

Title: Initial models and mechanisms for quantitative analysis of correlations between KPIs, SLAs and underlying business processes

Authors: Tilburg, UOC, USTUTT

Editor: Branimir Wetzstein (USTUTT)

Reviewers: Qing Gu (VUA)
Sonja Zaplata (UniHH)

Identifier: Deliverable # CD-JRA-2.1.2

Type: Deliverable

Version: 1.0

Date: 13 March 2009

Status: Final

Class: External

Management Summary

In this deliverable we present initial models and mechanisms for quantitative analysis of correlations between KPIs, SLAs and underlying business processes. We use service network (SN) models for quantitative analysis based on KPIs and SLAs, which enables strategic decisions for participants such as determination of optimal product prices or outsourcing decisions. In order to perform the analysis on the SN abstraction level and implement its results in operational business processes, SNs have to be connected to the BPM stack. We therefore introduce the SN4BPM architecture describing an enhanced BPM layering and lifecycle where SNs constitute a separate layer on top of the established BPM stack. In that context, we describe in particular a model-driven approach to generating abstract business process models from Service Network Models and vice versa. Finally, we deal with monitoring in the cross-organizational setting of service networks.

Members of the S-Cube consortium:

University of Duisburg-Essen (Coordinator)	Germany
Tilburg University	Netherlands
City University London	U.K.
Consiglio Nazionale delle Ricerche	Italy
Center for Scientific and Technological Research	Italy
The French National Institute for Research in Computer Science and Control	France
Lero - The Irish Software Engineering Research Centre	Ireland
Politecnico di Milano	Italy
MTA SZTAKI – Computer and Automation Research Institute	Hungary
Vienna University of Technology	Austria
Université Claude Bernard Lyon	France
University of Crete	Greece
Universidad Politécnica de Madrid	Spain
University of Stuttgart	Germany
University of Hamburg	Germany
VU Amsterdam	Netherlands

Published S-Cube documents

All public S-Cube deliverables are available from the S-Cube Web Portal at the following URL:

<http://www.s-cube-network.eu/results/deliverables/>

The S-CUBE Deliverable Series

Vision and Objectives of S-Cube

The Software Services and Systems Network (S-Cube) will establish a unified, multidisciplinary, vibrant research community which will enable Europe to lead the software-services revolution, helping shape the software-service based Internet which is the backbone of our future interactive society.

By integrating diverse research communities, S-Cube intends to achieve world-wide scientific excellence in a field that is critical for European competitiveness. S-Cube will accomplish its aims by meeting the following objectives:

- Re-aligning, re-shaping and integrating research agendas of key European players from diverse research areas and by synthesizing and integrating diversified knowledge, thereby establishing a long-lasting foundation for steering research and for achieving innovation at the highest level.
- Inaugurating a Europe-wide common program of education and training for researchers and industry thereby creating a common culture that will have a profound impact on the future of the field.
- Establishing a pro-active mobility plan to enable cross-fertilisation and thereby fostering the integration of research communities and the establishment of a common software services research culture.
- Establishing trust relationships with industry via European Technology Platforms (specifically NESSI) to achieve a catalytic effect in shaping European research, strengthening industrial competitiveness and addressing main societal challenges.
- Defining a broader research vision and perspective that will shape the software-service based Internet of the future and will accelerate economic growth and improve the living conditions of European citizens.

S-Cube will produce an integrated research community of international reputation and acclaim that will help define the future shape of the field of software services which is of critical for European competitiveness. S-Cube will provide service engineering methodologies which facilitate the development, deployment and adjustment of sophisticated hybrid service-based systems that cannot be addressed with today's limited software engineering approaches. S-Cube will further introduce an advanced training program for researchers and practitioners. Finally, S-Cube intends to bring strategic added value to European industry by using industry best-practice models and by implementing research results into pilot business cases and prototype systems.

S-Cube materials are available from URL: <http://www.s-cube-network.eu/>

Table of Contents

1	Introduction	5
1.1	<i>Relation to WP Research Challenges</i>	<i>6</i>
2	SN4BPM Architecture	7
2.1	<i>Enhanced BPM Layering</i>	<i>7</i>
2.2	<i>Service Network Notation</i>	<i>8</i>
2.3	<i>Enhanced BPM Lifecycle</i>	<i>9</i>
2.4	<i>Transformations in the Enhanced BPM Lifecycle.....</i>	<i>11</i>
3	SN4BPM: Non-Functional View	12
3.1	<i>Enhanced BPM Layering: Non-Functional View</i>	<i>12</i>
3.2	<i>Enhanced BPM Lifecycle: Non-Functional View</i>	<i>13</i>
3.3	<i>Monitoring of KPIs in Service Networks.....</i>	<i>14</i>
4	Quantitative Analysis of Service Networks	16
5	Summary and Future Work.....	18
6	References	19

1 Introduction

In today's networked economy, companies are not independent, isolated entities, but they must act in a concerted manner to survive in an ever increasing dynamic environment. Thereby, interacting companies build networks to serve their joint customers in a dynamic manner, focusing on optimizing their financial benefits at the individual and network level. Recently, *Service Networks* (SNs) have been proposed to model such networks and analyze and optimize company's business collaborations. Service Network models reside on a high abstraction business level depicting partners as nodes and their offerings and revenues as edges. On the one hand, modeling a business landscape as an SN allows for calculating the *value* gained by a single partner when joining the network. On the other hand, an SN perspective gives the possibility to measure the value of the whole network. The value calculation is used for measuring the profitability of the SN, which can lead to adaptation of SNs, e.g. through outsourcing.

There is an abstraction gap between Service Networks and the underlying *business processes*. Service Networks focus on co-operations between partners in terms of offerings and revenues; they do not detail the concrete interactions occurring between the partners. Moreover, the dependencies between participants in an SN do not necessarily express the temporal dependencies relating the partners' interactions. Each offering and revenue in the SN is realized by a set of complex interactions between the partners. On the level of Business Process Management (BPM), these partner interactions, as well as the internal process steps, are modeled in detail as part of the process choreographies and their executable implementations.

In this deliverable we address the currently existing gap between Service Networks and business processes which are implemented based on a service-oriented architecture. Strategic decisions in SNs (such as how to restructure the network; whether to leave a particular network to join another; or whether it is advantageous to join multiple networks at the same time) are made by the partners in order to increase their own value. The value calculations in an SN, the foundation for strategic decisions at business level, are based on a set of *Key Performance Indicators* (KPIs) and *Service Level Agreements* (SLAs) that measure the performance of the underlying business processes. In order to calculate value on SN level, the metrics underpinning the KPI and SLA definitions have to be obtained from the business process level by monitoring the execution of the business processes.

Change is perhaps the only constant in SNs. The restructuring of an SN may be required to respond to competing networks, or to embrace and enact innovation in processes and technologies. Changes in the structure of the SN can also have an impact on the structure of network partners' business processes. Therefore, partners have to understand the connections and dependencies between SNs and the underlying business processes in order to adapt their business processes upon changes on SN level and vice-versa.

Because of the deep relations between the SNs and the underlying business processes, there is a need for a comprehensive architecture and methodology for developing, monitoring, and optimizing SOA-enabled business processes in SNs. In this deliverable we present the novel *SN4BPM* architecture that links Service Networks and BPM by the means of an enhanced BPM layering and lifecycle. The currently accepted BPM layers (i.e. process models, service compositions, services) serve as a basis for the enhanced BPM layering. The new Service Networks layer deals with models based on the *Service Network Notation* (SNN) that we have developed to represent participants and their interactions in SNs.

The advantages brought by the introduction of Service Networks as an additional layer on top of the BPM stack are two-fold: firstly, it simplifies the modeling of business processes that achieve strategic business goals, hence reducing the gap between the business experts- and the IT view on business processes. Secondly, analysts focusing on strategic goals of a business benefit from the detailed description and functionality of the business processes without being directly involved with BPM, thus lowering the bar in terms of technical expertise on modeling notations for business processes.

In this deliverable we present the SN4BPM architecture from both the functional and non-functional perspectives. The functional view includes a model-driven approach for transforming SN models to business process model skeletons; in addition we provide means to extract the topology of a Service Network from existing business processes. The non-functional perspective describes the types of non-functional properties relevant to each layer, such as KPIs, process metrics, QoS, their correlations among each other and with the functional artifacts. We also present initial ideas on monitoring of KPIs across partners in an SN. Finally, we present an analytical analysis method for value maximization in Service Networks which is enabled by the previously presented mappings.

This deliverable is based on the material of three published papers [1, 2, 3]. The body of the deliverable concisely summarizes the results of the papers in a self-contained manner and references the papers containing more details, where required. The papers in particular explain the motivation and background of our work in more detail and contain an example scenario. The deliverable is organized as follows. Section 2 describes SN4BPM from the functional perspective, introducing the enhanced functional BPM layering and lifecycle. Section 3 focuses on the non-functional aspects in SN4BPM, explaining their relations and how they are monitored. In Section 4, we present a quantitative analysis method for SNs. Finally, Section 5 presents our future research objectives in this area.

