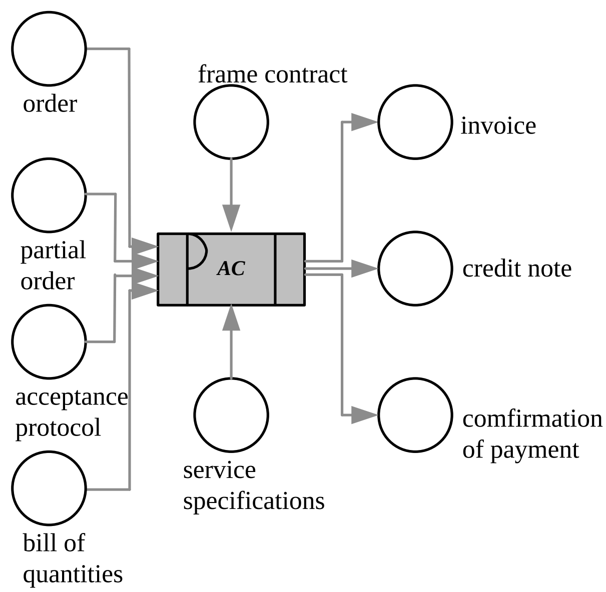

Figure 6. *SPP* example Accounting *AC* modeled as an XML net

Figure 7 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.

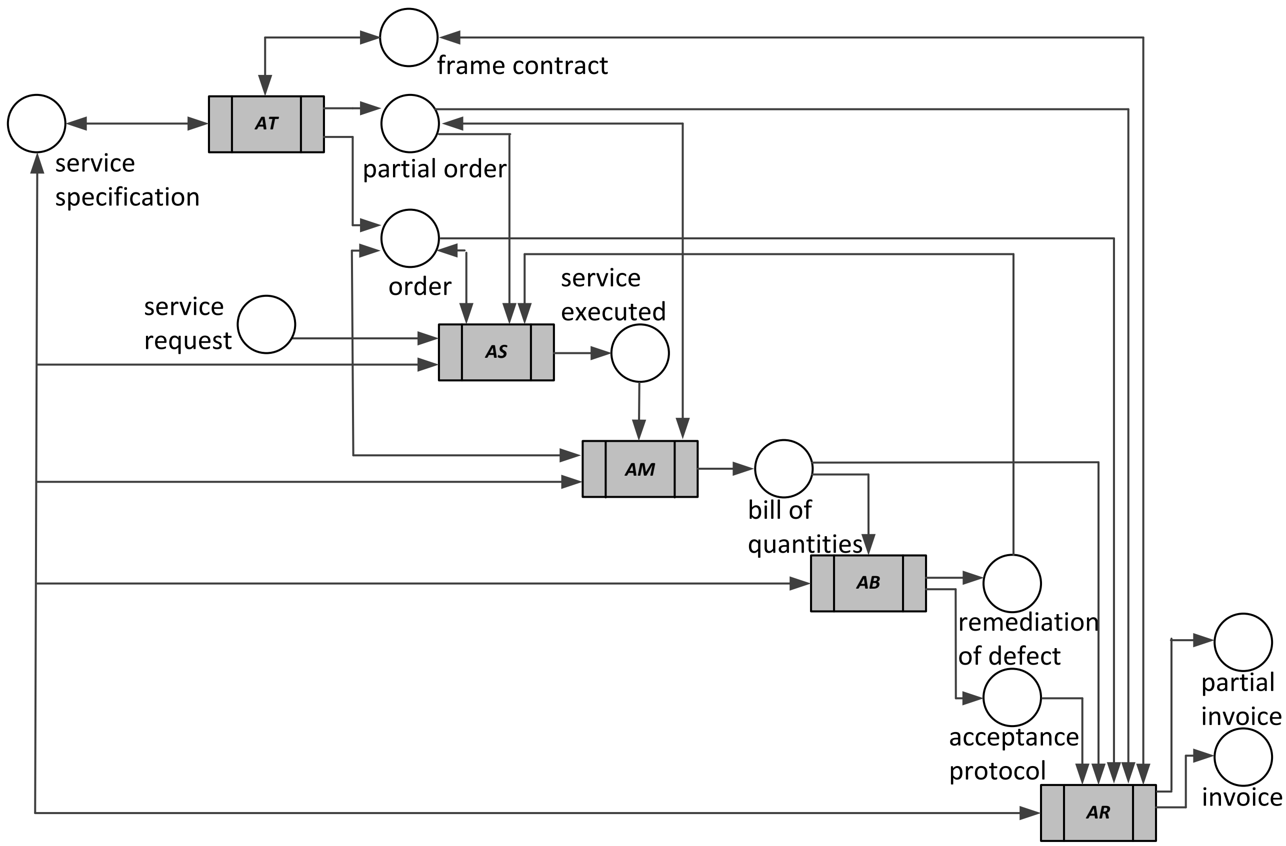

Figure 7. Service procurement type as choreography of *SPP* modeled as Petri net

F. Advantages of SPP

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 [50]. The definition of specific data and process sequences provides the basis for required process interfaces in complex B2B scenarios. Hence, *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 [50]. The definition of specific data and process sequences provides the basis for required process interfaces in complex B2B scenarios.

VIII. Service Module Pattern (SMP) for the Orchestration of Process Flow and Data Flow

Service procurement transactions and service processes are developed in a collaborative way: service providers and service consumers interact closely. Service processes and single service sub processes are characterized by a service provider or a service consumer. Based on service process phases, we define a new pattern. *Service Module Pattern* (*SMP*) are further introduced and described, the modeling support is presented, a definition is given and the composition and syntactical compatibility definition and as well as an example provided.