1.1 Relation to WP Research Challenges

In the following, we will give a short summary of the research challenges in WP-JRA-2.1 and explain how this deliverable is related to them.

The vision of work package JRA-2.1 is twofold:

1. *Developing concepts, mechanisms and techniques for analysis, rationalization and modeling (design) of end-to-end processes in SNs.* Analysis includes not only the design-time elicitation of functional requirements and performance metrics for end-to-end processes BPM, but also involves mining execution trails of choreographies to recover information about the run-time behavior of processes and transactions.
 - a. Analyzing, modeling and simulating end-to-end business processes in SNs. In particular this challenge concerns demand-driven creation and evolution of SNs;
 - b. Analysis and formal verification of business protocols involving bi-lateral and multi-lateral agreements between network nodes;
 - c. Requirements analysis and development of business-aware transaction concepts and mechanisms to support business protocols in SNs.
2. *Developing monitoring, measurement and adaptation concepts, mechanisms and techniques for evolving processes and protocols within SNs.* The second research objective addresses run-time behavior of business processes, and is particularly oriented towards developing and validating concepts, mechanisms and techniques for monitoring the execution of choreographies, measuring progress and performance of these processes against performance metrics, and, pro-actively adapting them before process anomalies or errors occur.
 - a. Mechanisms and concepts for monitoring and measuring events raised by business-aware transactions and related protocols and processes;
 - b. Mechanisms and concepts for adapting business-aware transactions and related protocols and processes in SNs.

In this deliverable CD-JRA-2.1.2 we deal with challenge 1a by introducing the SN4BPM architecture which connects SNs with business processes and thus enables analysis, modeling and simulation of business processes in SNs. We also cope with challenge 2a by introducing new concepts on monitoring of KPIs across partners in SNs.

2 SN4BPM Architecture

This Section provides an overview of the *Service Networks for Business Process Management* (SN4BPM) Architecture. More detailed material can be found in [1, 2]. The remainder is structured as follows: Section 2.1 introduces the Enhanced BPM Layering, which provides an outline of how the different technologies involved in the SN4BPM architecture are related to each other. Section 2.2 presents the SNN, while Section 2.3 introduces the Enhanced BPM Lifecycle that glues together SNN and the different elements of the SN4BPM architecture. Finally, Section 2.4 investigates more in depth the connections between the SNN and the process models, which are realized by the means of model transformation techniques.

2.1 Enhanced BPM Layering

The Enhanced BPM Layering, presented in Figure 1, relates the different technologies comprised in the SN4BPM Architecture. Its layers are designed to foster separation of concerns among the business level decisions, the modeling and management of the abstract business processes that realize the business decision, the executable business processes that implement the abstract business processes, and the underpinning IT infrastructure.

More in detail, the Enhanced BPM Layering is made of four layers:

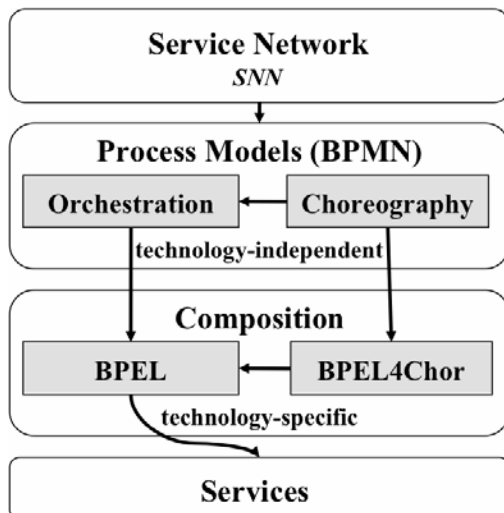


Figure 1: The Enhanced BPM Layering for the SN4BPM Architecture.

- The **Service Network** layer deals with the modeling, analysis and optimization of SN models expressing business interactions among the participants in the SNs. The focus is on supporting the decision-making and the definition of strategies and partnerships at the business level. The technologies at this layer allow for quantitative economic analysis of SNs to ascertain the optimal constellation of collaborative economic agents resulting in maximum economic value [1]. The models of SNs are expressed through the Service Network Notation introduced in Section 2.2, which is designed for an audience of non-IT specialists, and thus it abstracts from the nuances of, for instance, the message-based interactions among the participants in the SNs.

- The **Process Models** layer deals with the modeling of abstract business processes using

widely adopted and industry-supported standards such as the *Business Process Modeling Notation* (BPMN) [4] and *Abstract WS-BPEL* [5]. Abstract process models can be depicted as either *choreographies*, which describe the interaction protocol among multiple partners' services from a global perspective, and *orchestrations*, which formalizes point of view of one participant on the overall choreography and may also detail the participant's internal logic (which is not represented in the choreography). The abstract business processes are derived from Service Network Notation models defined at the Service Network layer using, for instance, model transformation approaches akin to the ones here presented in Section 2.4.

- The **Composition** layer encompasses the realization of executable business process models that, similarly to the abstract business processes modeled in the Process Models layer, can be either choreographies or orchestrations. Executable business processes can be obtained from the abstract ones through *refinement* (e.g. filling the missing details, see the *IT Refinement*

phase in Section 2.3). The technologies employed at this level are, for instance, *Executable WS-BPEL* [5] and *BPEL4Chor* [6].

- The **Services** layer comprises the actual services available in the SN, and the technologies to realize, manage and connect them, such as Web service frameworks like *Apache Axis* [7] and *Enterprise Service Bus* (ESB) implementations. Once deployed on the proper *Business Process Engine*, the executable business processes defined at the Composition layer realize services at the Service layer.

2.2 Service Network Notation

The Service Network Notation (SNN) provides the means to model business interactions among participants in SNs with a high-level of abstraction. The notation has as intended users the business analysts that focus on the quantitative economic analysis of SNs to support the decision-making and the definition of strategies and partnerships.

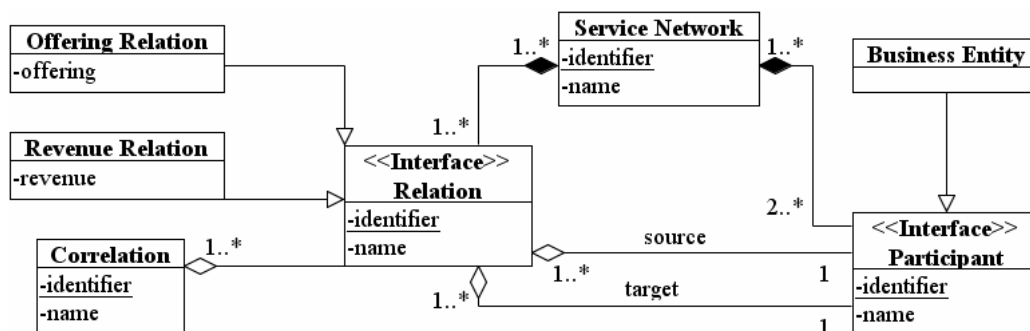


Figure 2: The SNN meta-model as a UML2 Class Diagram. [1]

Figure 2 presents the meta-model of the SNN notation using a UML2 Class Diagram. A *Service Network* is made of participants connected by relations, respectively represented by instances of the interfaces *Participant* and *Relation*. The interface *Participant* is implemented by the class *Business Entity*, which represents providers and consumers of functionalities that generate value in an SN. There are two kinds of relations, namely *offering* and *revenue*. Both kinds of relations connect a *source* and a *target* participant. Offering relations (modeled by the class *Offering Relation*) specify which services, specified by the field *offering*, does the source participant offer to the target. The *offering* field of an instance of *Offering Relation* specifies its actual content that could be, for instance, goods or services. Revenue relations (modeled by the class *Revenue Relation*) model the gain that the source participant has from the target in exchange for some provided offerings. The nature of the revenue is reported by the field *revenue*, and it usually sums of money. Offering and revenue relations can be grouped in *Correlations*, which identify the boundaries of cohesive business processes (e.g. choreographies) over which the interactions take place among participants that realize the offering and revenues. Correlations are important because they provide a way of co-relating offering and revenue relations (by grouping them in the same business process) with each other, and are instrumental in deriving abstract business processes from SNN models (see Section 2.4 for more details).

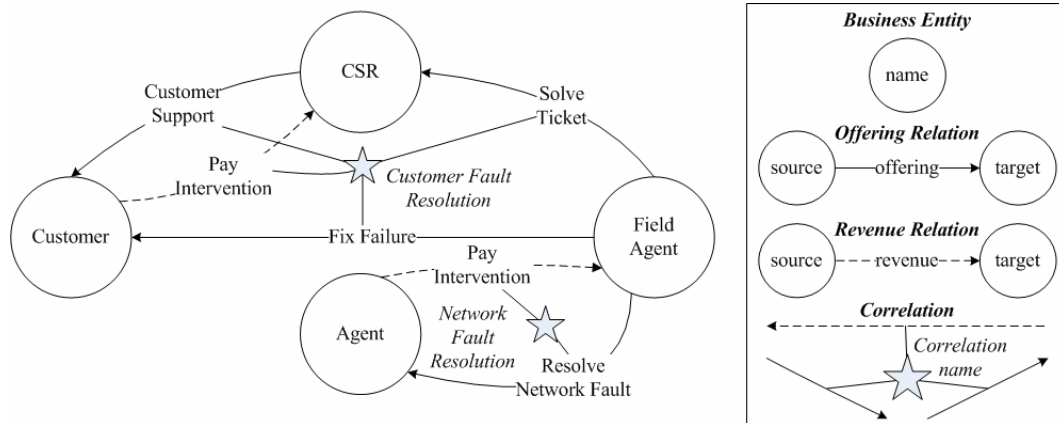


Figure 3: An example of SNN model based on the eTOM framework. [2]

Figure 3 presents an example of SNN model taken from one of the use cases presented in the eTOM framework [8]. It depicts an SN comprising four participants: the *Customer*, the *Customer Service Representative* (CSR), the *Field Agent* and the *Agent*. The SN comprises two essential business processes in fault resolution, namely *Customer Fault Resolution* and *Network Fault Resolution*. The Customer Fault Resolution process conceptualizes the Customer's procedure for reporting a fault to a CSR: after the reception of a trouble ticket, the CSR delegates the resolution job to a Field Agent, after which the Field Agent intervenes at the Customer's site to solve the issue. When the issue is solved, the Customer pays the CSR for the intervention (in our example we do not cover how the CSR pays the Field Agent). The Network Fault Resolution business process depicts from a high-level point of view of how faults are detected by an Agent, who delegates the resolution to the Field Agent in exchange for a payment.

2.3 Enhanced BPM Lifecycle

The Enhanced BPM Layering presented in Section 2.1 outlines how the different technologies involved in SNs relate to each other. However, it does not cover the "operational" dimension of the SN4BPM architecture, namely "what is done and when"; the Enhanced BPM Lifecycle here introduced covers this missing part. It builds on the established BPM Lifecycle (e.g., [9]) that relates the phases of the lifecycle with the artifacts that are manipulated and produced in them.

More to the detail, the established BPM Lifecycle we consider is made of the following phases (the artifacts manipulated by the different phases are underlined):

- **Analysis:** it deals with the definition (also known as *elicitation*) of the functional and non-functional requirements for the business processes and services that populate the SNs. The requirements can be either business- or technical in nature, such as which participants establish partnerships with, or which service provider to choose for the rendering of a particular service, and what SLA to establish.

that are extracted from the abstract business process models resulting from the Modeling phase.

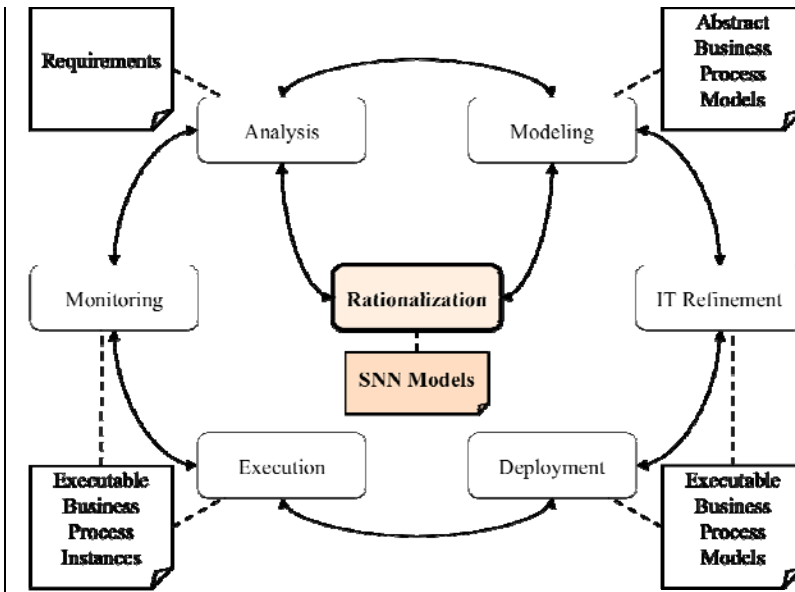


Figure 5: The Enhanced BPM Lifecycle (in highlight the additions with respect to the established BPM Lifecycle of Figure 4).

On the one hand, the bi-directional connection between the Analysis and Rationalization phases symbolizes the modeling of SNN models starting from the requirements; on the other end, new requirements (or changes to existing ones) may result from the analysis and optimizations performed on the SNN models, such as the decision to replace a partnership with another, or the dismissal of a service offering that does not pay off. The bi-directional correlation between the Rationalization and Modeling phases regards of obtaining abstract business process models from SNN ones and vice-versa, and it is covered

in Section 2.4.

The bi-directional correlations between the Rationalization, Analysis and Modeling phases allow for different ways of executing the Enhanced BPM Lifecycle, called *sequences*. Three different sequences are analysed in [2], focusing on achieving different goals, e.g. refining the requirements on the basis of the analysis performed in the Rationalization phase, or producing abstract process models that realize requirements that have been pre-optimized using the SNN analysis techniques.

2.4 Transformations in the Enhanced BPM Lifecycle

The Rationalization and the Modeling phases in the Enhanced BPM Lifecycle presented in Section 2.3 respectively deal with the modeling of SNN and abstract business process models. One of the added values of the SN4BPM architecture is the capability of semi-automatically producing skeletons of abstract business process models from business-like requirements expressed in the shape of SNN models. This is accomplished through two model transformations, called *Bottom-Up* and *Top-Down*, which respectively produce SNN models from the abstract business process ones and vice-versa.

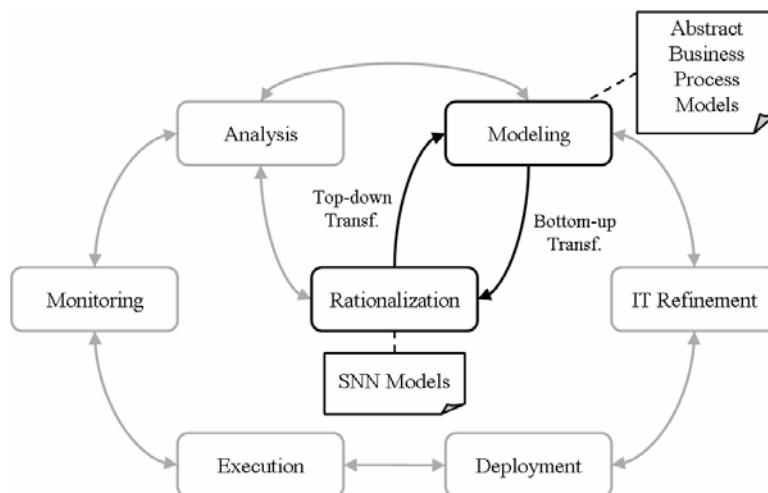


Figure 6. The role of Top-down and Bottom-up transformations in the Enhanced BPM Lifecycle.

As shown in Figure 6, the Top-down and Bottom-up transformations are the “glue” that binds the Rationalization and the Modeling phases in the Enhanced BPM Lifecycle. An implementation of the Bottom-up and Top-down transformations is presented in [2], targeting BPMN 1.0 as the language for describing the abstract business process models.

3 SN4BPM: Non-Functional View

In the last Section we introduced the SN4BPM architecture focusing thereby on the layering and the lifecycle of *functional* artifacts, i.e. SNs, process models, service compositions, and services. In this Section we will explain how *non-functional* artifacts, such as KPIs and SLAs, which are needed for quantitative analysis of SNs, fit into this picture.

We first give an overview of how non-functional aspects fit into the enhanced BPM Layering in Section 3.1. Then, we explain in Section 3.2 how they are addressed in the phases of the enhanced BPM Lifecycle. In Section 3.3, we focus on the monitoring of KPIs across participants in SNs.

3.1 Enhanced BPM Layering: Non-Functional View

Section 2 covers the layering and the lifecycle in SN4BPM focusing on functional artifacts such as SNs, abstract and executable process models, and services. The remainder will explain how non-functional properties (NfPs) fit into the picture of SN4BPM.

NfPs are defined based on *metrics*, which range from simple metrics obtained by measurement (e.g. service delivery time) or have predefined values (e.g., product price), to composite metrics that are defined using functions over other metrics (e.g. average service delivery time in a certain time period). Composite metrics are thus recursively composed using functions that are typically based on arithmetic and aggregation (avg, max, min, count) operators.