A. Pattern description

A standard pattern-based formal modeling approach based on best practice collaborative service processes in the domain of service procurement has not been provided yet. Recurring service processes as well as a collaborative and modularized modeling approach based on the process participants for service processes are not supported. The pattern offers the description of collaborative service processes and service sub processes by the orchestration of service modules based on data flow. The valid composition of service process modules is ensured by the syntactical compatibility. The predefined 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. *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. *Modeling support*

This pattern can be modeled by Petri nets and high-level Petri nets and can be integrated into Petri net-based and high-level Petri net-based service processes. To support the pattern-based modeling approach, a domain-specific modeling extension for service processes is necessary to support modelers in the design phase.

C. *Service Module Pattern definition*

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* are defined as self-contained collaborative service processes of one collaboration participant (service provider 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 Figure 8). The set of *SI* and *SD* are input and output places of *SMP*.

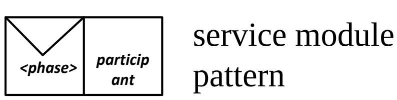

Figure 8. SMP modeled as Petri net

SMP are defined based on high-level Petri nets. *SMP* represent coarsened collaborative service processes of one process participant. The collaborative service process consists out 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 *SMP* is built by a coarsened service net and consists out 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 *SMP*. The internal structure of a *SMP* fulfills the requirements of a workflow net [51] and soundness criteria [52]. A *SMP* interface $S_{sm_1}^{IN/OUT}$ of one *SMP* t_{sm}^1 can be melted with the *SMP* interface $S_{sm_2}^{IN/OUT}$ of another *SMP* t_{sm}^2. The set of *SMP 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 *SPP* of the set of *SPP TSP*. The composition of *SMP* causes the melting of the common set of interface places.

D. *Composition and syntactical compatibility*

The syntactical compatibility of *SMP* enable the composition of *SMP*. Syntactically compatible *SMP* have completely overlapping process interfaces. The syntactical compatibility postulates that each module interface of *SPP* is an output place S_{SM}^{OUT} of a *SPP* $t_{sm_i}^1$ and an input place S_{SM}^{IN} of a *SPP* $t_{sm_j}^2$. For every output place, it exists a corresponding input place. Syntactically compatible *SMP* have completely overlapped interfaces. While the interfaces of two *SMP* are melted during a composition, the common non-empty subset of interfaces is modeled. A mandatory condition for the syntactical compatibility is the partly overlapping of interfaces. The completely overlapping of interfaces is also a sufficient condition for *SMP*.

E. *Example*

Based on a specific *SPP*, a capsulated service process can be further detailed into *SMP* to orchestrate the data flow and process flow (see Figure 9). The input place $S_{sm_1}^{IN_1}$ and the output place $S_{sm_1}^{OUT_1}$ of service process hSN'_{sm_1} and hSN'_{sm_2} of a *SMP* are melted and create a common internal place S_{sm_3}.

F. *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.

IX. Service Nets eSN

For the modeling of *eSN*, we utilize *place/transition nets* (*P/T nets*) to support initial modeling phases. Places and tokens of these nets represent the current status of a service chain. In addition, we define domain-specific process interface place types and process transition types to standardize the modeling approach. Static interface places between service processes and dynamic interface places for capsulated interface processes are defined.