Metrics can be used as a basis for the specification of indicators, namely *Key Performance Indicators* (KPIs) and *Service Level Objectives* (SLOs) as part of SLAs. An indicator is defined on a metric and specifies a target value to be achieved in an analysis period, and allowed thresholds. Indicators are typically used in performance measurement, in particular in relation with business dashboards [12]. In that case, an indicator could use the “traffic light function” to specify which metric value ranges lead to “red”, “yellow”, or “green” results. Key performance indicators (KPIs) are based on metrics chosen to assess the achievement of business goals. A metric can also be used in guarantees (i.e. SLOs) as part of an SLA. A guarantee is typically a predicate over a metric (e.g. $\max(\text{response time}) < 20$ seconds) that specifies constraints on its values.

Different types of metrics are relevant to the different layers in SN4BPM (see also Section 2.1). Metrics on layers above may be calculated on the basis of metrics from layers below:

- **Service Network Layer:** *business metrics* are at this layer used for calculating the value of the SN (detailed in Section 4). Business metrics can be classified in *financial metrics* (e.g. revenues), *customer-related metrics* (e.g. customer satisfaction index), *process metrics* (e.g. order fulfillment cycle time), and “*learning & growth*” metrics (e.g. innovation rate), as used in the Balanced Scorecard [13]. Some of these metrics, and in particular process metrics, are obtained from the Process Model and Composition layers below by the means of measurement and monitoring. Naturally, business metrics can be defined recursively, e.g. customer satisfaction can be defined using the customer satisfaction index, the number of customer complaints, deadline adherence, and the average service delivery time. When business metrics are assigned target values and they are used for assessing the achievement of business goals, they become KPIs.
- **Process Model and Composition Layers:** these two layers deal with *process metrics*, which measure process cost, process quality and process duration. Process metrics can be based on several process models, e.g. when calculating duration across processes in a choreography.

Process metrics in BPM are evaluated by monitoring the business processes (see Section 3.3, and are needed on SN level for calculating higher-level business metrics).

- **Service Layer:** The bottom layer contains services implemented by the process models of the choreography and the service infrastructure on which the services are deployed. This layer deals with *QoS metrics*, e.g. response time and availability, which are typically evaluated from the perspective of the service consumer, and are used in the specification of SLOs in SLAs between participants. QoS metrics are technical, insofar they measure properties of service endpoints and infrastructure, but are also process metrics corresponding to QoS properties of the service implemented by that process, e.g. order processing cycle time (process metric) corresponds to order delivery time (QoS metric).

3.2 *Enhanced BPM Lifecycle: Non-Functional View*

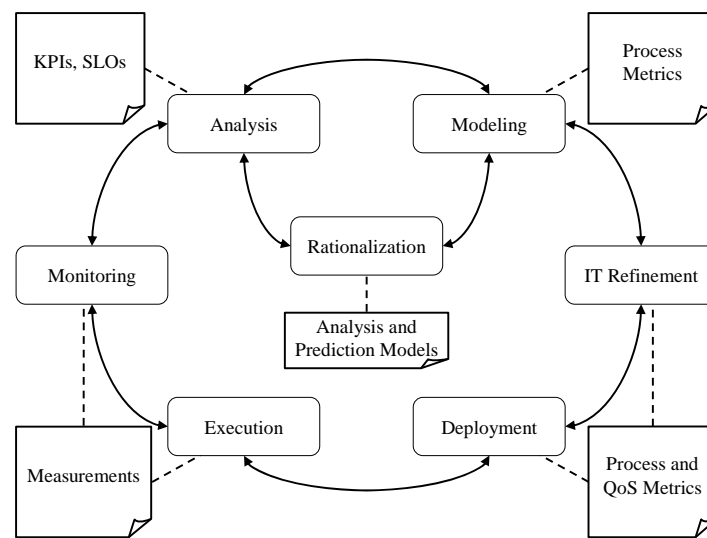


Figure 7: The Enhanced BPM Lifecycle in relation with the metrics used in the different phases.

Figure 7 shows how non-functional aspects fit into the enhanced BPM lifecycle:

- **Analysis:** this phase deals with the definition of the functional and non-functional requirements for the business processes and services that populate the SNs. The non-functional requirements are specified as KPIs and SLOs as part of SLAs. Both KPIs and SLOs are based on metrics that contain target values, and guarantees to be achieved; these in turn pose requirements on the further phases of the lifecycle dealing with the design and monitoring of business processes.
- **Rationalization:** in this phase SNs are modeled and analyzed. Quantitative analysis techniques are used for calculating the value (defined as a KPI) of each participant and of the SN as a whole (see Section 4). The value calculation and optimization is based on metrics whose values are either obtained from monitoring of business processes (in a later phase of the lifecycle) or they are statically defined (e.g. number of workers, product prices), or they are estimated in case there are no monitoring results available. The results of the quantitative analysis can be used for the optimization of the SN, which can lead to re-modeling of abstract business processes in the modeling phase, or even runtime adaptation of business processes if such mechanisms are in place.
- **Modeling:** in this phase, based on dealing with abstract business process models, the process metrics which KPIs and SLOs are based on are modeled for the business process, specifying its calculation based on process probes. That means, one specifies which information has to be measured at process runtime in order to be able to calculate the process metrics (that information includes state changes of process activities including timestamps, data from

process variables etc.). At this stage, simulation techniques can be used for checking whether KPI and SLO targets can be achieved with existing resources or selected services based on the process model. Simulation results can lead to optimization of the process model and/or selection of alternative services or re-planning of resources.

- **IT Refinement:** in this phase the abstract processes are refined into executable ones. This phase shares obvious similarities with the Modeling phase. However, at this stage is available more technical information, such as the concrete services, leading to the refinement of metric definitions into a monitoring solution. In this phase, instrumentation of systems for providing events might be needed. Moreover simulation and QoS aggregation techniques may be employed to check at design-time whether KPI and SLO targets can be achieved.
- **Deployment and Execution:** The processes and the monitoring solutions are deployed to the corresponding IT infrastructure.
- **Monitoring:** The metrics used in KPIs and SLOs are monitored at process run-time. Their values are typically displayed in dashboards, but they could also be provided to the rationalization step that calculates the value of the SN and, based on the results, adapts the process at run-time.

3.3 Monitoring of KPIs in Service Networks

The value calculations in an SN are based on a set of Key Performance Indicators (KPIs) which are measured based on process metrics of the choreographies of the SN. Traditionally, companies have monitored the performance of their internal processes using established technologies such as Business Activity Monitoring (BAM) [14]. In the setting of an SN, this is no more sufficient. Organizations participating in the SN now have and want to share SN relevant information of their internal processes among each other in order to be able to analyze their own performance in the SN as well as the overall performance of the SN. Therefore, partner organizations in the SN have to exchange monitoring information between them.

In [3], we propose an approach of how to model and monitor KPIs across participants in a service network. We assume that the SN is mapped to service choreography descriptions, as described in Section 2.3.

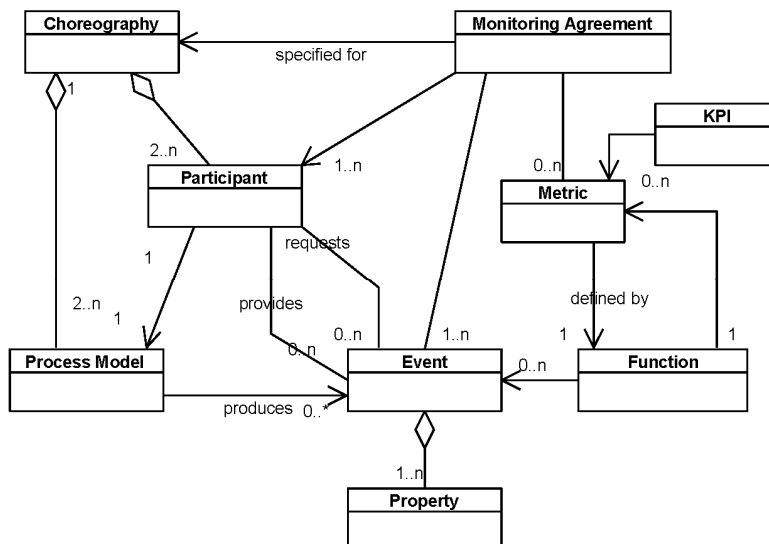


Figure 8: Main concepts of the Monitoring Agreement.

For monitoring of the choreographies in the SN we take an event-based approach whereby the participants create a *monitoring agreement* that specifies which events each participant has to provide and how these events are aggregated to calculate KPIs. Figure 8 shows the main concepts and their interrelation needed for the specification of a monitoring agreement. The *choreography* description consists of *participants* and *abstract process models* implemented by those participants. Based on the process models of the

choreography, each participant provides and requests *events*. An event definition references a process element (i.e., process, activity, variable) and specifies at which state of that element the event is emitted (e.g., OrderReceivedEvent is published when Receive Order activity has completed). In addition, events contain *properties* needed for calculation of metrics (e.g., timestamp for duration

specific metrics, or other domain specific process data such as number of ordered products, or customer type) and correlation with other events (e.g. by using an order identifier). Finally, one has to specify how events can be obtained at process runtime, e.g. by subscribing to a publish/subscribe topic. *Metrics* which serve as basis for *KPI* definitions are calculated based on *functions* over events and other metrics.

The monitoring agreement is specified in the composition layer for a service choreography description. A choreography description (using an interconnected interface choreography model) specifies the public processes of the participants in the SN and the message interactions they agree on. In the same manner, in the monitoring agreement the participants agree on produced and consumed events.

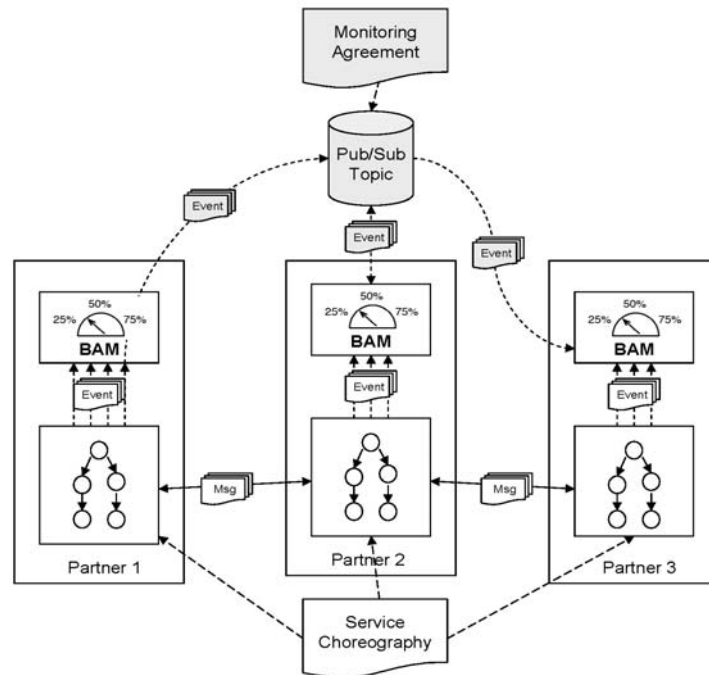


Figure 9: Monitoring Architecture. [3]

After creation of a monitoring agreement, each participant in the choreography implements its abstract process model of the choreography, i.e. refining it to an executable process. At the same time, each participant instruments its process implementation to provide monitoring events to other participants. The exchange of events can be done using a shared publish/subscribe topic where participants publish events to and subscribe for events from the others (Figure 9). Each participant can then monitor the KPIs of the SN using its own BAM solution.

4 Quantitative Analysis of Service Networks

In this Section we describe a quantitative analysis method for SNs which supports calculation and maximization of the value of a participant in the SN. This method is applied in the rationalization phase of the SN4BPM lifecycle (Sections 2.3 and 3.2).

To evaluate and measure their performance within a Service Network and to define business objectives as part of their strategic behavior, organizations identify KPIs based on key financial metrics, process metrics, and QoS metrics used in SLAs (see Section 3). For example, the *value* that a participant derives from the SN is one such KPI which is again based on several other KPIs such as the satisfaction of this participant's customers and revenues. Satisfaction, in turn, depends on many other metrics such as the service delivery time, which usually should not exceed an upper bound specified in the relevant SLA.

The participants of a Service Network need to monitor on a periodic basis their KPIs and take corrective action if needed. The participants' job could be made significantly easier if they could use models that predict what the effect on a specific KPI, of a corrective action will be, and even better, what would be the optimal change (if it can be found) of parameter values (e.g., product prices or guarantees in SLAs) and processes to yield the best possible change of a specific KPI. In the following we describe an analysis technique which enables *maximization of the value* of a participant in the SN by *adjusting dependent metrics* (e.g., product prices) [1].

In our model, the KPIs are perceived as functions of all parameters that may affect value. Let $\bar{x}_i = (x_{i1}, \dots, x_{iK}), i = 1, \dots, n$ be the input vector (e.g. services, resources, prices) of a partner b_i that is used by the various functions expressing the KPIs of interest. In the telecommunications example introduced in [1], the vector \bar{x}_i for the service provider (SP) could be prices he imposes for the services he offers and the labor rates he pays to his employees. Consider now the function $f_i(\bar{x}_i)$ that denotes a KPI for b_i due to its participation in the network. For example, this function could represent a revenues KPI, resulting from the sum of revenues of b_i , from all its network partners, to whom b_i sells his services.

Predictions of improvement and optimization of a KPI in our models should also take into account the *constraints* that exist. There are two forms of constraints: *intrinsic* to the partner, such as maximum capacity of resources (number of people employed, maximum storage and CPU power available, etc.), and those imposed to the partner through the SLAs, e.g. the maximum price tolerated by a partner's services buyer and the maximum delay tolerated for installing a new service.

In the general case, the maximization problem is defined as follows:

$$\max f_i(\bar{x}_i) \text{ s.t. } \bar{x}_i < \bar{C} \quad (1)$$

where $\bar{C} = (C_1, \dots, C_K)$ is the vector of constraints.

In an SN, each business entity captures value that is given by the sum of the revenues obtained from interacting with other business entities in a time interval, plus the expected value in the next time interval. The expected value of a business entity represents the effect that all its relations have upon it and depends on the expected revenues of the next time period and on the expected degree of satisfaction that the participant's buyers have for his services.

The value of a business entity is estimated as the sum of several metrics. Some of these metrics are relevant to our SNs such as the profits of one of its business units over a certain period (i.e. revenues minus costs) and the expectation of revenues over the next time period; others are not related, e.g. savings and capital equipment. Estimating revenues is harder when a business unit is operating alone in the marketplace (i.e. its customer list is unpredictable and volatile), as opposed to when it is operating within a network where buyers and sellers are fixed (at least for some time) and customers

tend to have long-term relationships with their service providers. In such a network it is also feasible to get customers evaluations about the quality of their providers' services and integrate them into a *satisfaction index*. The satisfaction index Sat in our example is a function of the service delivery time, the price p paid by the consumer for the service, the throughput requests/hour n_1 performed by agents, the number n_2 of customers that withdrew in the last period and the number n_3 of customers that complained in the last period. Although we assume simple dependencies between the satisfaction index and the other metrics, in a real-case scenario empirical market studies can establish more accurate relationships.

We next apply the above ideas to our example and formulate a simple price optimization problem. We assume that calculations take place within a fixed time interval in which the network remains stable in number of participants. The value V_{sp} of the service provider at the end of time interval $[T_{N-1}, T_N]$ as given in [15] is:

$$V_{sp}(T_N) = R_{sp}(T_N) - P_{sp}(T_N) + v_{sp}(T_N) \quad (2)$$

where $R_{sp}(T_N) = \sum_{i=1}^n p_i$ are the revenues by setting price p_i for service type i , $P_{sp}(T_N) = \sum_{i=1}^m r_i$ are the payments by setting labor rate r_i for type of employee i and $v_i(T_N, Sat)$ is the expected value due to all the relations partner b_i has in $[T_N, T_{N+1}]$. One way to estimate the expected value is to include the satisfaction index, the intuition being that a declining satisfaction index should lower revenue expectations and therefore the value of a relationship, whereas an increasing satisfaction index would raise revenue expectations and therefore the relationship value. We assume that each partner acting as a customer to another one, knows its own satisfaction index. We also assume that through market research, questionnaires to their customers and so on, the suppliers have also knowledge of their customers' satisfaction indices. Let $Sat_{ij}(\tau)$, be the satisfaction of partner b_j being a customer of partner b_i at time τ . One way to estimate $Sat_{ij}(\tau)$, is by using its weighted averages:

$$\overline{Sat}_{ij}(T_N) = \gamma_i \overline{Sat}_{ij}(T_N) + \delta_i \overline{Sat}_{ij}(T_{N-1}) \quad (3)$$

where $0 \leq \gamma_i, \delta_i \leq 1$ and $\gamma_i + \delta_i = 1$. The estimation of the expected value of the relationship between partners b_i and b_j in $[T_N, T_{N+1}]$ is thus:

$$v_{ij}(T_N) = \overline{R}_{ij}(T_N) + \frac{\overline{Sat}_{ij}(T_N) - \overline{Sat}_{ij}(T_{N-1})}{\overline{Sat}_{ij}(T_{N-1})} \overline{R}_{ij}(T_N) = \frac{\overline{Sat}_{ij}(T_N)}{\overline{Sat}_{ij}(T_{N-1})} \overline{R}_{ij}(T_N). \quad (4)$$