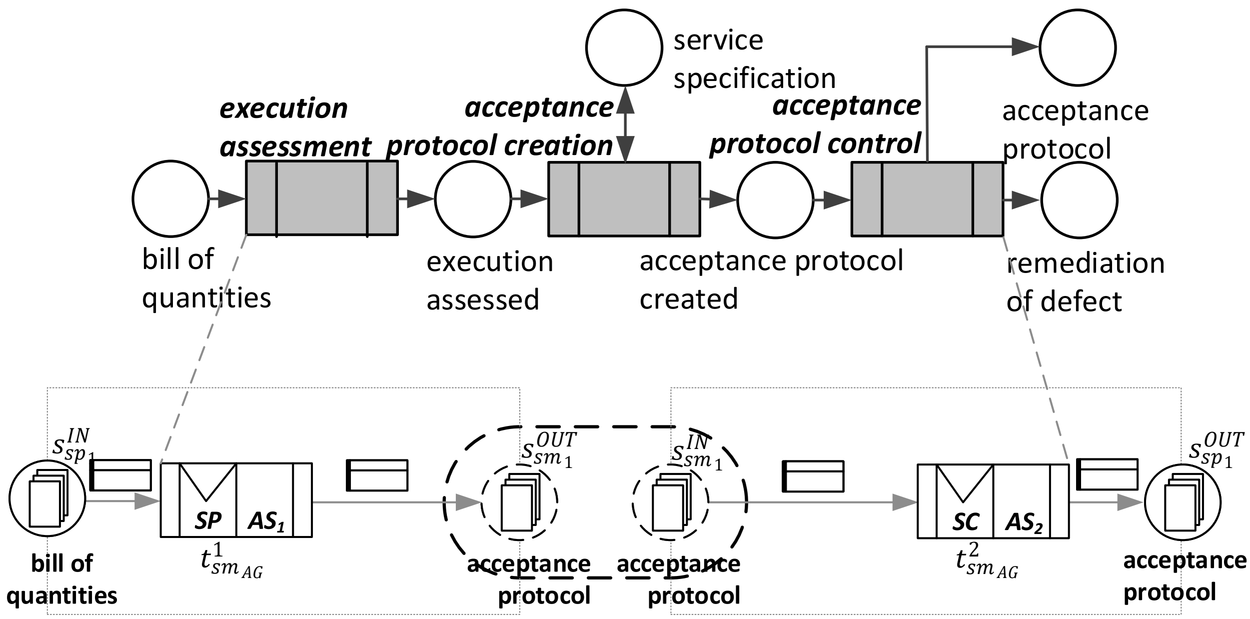

Figure 9. Composition of *SMP* modeled as an XML net

Therefore, we define the interface place types *service object places* (*SO*), *service interface places* (*SS*) and *service document places* (*SD*). *SO* places are containers for general service objects. *SS* places are internal and cross-company interfaces for the data flow. *SD* places serve as containers for service document types. For transition concepts, *Service Phase Patterns* are defined to reflect the specific process phases in service procurement.

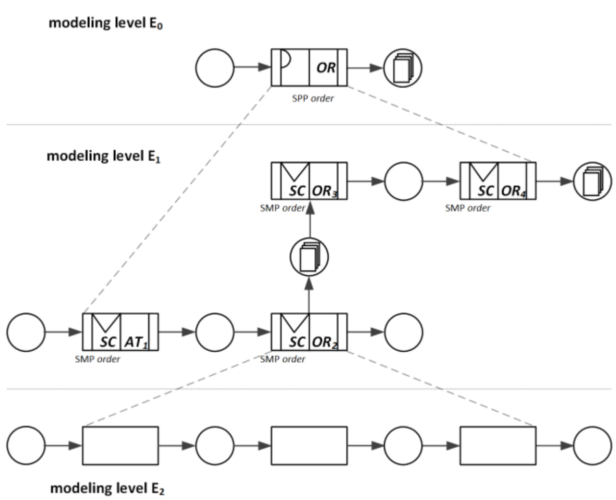

Figure 10. Hierarchic structuring of service processes

Service Module Patterns represent service processes of a collaboration participant (service provider or service consumer). *SPP* and *SMP* represent both a black box for subprocesses, which can be modeled separately in further detailed service processes. A *SPP* consists out of several *SMP*. *SPP* and *SMP* represent and apply hierarchical modeling concepts to support the layer model. In an early modeling stage, *SPP* and *SMP* act as a place holder for concrete service processes, which can be modeled at a later point of time. These internal (private) service processes are modeling by using pools. Specific *SMP* are assigned to *SPP* to structure service processes and organize them into a hierarchy (see Figure 10). As an example, a single collaborative service process between a service provider and a service consumer is shown in Figure 11.

X. High-level Service Nets hSN

The use of P/T net concepts accompanies a couple of disadvantages: the semantic correctness cannot be checked, domain-specific modeling constructs are not supported, communication and information concepts are not designed, the structured hierarchical modeling is not supported and tokens cannot be specified individually. Thus, we introduce modeling extensions of *eSN* with transfer to high-level Service nets based on XML nets with individual tokens. High-level Service nets (*hSN*) are based on XML. Operational sequences and the data flow are based on XML, tokens are represented by complex structured XML objects. All activities correspond to operations on XML documents. *hSN* are characterized by domain-specific extensions, and individual tokens. Furthermore, within *hSN*, the phases of service chains are standardized.

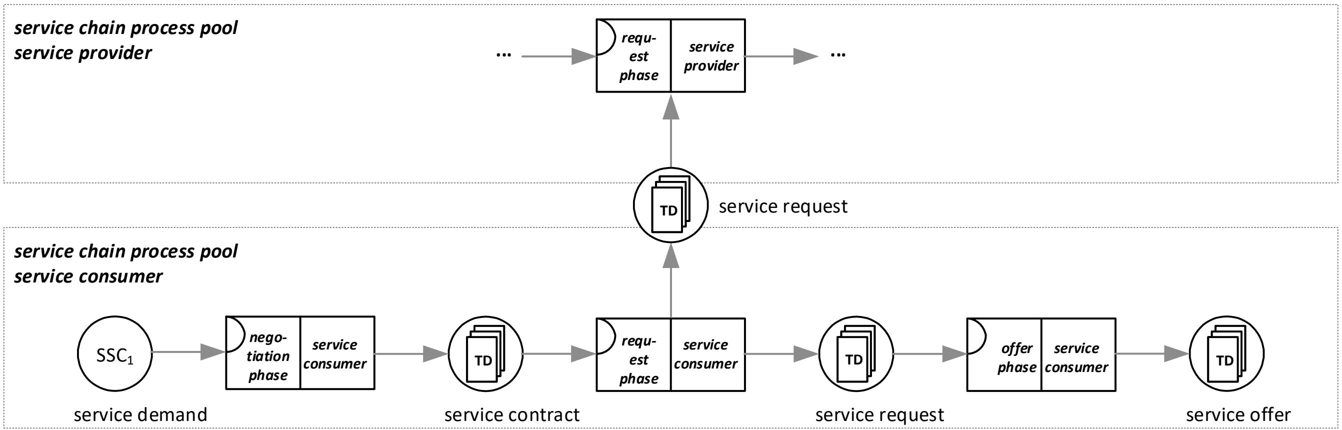

Figure 11. Single collaborative service process with pools of service provider and service consumer

A high-level Service net *hSN* is defined as follows:

A Service net is defined as a tuple $hSN = (S, SSO, SSI, SSD, T, TSP, TSM, F, \Phi, I_S, I_T, I_F, I_{SSO}, I_{SSI}, I_{SSD}, M_0)$, where