The expected value of all the relationships that a partner has as "downstream" in the service system, i.e. with all those partners who are the receivers of its offerings, is given by:

$$v_i(T_N) = \sum_{t_{ij} \in X} v_{ij}(T_N). \quad (5)$$

The above parameters are needed to calculate value according to Equation 2. We assume that an upper bound on p and a labor rate r are given. Response time t is specified in SLAs as upper bound and is monitored and calculated in the BPM layering stack (see Section 3.3). n and n_1 are monitored and calculated in the BPM layering stack and are used in order to calculate t . n_2 and n_3 are calculated by the BPM layering stack and are given together with t and n in the SN level in order to calculate the satisfaction and the value of the partners according to the equation 2. In order to determine a price p such that the value of the service provider is maximized we solve the maximization problem given in equation 1 that is formed in the given example as follows:

$$\left. \begin{array}{l} \max V_{sp}(\vec{p}) \\ s.t. \vec{p} < \vec{p}_{SLA} \end{array} \right\} \Rightarrow \left. \begin{array}{l} \max \left(\sum_{i=1}^n p_i - \sum_{i=1}^m r_i + v_{sp}(T_N, Sat(t, p, n_1, n_2, n_3)) \right) \\ s.t. \vec{p} < \vec{p}_{SLA} \end{array} \right\} \quad (3)$$

where \vec{r} is a function of \vec{p} : $\vec{r} = g(\vec{p})$ and \vec{p}_{SLA} is the upper bound of the price vector given in the SLA between the customer and the service provider. We assume that time t is a parameter that is given to us by the analysis phase of the lifecycle described in Section 2.3. We then calculate the price vector that maximizes value according to that price vector.

5 Summary and Future Work

In this deliverable we have presented initial results on models and mechanisms for quantitative analysis of correlations between KPIs, and SLAs, defined in SNs and underlying business processes. We have employed service network models for quantitative analysis based on KPIs, which enables strategic decisions for participants such as determination of optimal prices. In order to perform the analysis on SN level and implement its results in BPM, SNs have to be connected to the BPM stack. We have therefore introduced the SN4BPM architecture describing an enhanced BPM layering and lifecycle.

The future work in this area includes refining and implementing the initial results and extending the approach. First, we intend to develop new quantitative analysis methods for Service Networks based on game theory involving domain specific KPI-models as well as simulation techniques for SNs. In particular, we want to consider not only optimization from the point of view of a single partner in the SN but also global optimization from the point of view of the whole SN.

In a similar way we plan to apply social network analysis techniques to SNs in specific domains. An example of currently ongoing investigation is in the analysis of the software architecting domain of on-line communities of practice, as described in [16]. Next to monitoring KPIs, social network analysis techniques can be used to also detect SN behaviors needing improvement, and provide feedback for bottom-up refinement of the SN models.

Considering SN4BPM, we will refine the existing modeling notation of service networks in order to achieve a higher automation of the transformation from SN models to abstract BPMN process models. The cross-organizational monitoring approach will be refined and integrated with the SN analysis, providing results directly into the rationalization phase enabling prediction and faster reaction to the violation of KPI targets. The analysis and optimization on SN level results in need for adaptations on process level. In particular, in case of outsourcing decisions on SN level, the corresponding underlying business processes have to be fragmented. Fragmentation mechanisms will be devised together with the work package JRA-2.2.

6 References

- [1] Bitsaki, M., Danylevych, O., Van den Heuvel, W.J., Koutras, G., Leymann, F., Mancipopi, M., Nikolaou, C., Papazoglou, M.: An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network. In: ServiceWave2008. (2008)
- [2] Bitsaki, M., Danylevych, O., Van den Heuvel, W.J., Koutras, G., Leymann, F., Mancipopi, M., Nikolaou, C., Papazoglou, M.: Model Transformations to Leverage Service Networks. In: Fourth International Workshop on Engineering Service- Oriented Applications: Analysis and Design (WESOA 2008), Proceedings. Lecture Notes in Computer Science, Springer-Verlag (2008)
- [3] Wetzstein, B., Danylevych, O., Leymann, F., Bitsaki, M., Nikolaou, C., van den Heuvel, W.-J., Papazoglou, M.: Towards Monitoring of KPIs Across Partners in Service Networks. In: Workshop on Monitoring, Adaptation and Beyond (MONA+). (2008)
- [4] Object Management Group (OMG). Business Process Modeling Notation Specification. OMG Standard (February 2006)
- [5] Organization for the Advancement of Structured Information Standards (OASIS). Web Services Business Process Execution Language Version 2.0. OASIS Standard (April 2007)
- [6] Decker, G., Kopp, O., Leymann, F., Weske, M.: BPEL4Chor: Extending BPEL for Modeling Choreographies. In: IEEE International Conference on Web Services (ICWS'07). IEEE Press, New York (2007) 296—303
- [7] Apache Axis. Website: <http://ws.apache.org/axis/>
- [8] TeleManagement Forum: Enhanced Telecom Operations Map The Business Process Framework For The Information and Communications Services Industry. <http://www.tmforum.org>
- [9] Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)
- [10] Apache ODE. Website: <http://ode.apache.org/>
- [11] S-Cube Deliverable. PO-JRA-1.2.1 State-of-the-Art Report on Principles, Techniques and Methodologies for Monitoring and Adaptation (2008)
- [12] Eckerson, W.: Performance Dashboards: Measuring, Monitoring, and Managing Your Business. Wiley (2005)
- [13] Kaplan, R.S., Norton, D.P.: The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press, USA (1996)
- [14] Jeng, J.J., Schiefer, J., Chang, H.: An Agent-based Architecture for Analyzing Business Processes of Real-Time Enterprises. In: Proceedings of the 7th International Conference on Enterprise Distributed Object Computing (EDOC '03), Washington, DC, USA, IEEE Computer Society (2003)
- [15] Caswell, N., Feldman, S., Nikolaou, C., Sairamesh, J., Bitsaki, M., Koutras, G.D., Iacovidis, G.: Estimating Value in Service Systems – A theory and an example. IBM Systems Journal 47(1) (2008)
- [16] Babar, M. A.; Dingsoyr, T.; Lago, P. & van Vliet, H. (eds.): Establishing and Managing Knowledge Sharing Networks. In: Software Architecture Knowledge Management: Theory and Practice, Springer (2009)

CD – JRA – 2.1.2

Paper [1]:

Bitsaki, M., Danylevych, O., Van den Heuvel, W.J., Koutras, G., Leymann, F., Mancioppi, M., Nikolaou, C., Papazoglou, M.:

An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network.

In: *ServiceWave2008*. (2008)

An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network^{*}

Marina Bitsaki¹, Ohla Danylevych², Willem-Jan van den Heuvel³, George Koutras¹, Frank Leymann², Michele Mancioppi³, Christos Nikolaou¹, and Mike Papazoglou³

¹ Computer Science Department, University of Crete, Greece
{bitsaki,koutras,nikolau}@tsl.gr

² Institute of Architecture of Application Systems, University of Stuttgart, Germany
{olha.danylevych, frank.leymann}@iaas.uni-stuttgart.de

³ Department of Information Systems and Management, Tilburg University, Netherlands
{W.J.A.M.vdnHeuvel,michele.mancioppi,mikep}@uvt.nl

Abstract. Networks of interdependent organizations cooperate to produce goods or, nowadays, services that are of value to their markets as well as to the participating organizations. Such co-operations can be supported by corresponding business processes which are based on SOA technology. Developing and managing SOA-based business processes in such service networks necessitates a *comprehensive* architecture which is on the one hand grounded on solid design principles, and on the other hand capturing best-practices and experiences. Such an architecture is currently lacking. This paper outlines a first attempt to develop and validate an architecture for developing, monitoring, measuring and optimizing SOA-enabled business processes in service networks. A case study from the telecommunications industry is analyzed, and different aspects of service networks are addressed.

Keywords: Service Value Network, Key Performance Indicator, Business Process Management, Business Activity Monitoring.

1 Introduction

The emerging service economy and the advances in information technology have dramatically increased the complexity of understanding how organizations evolve within a world of interactions and partnerships. Instead of large, vertically integrated organizations, we observe the emergence of globe-spanning networks of interdependent companies that cooperate to provide value to their markets based on services (so-called *service value networks*). Business processes technology is used to prescribe how organizations work internally and how they work together to achieve the value of the service network. But the overall management of the corresponding business processes is growing more complex because of the inter-organizational and intra-organizational nature of business processes supporting the complex web of interactions of service value networks.

^{*} The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

Several studies focus on creating and reconfiguring service value networks (see [1,3]). [1] proposes a methodology for analyzing the dynamics of value in networks at the operational, tactical, and strategic level with an emphasis on visualization and qualitative methods. In [2], the authors combine IT systems analysis with economic-based business modeling in order to build an e-business model that specifies e-business scenarios rather than on defining values. Besides the qualitative approaches, there is a growing need for quantitative methods. [3] presents a method for computing values by taking into consideration partners' satisfaction and additional value that is accrued by the relationship levels developed by the various partners.