1. $XN = (S, T, F, \Phi, I_S, I_T, I_F, M_0)$ is an XML net.
2. $\Phi = (\mathrm{E}, \mathrm{FKT}, \mathrm{PRE})$ is a structure consisting of a non-empty and finite individual set E of Φ, a set of formula and term functions FKT defined on E, and a set of predicates PRE defined on E.
3. The set of places is structured in the sets process object places SPO and service object specific places SPS. The set of SPS is further structured in the sets of service object places SSO, service interface places SSI and service document places SSD.
4. The set of transitions is structured in the sets of process activities TPA and service process activities TPS. The set of TPS contains of *SPPs* TSP and *SMPs* TSM. The set of TPS is defined as a real set of transitions T: $TPS \subseteq T$.
5. I_{SSO} is the function that assigns a valid XML schema as a place typification to each place $s_{so} \in SSO$.
6. I_{SSI} is the function that assigns a valid XML schema as a place typification to each place $s_{si_s} \in SSI$.
7. I_{SSD} is the function that assigns a valid XML schema of a service documents type *j* as a place typification to each place $s_{sd_j} \in SSD$.
8. I_T is the function that assigns a predicate logical expression as inscription to each transition on a given structure Φ and the set of variables, which are contained on all adjacent arcs.
9. I_F is the function that assigns a valid XSLT expression to each arc, which is conform to the adjacent XML scheme.

Electronic service processes and their data flow can be precisely modeled, analyzed, simulated, executed and maintained. For the process interface place types, places are typified based on the domain-specific context. *SPP* are represented by coarsened transitions with capsulated service processes based on *SMP*. *SMP* are also coarsened transitions and contain capsulated service processes of one process participant (internal service process). A *SMP* is defined by an internal structure and communicates with other modules based on process interface places. *SPP* and *SMP* contains typified input and output places *SO*, *SS* and *SD*. *SPP* are defined with specific typified places. The *SPP offer* ($t_{sp_{AG}}$) is modularized by two *SMPs* (Figure 12).

The service process consists of the service process hSN'_{sm_1} of the *SMP* $t^1_{sm_{AG}}$ and hSN'_{sm_2} of the *SMP* $t^2_{sm_{AG}}$. hSN'_{sm_1} of the service provider is modeled as an XML net. The representation of XML filter schemes FS_i, transition inscriptions TI_i and place type definition ST_i are not modeled. The service process of the service requester is represented by the *SMP* $t^2_{sm_{AG}}$. The place $s^{IN_1}_{sp_1}$ and the place $s^{OUT_1}_{sp_1}$ are the in- and output of hSN. The input place $s^{IN_1}_{sm_1}$ ($s^{IN_1}_{sp_1}$) and the output place $s^{OUT_1}_{sm_1}$ as the input place $s^{IN_1}_{sm_2}$ and the output place $s^{OUT_1}_{sm_1}$ ($s^{OUT_1}_{sp_2}$) are module interface places of the *SMP*.

A. *Modeling of distributed service processes based on SOA*

Service processes can be modeled as dynamic interface processes and be executed by web services in *service-oriented architectures* (*SOA*). *iServMod* supports the modeling and analysis of distributed service processes based on web services by *Web Service nets* (*WSN*) [53]. A web service is considered as an implementation of a local service process (interface process). A distributed service process can be realized by the composition of web services. Input and output messages represent the data flow in Web Service nets.

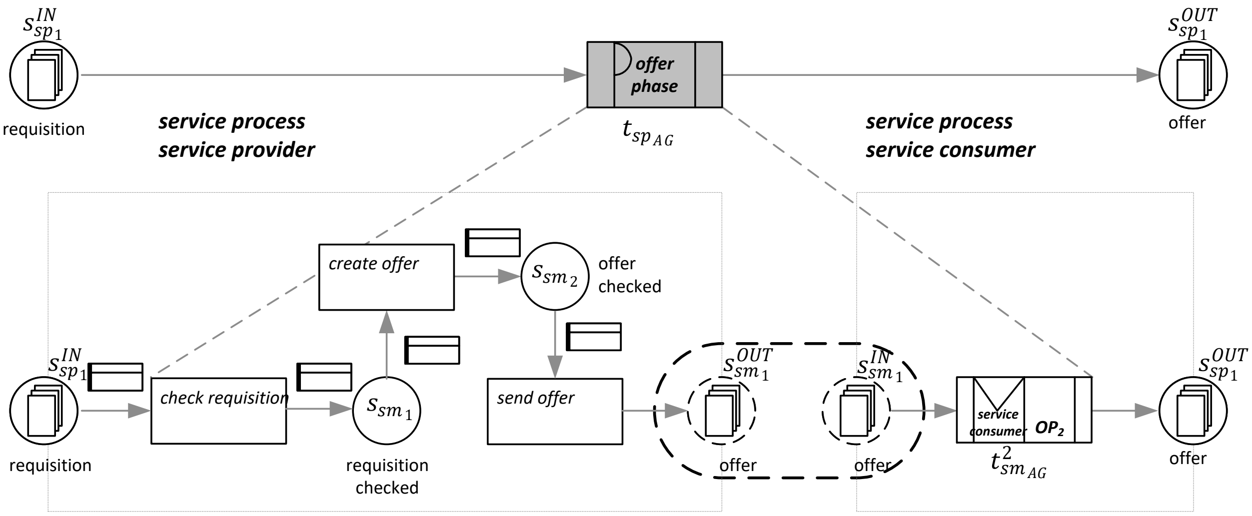

Figure 12. Modularization of a *SPP* by a *SMP*

Web Service nets also support the composition concepts like the orchestration and choreography. The example *service process validation of incorrect orders* is models as an abstract Web Service net with web service selection (Figure 13).

B. Transformation and execution of Web Service Nets

Based on the precise modeling of all aspects of service processes with Web Service nets, *Web Service Business Process Execution Language* (*WS-BPEL*) elements can be derived to use standardized web service technologies like interface description (*WSDL*), protocols (*SOAP*) and mechanisms for service discovery.

Web service process models [54] can be modeled and the transformation of Web Service nets into executable WS-BPEL code is based on control flow and data flow structures. An XML-based notation and semantics for the description of the behavior of service processes enrich Web Service nets based on web service calls. In this context, the WS-BPEL model is called *web service process model*. A web service or several web services describe the behavior and the interaction of process instances, process participants and resources using web service interface places of the Web Service net. Specific structures and elements can be identified and transformed into equivalent XML and WS-BPEL structures. A detailed transformation of Web Service net structures into WS-BPEL code with a transformation algorithm is defined in [29].

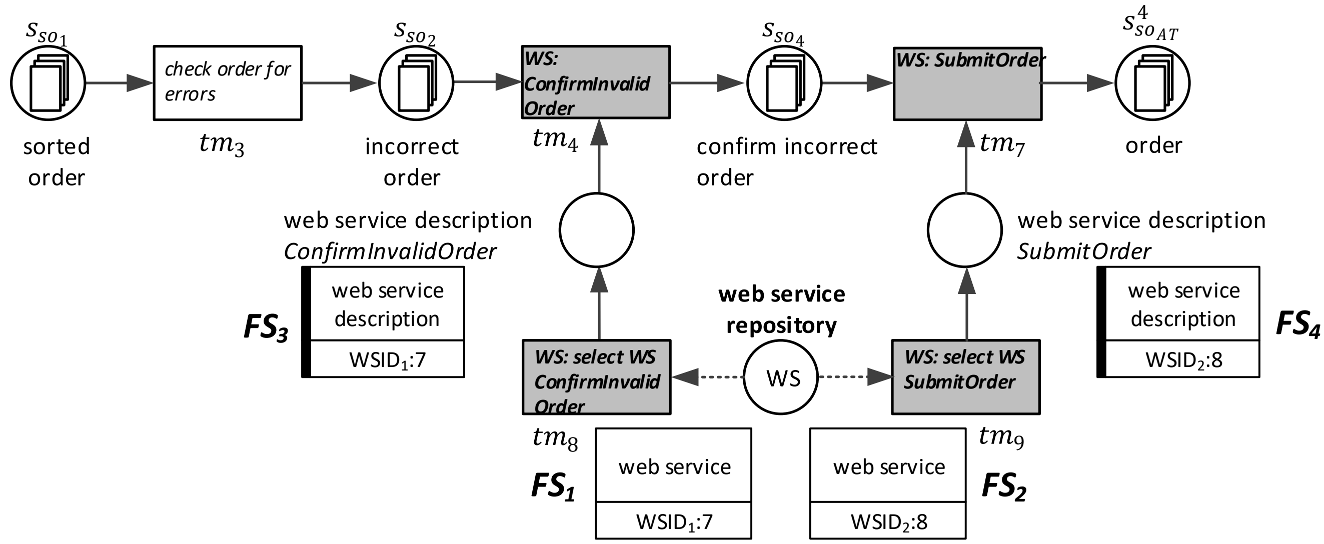

Figure 13. Web Service net with web service selection

XI. Evolutionary Procedure Model for the Integrated Modeling of Service Processes

We propose the *Evolutionary Procedure Model* (*EPM*) for the development of process-oriented information systems based on service processes. *EPM* is structuring the development and the application of the integrated modeling method *iServMod* for the domain service e-procurement. *EPM* incorporates the developed methods, models and modeling concepts of the integration service modeling approach. The evolutionary procedure model consists out of several phases, which can be aggregated to the main phases modeling, model analysis and implementation. The phases of the *EPM* are presented (Figure 14).

A. Project initialization

The phase project initialization defines and plans a project for the development of an information systems. The scope of the project is defined, the involved organization and teams are identified. The project parameters project budget, available resources, and temporal frame conditions are defined and a project plan is created. The project-specific targets are determined and are prioritized. As a first step, a process landscape is created, which represents the topmost hierarchical layer for a service process model for the gradual modeling of service processes.

Single service processes are graphically represented amongst their reciprocities. The modeling language group of application-oriented, informal modeling language is used. On an abstract level, relevant service processes are represented by single transitions and their relations based on places, which represent interfaces.

B. Functional modeling design

The existing service processes are modeled in a semi-formal way. For the adequate modeling of process instances, a detailed process structure of service process models is developed. Interfaces and data objects are identified and described. The description of service processes is based on an informal modeling level in a first modeling round. As applications-oriented, informal modeling language, Service nets can be used. The explicit interface description is a central modeling aspect. Interfaces are defined as input and output places, which represent input service objects and output service objects. Roles and resources are identified and assigned to transitions. For the functional modeling design, roles and resources are annotated and modeled by resource places. Resources are represented as anonymous, non-typified tokens in places.