In this paper we will focus our attention on *Service Networks* (SNs) (see [4,5]): it offers services that are obtained by composing other services provided inside the SN by a diversity of service providers by means of business processes.

From the operational view of the service network, one should focus on the management of the business processes and the monitoring of financial and operational measures of performance also called *Key Performance Indicators* (KPIs) in order to evaluate or improve them. Examples are overall process execution time, percentage of service requests fulfilling *Quality of Service* specifications, customer satisfaction index, etc. *Business Process Management* (BPM) together with Service Oriented Architecture (SOA) support organizations in the continuous improvement of their business's performance through the effective convergence of IT and business [6].

From the business view of the service network, there is a need to define the activities that achieve business goals such as cost cuts, market share increase, profit increase, customer satisfaction increase etc. Moreover, different partners may have different business goals, which may possibly be conflicting. For instance, one partner may be more interested in customer satisfaction, which may require an increase in costs to be achieved. This may be unacceptable for partners whose first priority is cost reduction. In [3] it is shown how the concept of value, properly defined, can be used as a unifying concept for studying service networks (called service value networks in that context) instead of the various heterogeneous business goals.

In this paper, we address the currently existing gap between business strategy and business models from one side and service system implementations on the other side. Strategic decisions (such as how to restructure the network; whether to leave a particular network to join another; or whether it is advantageous to join multiple networks at the same time; etc.) have to be made by the partners in order to increase their own value. Restructuring of a service network may be required to respond to competing networks or innovation in processes and technologies. Changes in the structure of the service network could drastically affect network partners' business objectives and/or network-wide business processes. Unfortunately, the current methods and tools for developing and managing service networks are highly fragmented, merely providing support for isolated parts of the huge task. This paper outlines a first attempt to develop and validate a comprehensive methodology for developing, monitoring, measuring and optimizing SOA-enabled business processes in SNs. We have developed the *Service Network Notation* (SNN) to represent participants in a SN and their interactions in terms of offerings and revenues. Such a comprehensive methodology is currently lacking. By adding SNs on top of the current BPM stack, analysts focusing on strategic goals of a business benefit from the detailed description and functionality of the business processes without being directly involved with BPM. This level of

abstraction that is achieved through the linkage of SN to BPM provides them a better understanding of how to accomplish their goals.

The remainder of the paper is organized as follows. Section 2 introduces SNs through an example borrowed from the telecommunications industry. Section 3 introduces a meta-model of the SN. Section 4 shows how to analyze SNs and describe their basic properties. Section 5 describes standard BPM approaches, while section 6 proposes a novel architecture, SN4BPM, linking SN and BPM. Finally, section 7 provides some concluding remarks and discusses directions for future work.

2 SN by Example

In this section we describe the structure of service networks through an example taken from the telecommunications industry. Considering the methodology developed in [3], we model the service network of the telecommunications companies as a flow graph which comprises nodes (economic entities) and transfer objects (offerings which could be goods, services, information).

Our example is based on the Enhanced Telecom Operations Map (eTOM) [7] which is a reference framework for categorizing all the business activities that a service provider may use. In particular, we will describe the service network that is formed in order to set up a new service. We consider the following entities that collaborate with each other: the service provider (SP) offers services (realized as bundles of services such as orders for digital subscriber line, wireless, Internet data centre services, etc) to the subscribers. The external partners of the SP include the suppliers who provide resources (equipment, infrastructure, etc) and content providers with whom the SP co-operates in order to produce the bundle of services offered to the subscriber (e.g. video on demand, music educational content etc.) The internal partners of the SP (who can be outsourced and become external partners as well) are the call centre who provides information to subscribers over the telephone, the sales agent who provides prices for the different services to the subscribers, the service agent who is responsible for the set up and configuration of a subscriber's order, the field agent who performs service installations at the subscriber's site, the account manager who creates updates and manages accounts once the order is fulfilled and the billing agent who is responsible for the management of the billing system.

In Fig. 1, we provide a representation of the service network showing the relations created among the various entities. The economic entities are represented by circles and offering flows are represented through arcs. There are two types of offerings: services (depicted by solid arrows) and revenues (depicted by dashed arrows). A possible scenario for this example could be the following: A new subscriber contacts the call centre and orders the digital subscriber line service. The call centre enters the subscriber's information (name, address, etc.) to a customer information system and asks the sales agent to determine which services can be provided to this specific subscriber. The sales agent provides a list of possible services to the call centre which in turn informs the subscriber. The subscriber selects the service he wants and makes the order. The call centre submits the order to an order management system of the service provider. The account manager creates a new account for the subscriber and the service

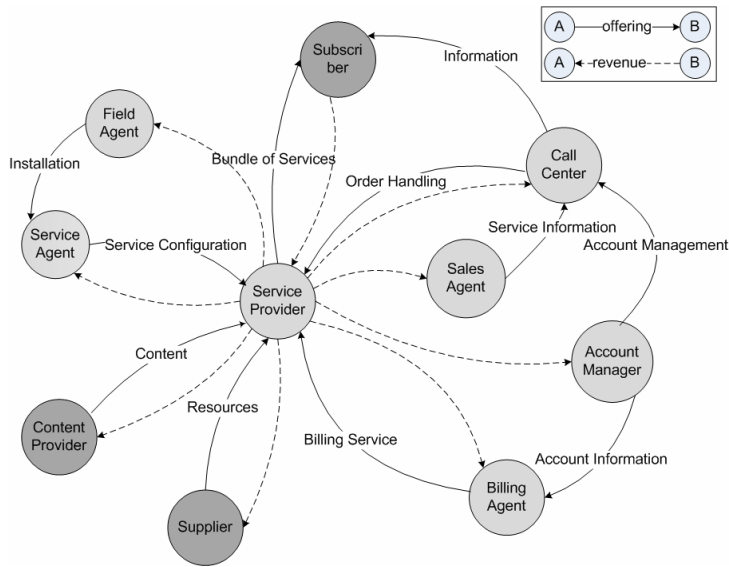


Fig. 1. The service network for a new service set up

agent configures the requested service and asks the field agent to install the equipment at the subscriber's site. As soon as the field agent completes his work, the service agent activates the new service.

The participants of the network, at the business level, are primarily interested in making sure that they derive value from their participation in the network. Participants in the network are also interested in promoting their own more general business objectives through their participation in the network, such as for example their market share, or their effectiveness in responding to market needs and being innovative, or their customer satisfaction. In section 4, we show how all these business objectives can be interconnected and also linked to IT level performance criteria such as SLAs, business processes, workflows performance etc.

3 SNN Meta-model

The meta-model for the SNN is shown in Fig. 2 as a UML2 Class Diagram. A Service Network consists of participants that are connected by relations. Participants and relations are represented by instances of the interfaces Participant and Relation. Instances of service networks, participants and relations have a name and are uniquely identified by an identifier. The interface Participant is implemented by the class Business Entity, representing providers and consumers of functionalities that generate value in a service network. SNN models comprise two kinds of relations: offering and revenue. Both kinds of relations connect a source and a target participant. Offering relations (modeled by the class Offering Relation) specify what services are offered (specified by the field offering) by the source participant (acting as service provider)

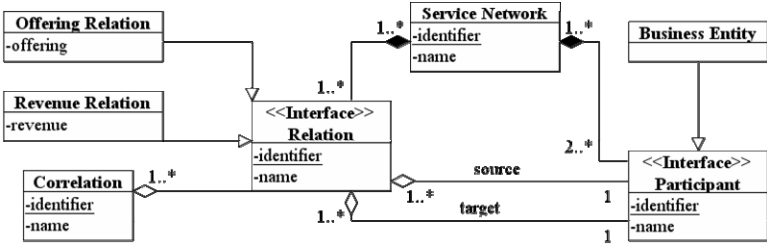


Fig. 2. A UML2 Class Diagram describing the SNN meta-model

to the target. Offerings could be goods or services, or a combination of both. Revenue relations (class Revenue Relation) describe the gain that the target participant has from the source in exchange for provided service. Revenues (modeled by the field revenue) are usually sums of money.

Generally, a SNN model describes interactions among a set of participants that take place over multiple, unrelated business processes. All the offering and revenue relations that take place over the same business process are *correlated*. Correlations allow to immediately visualize which parts of an SNN models pertain to a given business process, and which not.