C. Detailed modeling design

The service processes are modeled top-down on different abstraction layers based on detailed levels. The layer model [29] supports the hierarchical modeling. The modeling language group of application-neutral, semi-formal modeling language of Service nets is used. The definition of hierarchies and a modularized concept is based on process phase patterns and process module patterns. The target of the modularization of service processes is the reduction of complexity of service process modeling by a break down into small process units [55]. As soon as the process phases are identified, process modules are modeled successively as refinement. Process modules build up the corresponding collaborative service process of a service consumer or service provider. The input and output of a service process of an abstraction layer is derived from the descriptions of abstraction layer, which is dominated by it. The hierarchical modeling approach supports the collaborative modeling since service process models are modeled in a distributed way and can be integrated in a structured way. This decentral modeling approach incorporates several autonomous modelers, which can model partial service processes independently of each other. The procedure model supports the consistent, individualized model creation and the integration of partial service process models. Interfaces of service processes are illustrated on each modeling layer and data objects are identified.

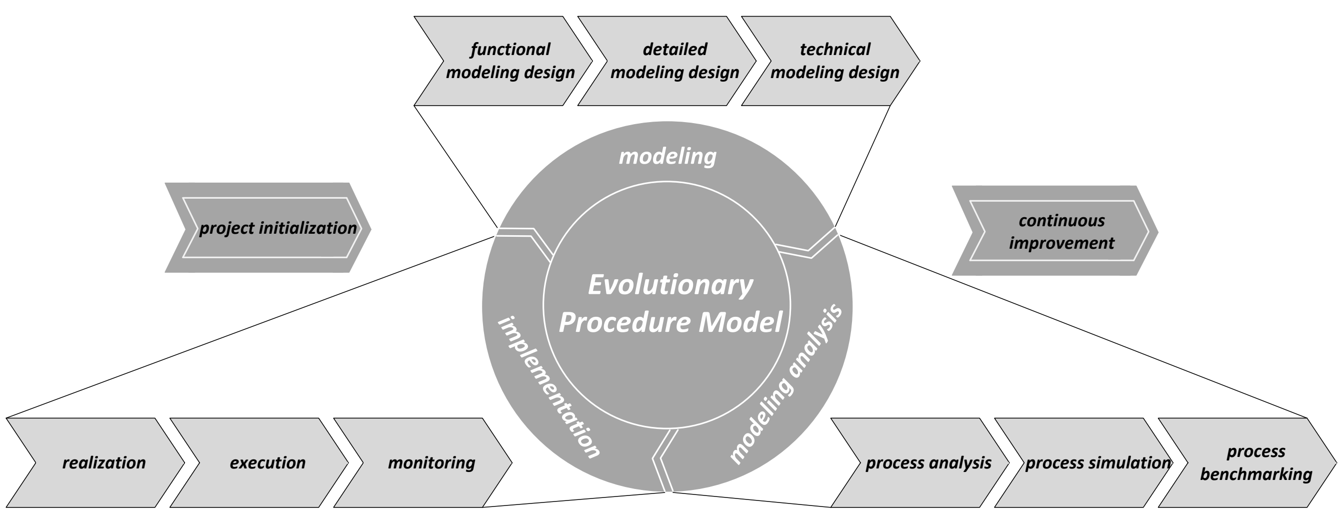

Figure 14. EPM for the development of process-oriented information systems

D. Technical modeling design

The technical modeling design is determined and incorporated. New requirements decide whether existing information systems are still used, or will be changed or replaced. Existing infrastructures of collaborating service providers and service consumers are analyzed in detail and relevant systems are being documented. The analysis data will be used for the technical modeling design. Modeled hierarchical service process models will be enriched by technical modeling details. Out of the group of formal, platform-independent modeling languages, high-level Service nets are used.

They support the modeling typified places for the detailed description of interfaces and data objects. Data objects are modeled based on XML schemata. Static and dynamic interfaces for coupling of collaborative service processes are differentiated. The modeling of *SPP* and *SMP* detail the typified places. The detailed service process of a *SMP* are modeled by (semi-)automated service processes.

E. Process analysis

For the process analysis, qualitative and quantitative analysis methods are used. Qualitative process analysis checks for the logical behavior of service process models to exclude modeling errors. Quantitative analysis methods define process targets to prevent exceptions.

F. Process simulation

The quantitative analysis method of simulation serves as an efficient method for the validation of potential behavior of service process models compared to the behavior of the intended service processes. The complex system of service process models is analyzed by process simulation.

G. Process benchmarking

The simulation-based process benchmarking can be used to discover the root causes and potentials of improvements by a systematic analysis of performance differences [4].

H. Realization

The modeled service process models are implemented into process-oriented information systems. The modeling language group of machine-readable programming language is used. Based on service-oriented architectures, web services can be modeled and can be related to corresponding WS-BPEL implementations of service processes. Web services are implemented and be called and executed with WS-BPEL after the implementation of service processes [28].

I. Execution

Electronic service processes are executed for the commence operations. Distributed, collaborative service processes of all involved parties are integrated to improve the execution of service processes.

J. Monitoring

The operative service processes must be monitored continuously. The monitoring serves as a base for a continuous improvement. As soon as potentials for improvement are identified, single phases of the procedure models are traced back and service process models are improved iteratively.

XII. Application of iServMod in use cases

The integrated modeling approach *integrated Service Modeling (iServMod)* serves as an adequate method for a precise modeling and analysis of service processes. The advantages of *iServMod* increase the value of business process simulation and business process benchmarking. The simulation of service processes based on key performance indicators reveals gaps and weaknesses. The execution of a process benchmarking identifies differences of relevant factors like throughput times, resource assignments or cost items. The causes of performance gaps can be analyzed. For the modeling and simulation of service process models, the software tool *Horus* [56] was enhanced by the new modeling extensions of Service nets and used for a software based simulation. As business process simulation method, a discrete event driven business process oriented simulation was used [57]. The strengths of the independent simulative analysis are the possibility of a "*playground*" by simulating different process alternatives. Evaluation of simulation results can shed light on correlations of system parameters at build time and can be used to develop action strategies [4]. Unlike analytical procedures, the simulation can be used for the analysis of large systems. Based on benchmarks, performance gaps can be quantified. Redundant service processes and non-value creating activities as well as automation potentials for service processes can be identified and the error data is reduced. Also, the cost-effectiveness of service processes can be ensured.

The integrated modeling approach *iServMod* has been successfully applied in a research projects in the domain of service procurement [5]. The service process models of 18 use cases between six service suppliers and four service requesters were analyzed, modeled, simulated and benchmarked [4],[10],[58]. Service process models were modeled with high-level Service nets. The modeling of Service nets was based on a reference process model [5] to structure and align the individual service processes. *iServMod* supported the precise modeling of service processes and service objects in a syntactical correct and semantic formal way. The data flow could be modeled based on XML. Service processes could be modeled in a hierarchical modeling approach based on different abstraction levels to support modeling user groups with different modeling experiences.

Service processes could be modeled top down from high-level process description to detailed service processes as workflows using web services and representing and supporting a further implementation in information systems. *SPPs* and *SMPs* allow for reusability of pre-defined concepts by assuring the syntactical and semantic compatibility in service process models. The evaluated uses cases were compared pairwise for benchmarking by applying *EPM*.

XIII. Conclusion and Outlook

We presented the integrated modeling approach *iServMod*. *iServMod* supports integrated modeling of service processes and service objects based on formalized modeling techniques. Additionally, *iServMod* offers an adequate modeling approach for the precise analysis and implementation of service chains. *iServMod* is focusing on collaborative service processes, which are modeled independently by different companies and their domain experts (modelers). It furthermore supports the domain-specific requirements of service e-procurement in service chains.

The presented modeling concepts enact different formalization levels, starting from a semi-formal description of service process models up to formalized and executable service process models. This includes a hierarchical order of service processes typically modeled in a top down approach. The patterns *SPP* and *SMP* enable for both the choreography of service phases and the orchestration of service modules in collaborative cross-company service chains. These 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 also defined within our Petri net based modeling approach for service e-procurement. While *eSN* serve as an initial modeling approach, further analysis is based on *hSN*, which enable an integrated modeling of service processes and service objects for the design of information systems.

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 and 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*,
- a step-wise transformation of modeling different *formalization levels*,
- the support of *distributed business processes* based on service-oriented architectures (SOA), and
- syntactic and semantic correctness to verify service process models.

The pattern-based modeling approach is concluded by integrated description of web service calls to implement sound information systems at execution level. The syntactic domain-specific extensions both of service object-specific process interface place types and process transition types enable a precise hierarchical modeling of process participants, modular service processes, e-procurement service phases, pre-defined process patterns, interfaces and service data objects.

Our pattern-based approach is derived from typical use cases of service e-procurement of industrial services. Hence, the integrated modeling and design approach was evaluated with real-life case studies [5]. The service process models have been modeled and analyzed. The developed extension of the software tool *Horus* supports the overall modeling of collaborative service chains.

As next steps, we intend to extend the verification of these patterns. We will evaluate our pattern-based approach by analysis of further service e-procurement use cases. Furthermore, the adoption and application of *iServMod* in different domains and hence different types of service chains are planned. E-procurement of other service domains will also be analyzed and validated. We also intend to formulate experiments in other service domains to ensure the general usability, the level of detail and completeness of the defined patterns. In addition, the pattern-based modeling approach can be transferred to further service process types besides procurement. Analog service process types are repair orders, return orders and warranty service orders. The pattern based modeling approach will also be used for simulation and benchmarking of collaborative service processes of different service provider and service consumer combinations. We also strive to further evaluate the performance, quality and efficiency of this approach together with several leading companies in the domain of industrial services. Thus, we foresee the evaluation of *iServMod* by a survey of domain specific users.

References

[1] M. Herfurth and T. Schuster, "Integrated Modeling Approach iServMod for Modeling, Analysis and Execution of Collaborative Service Processes in Service Chains," COLLA 2016, The Sixth International Conference on Advanced Collaborative Networks, Systems and Applications, Spain, pp. 70-77, 2016.

[2] 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.

[3] R. F. Lusch and R. Nambisan, “Service innovation: a service-dominant logic perspective,” MIS Quarterly volume 39 (1), pp. 155-175, 2015.

[4] M. Herfurth, T. Schuster, and P. Weiß, “Simulation-based Service Process Benchmarking in Product-Service Ecosystems,” Proceedings of Collaborative Systems for Reindustrialization: 14th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2013, Germany, pp. 40-47, 2013.

[5] P. Weiß, M. Herfurth, and J. Schumacher, “Leverage Productivity Potentials in Service-oriented Procurement Transactions: E-Standards in Service Procurement,” Proceedings of RESER 2011, pp. 1-20, 2011.

[6] A. Eggert, C. Thiesbrummel, and C. Deutscher, “Heading for new shores: Do service and hybrid innovations outperform

product innovations in industrial companies?," Industrial Marketing Management, volume 45, pp. 173-183, 2015.

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

[8] J. Johanson and L. G. Mattsson, "Internationalisation in Industrial Systems — A Network Approach," in Knowledge, Networks and Power, pp. 111-132, 2015.

[9] T. Baltacioglu, E. Ada, M. Kaplan, O. Yurt, and C. Kaplan, "A New Framework for Service Supply Chains," Services Industry Journal, volume 27, No. 2, pp. 105-124, 2007.

[10] M. Herfurth, T. Schuster, and P. Weiß, "E-Business for Services: Simulation-based Analysis of E-Business Solution Concepts for Service Procurement Scenarios," eChallenges e-2012 Conference Proceedings, Paul Cunningham and Miriam Cunningham (editors), IIMC International Information Management Corporation, 2012.

[11] 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.

[12] 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.

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

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

[15] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," MIS Quarterly, 28(1), pp. 75-105, 2004.

[16] W. M. P. van der Aalst, "The Application of Petri Nets to Workflow Management," The Journal of Circuits, Systems and Computers, volume 8(1), pp. 21-66, 1998.

[17] J. Becker, D. Beverungen, R. Knackstedt, M. Matzner, O. Müller, and J. Pöppelbuß, "Flexible information system architectures for hybrid value creation networks (FlexNet)," work report of the institute for business informatics no. 130, Westpahlian Wilhelms University Muenster, 2011.

[18] J. Becker, D. Beverungen, R. Knackstedt, and O. Müller, "Conception of a modeling language for the software supported modeling, configuration and evaluation of hybrid bundles of services," in: O. Thomas and M. Nüttgens (editors), "Service modeling," Physica, pp. 53-70, 2009.

[19] P. P.-S. Chen, "The entity-relationship model - towards a unified view of data," ACM Transactions on Database Systems 1 (1976) 1, pp. 9-36, 1976.

[20] K. Winkelmann, "Prospective evaluation of cooperative service delivery of industrial services in manufacturing systems engineering via simulation of Petri nets," Dissertation RWTH Aachen, 2007.

[21] K. Jensen, "Coloured Petri Nets: A high-level Language for System Design and Analysis," LNCS volume 483, Springer, 1990.

[22] J. Becker and S. Neumann, "Reference models for workflow applications of technical services," in: H.-J. Bullinger, and A.-W. Scheer, "Service Engineering", Springer, pp. 623-647, 2006.

[23] H. Che, M. Mevius, Y. Ju, W. Stucky, and R. Trunko, "A Method for Interorganizational Business Process Management," Proceedings of the IEEE International Conference on Automation and Logistics, China, 2007.

[24] K. Lenz and A. Oberweis, "Interorganizational Business Process Management with XML Nets," in: Petri Net Technology for Communication Based Systems, LNCS 2472, pp. 243-263, 2003.

[25] M. Mevius and R. Pibernik, "Process Management in Supply Chains - A New Petri-Net Based Approach," Proceeding of the 37th Hawaii International Conference on System Sciences (HICCS'04), IEEE Computer Society, 2004.

[26] W. Brauer, W. Reisig, and G. Rozenberg, "Petri Nets. Part I and Part II," Lecture Notes in Computer Science 254 and 255, Springer Verlag, 1987.

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

[28] M. Herfurth, T. Karle, and F. Schönthaler, "Reference Model for service oriented Business Software based on Web Service Nets," Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems SIGSAND-EUROPE, pp. 55-69, 2008.

[29] M. Herfurth, T. Karle, and R. Trunko, "Model Driven Implementation of Business Processes Based on Web Service Nets," Proceedings of the 2008 SIWN Congress 2nd International Conference on Adaptive Business Systems (ABS 2008), Glasgow, pp. 32-38, 2008.

[30] W.M.P. van der Aalst and K. van Hee, "Workflow Management: Models, Methods, and Systems," MIT Press, 2004.

[31] W. Reisig, "Petri nets – an Introduction," Springer, 1982.

[32] 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.

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

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

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

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

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

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

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

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

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

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

[47] 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.

[48] 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.

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

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

[51] W.M.P. van der Aalst, "Modeling and Analyzing Interorganizational Workflows," in: L. Lavagno and W. Reisig (editors), Proceedings of the International Conference on Application of Concurrency to System Design (CSD'98), IEEE Computer Society Press, pp. 1-15, 1998.

[52] 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.

[53] A. Koschmider and M. Mevius, "A Petri Net based Approach for Process Model Driven Deduction of BPEL Code," OTM Confederated International Conferences, Agia Napa, Cyprus, Springer, pp. 495-505, 2005.

[54] A. Martens, "Analyzing Web Service based Business Processes," Proceedings International Conference on Fundamental Approaches to Software Engineering (FASE), volume 3442 of LNCS, Scotland, pp. 19-33, 2005.

[55] W. Kersten, B. Koeppen, C. Meyer, and E.-M. Kern, "Reduction of process complexity through modularization," Industry Management, volume 21 (4), pp. 11-14, 2005.

[56] Y. Li and A. Oberweis, "A Petri Net-Based Software Process Model for Developing Process-Oriented Information Systems," Proceedings of the 18th International Conference on Information Systems Development (ISD2009), China, pp. 27-39, 2009.

[57] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, "Discrete-Event System Simulation," Pearson Education, 2005.

[58] M. Herfurth, T. Schuster, and P. Weiß, "Decision Support Based on Simulation Analysis of Service Procurement Scenarios," eChallenges e-2013 Conference Proceedings, Paul Cunningham and Miriam Cunningham (editors), 2013.