4 Analysis of SNs

Organizations are expected to work worldwide fostering complex relations and developing complementary skills to generate and exchange goods, services or information. In order to evaluate and measure the performance of an organization within a service network and define business objectives as part of the firm’s strategic behavior, the organization identifies specific KPIs [8]. Apart from measurements that take place at the BPM lifecycle (described in Section 6), KPIs are connected to parameters given in SLAs and parameters given by the interacting participants. For example, the value that a participant derives from the network is a KPI and could be connected, among other factors, to the satisfaction of this participant’s customers. Satisfaction, in turn, depends on many factors such as the service delivery time, which usually should not exceed an upper bound specified in the relevant SLA.

To implement this service network, quite a few business processes must be deployed and operate such as: “order receipt”, “order handling”, “service configuration”, “service installation”, and “inquiries and complaint handling”. These processes are distributed between several business units and business partners. To efficiently implement all these processes, SLAs will have to be agreed between partners. For example a cost KPI and cost reduction target for the SP will be affected by SLA requirements that a new service installation has to handled within a very limited timeframe, since the SP will have to pay service technicians and engineers to be available and on call to cover all new services requests by customers. Value derived from the network for, say, a content provider is affected by costs incurred for having sufficient equipment available to handle any real-time requests for content. On the other hand, if SLAs are not satisfied, then penalties for non-SLA compliance may have to be applied and customers’ satisfaction may drop,

thereby reducing the value derived for the content provider from its participation in the network.

It can therefore be seen that if business processes are implemented in sloppy and inefficient ways, or system and/or human resources are not used judiciously and are either wasted or under-provided, then the whole service network may break down, simply because the individual partners will not be achieving their desired KPIs and/or they will not be deriving sufficient value from their participation in the network. We now present elements of our modelling effort that tries to link satisfaction of business objectives and KPIs with SLAs and business process performance yardsticks.

The partners of a service network need to monitor on a periodic basis their KPIs and take corrective action as need be. The partners' job could be made significantly easier if they could use models that predict what the effect on a specific KPI, of a corrective action will be, and even better, what would be the optimal change (if this can be found) of parameter values and processes to yield the best possible change of a specific KPI. We are working on such models, and in what follows, we show how these models could be applied to our telecom' example to improve a specific KPI.

In our models, the KPIs are perceived as functions of all parameters that may affect their value. The shape of these functions can be affected by the structure of the business processes (for example, if the telecom provider in our example innovates and eliminates the need of technicians to install a new service, then a technician labor rate will obviously cease to have an effect on the function expressing the dependence of a cost KPI to various cost parameters). Let $\vec{x}_i = (x_{i1}, \dots, x_{iK})$, $i = 1, \dots, n$ be the input vector (e.g. services, resources, prices etc.) of a node (economic entity) b_i that is used by the various functions expressing the KPIs of interest. For example, in the telecommunications example the vector \vec{x}_i for the SP could be prices he imposes for the services he offers and the labor rates he pays to his employees. Consider now the function $f_i(\vec{x}_i)$ that denotes a KPI for b_i due to its participation in the network. For example, this function could represent a revenues KPI, resulting from the sum of revenues of b_i , from all its network partners, to whom b_i sells his services.

On the other hand, any prediction of improvement or even optimization of a KPI in our models, should also take into account constraints that exist. There are two forms of constraints: those that are intrinsic to the partner, such as maximum capacity of resources (number of people employed, maximum storage and CPU power available, etc.) and those that are imposed to the partner through the SLAs, for example maximum price tolerated by a partner's services buyer, or maximum delay tolerated for installing a new service in our telecom example, etc.

In general therefore, we can define the following maximization problem:

$$\max f_i(\vec{x}_i) \text{ s.t. } \vec{x}_i < \vec{C} \quad (1)$$

where $\vec{C} = (C_1, \dots, C_K)$ is the vector of constraints.

Next, we apply this framework to the telecommunications example. We choose to focus on value created for each partner, since this KPI has also been studied by us for other examples as well, see [3]. Though there are multiple ways to express value in models, we choose a relatively simple one: each participant captures value which is given by the sum of profits from interacting with nodes in a time interval and the

expected value in the next time interval. The expected value of a participant represents the effect that all its relations have upon it and depends on the expected revenues of the next time period and on the expected degree of satisfaction that the participant's buyers have for his services.

How close is this representation of value to common practices in the marketplace? We claim that it is very close. The value of a business entity is usually estimated as the sum of several components, some of which are relevant to our service networks such as the profits of a business unit over a certain period (revenues minus costs) and the expectation of revenues over the next time period, and some of which are not related such as savings, capital equipment, etc. Notice also that estimating revenues is harder when a business unit is operating alone in the marketplace (its customer list being unpredictable and volatile) as opposed to when a business entity is operating within a network where buyers and sellers are fixed (at least for some period of time) and where customers tend to have long term relationships with their service providers. In such a network it is also feasible to get customers evaluations about the quality of their providers' services and integrate them into a "satisfaction index". Satisfaction index Sat in our example is a function of the service delivery time, the price p paid by the customer for the service, the requests/hour n_1 performed by agents, the number n_2 of customers that withdrew in the last period and the number n_3 of customers that complained in the last period. Although we give here simple examples of dependencies between the satisfaction index and the other parameters, empirical market studies can establish more accurate relationships.

Let us now apply the above ideas to our example and formulate a simple price optimization problem. We assume that calculations take place within a fixed time interval in which the network remains stable in number of participants. The value V_{sp} of the service provider at the end of time interval $[T_{N-1}, T_N]$ as given in [3] is:

$$V_{sp}(T_N) = R_{sp}(T_N) - P_{sp}(T_N) + v_{sp}(T_N) \quad (2)$$

where $R_{sp}(T_N) = \sum_{i=1}^n p_i$ are the revenues by setting price p_i for service type i ,

$P_{sp}(T_N) = \sum_{i=1}^m r_i$ are the payments by setting labor rate r_i for type of employee i and $v_i(T_N, Sat)$ is the expected value due to all the relations partner b_i has in $[T_N, T_{N+1}]$. (For a more detailed description see [3].)

In order to calculate value according to equation 2 we need to calculate the above parameters. An upper bound on price p and a labor rate r are given in Service Level Agreements (SLAs) between the service provider and the customer and the service provider and his employees respectively. Response time t is given in SLAs as upper bound and is calculated by the lower levels of the BPM layering stack. n and n_1 are calculated by the BPM layering stack and are used in order to calculate t . n_2 and n_3 are calculated by the BPM layering stack and are given together with t and n in the SN level in order to calculate the satisfaction and the value of the participants according to the equation 2.

In order to determine price p such that the value of the service provider is maximized we solve the maximization problem given in equation 1 that is formed in the given example as follows:

$$\left. \begin{array}{l} \max V_{sp}(\bar{p}) \\ \text{s.t. } \bar{p} < \bar{p}_{SLA} \end{array} \right\} \Rightarrow \left. \begin{array}{l} \max \left(\sum_{i=1}^n p_i - \sum_{i=1}^m r_i + v_{sp}(T_N, Sat(t, p, n_1, n_2, n_3)) \right) \\ \text{s.t. } \bar{p} < \bar{p}_{SLA} \end{array} \right\} \quad (3)$$

where \bar{r} is a function of \bar{p} : $\bar{r} = g(\bar{p})$ and \bar{p}_{SLA} is the upper bound of the price vector given in the SLA between the customer and the service provider. We assume that time t is a parameter that is given to us by the analysis phase of the lifecycle described in section 6. We then calculate the price vector that maximizes value according to that price vector. In section 6 we will explain how this procedure enables the business analyst to adapt a changing environment to the participants' needs.

5 BPM Layering

From our study so far we have realized that in order to calculate KPIs and improve the performance of the network, we need to connect SN to BPM. For example, the response time depends on how business processes are performed and can only be calculated based on a detailed description of the corresponding business processes.

The currently accepted Business Process Management Layers will serve as a basis for the implementation/enactment of SNs. These different layers exhibit different levels of abstraction and different purpose of the models involved. The introduction of SNs as an additional layer on top of that stack has the goal of simplifying the procedure of modeling business processes that achieve strategic goals and hence reducing the gap between the business experts' view and the IT view on business processes. The extended BPM layering is shown in Fig. 3.

The process models layer contains process models defined in an abstract technology-independent manner. The target user group is mainly the group of business analysts. The processes are modeled in a coarse-grained manner - the main functional blocks are identified and connected, and no implementation details are specified here.

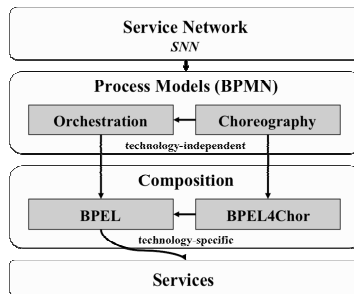


Fig. 3. Enhanced BPM Layering

This layer contains choreographies as well as orchestrations [9], [10], [11]. The composition layer is the one with technology-specific definitions of process models. The target user group is the technical analysts. Both, choreographies and orchestrations are represented at this layer in terms of artifacts of a particular technology and refined and enriched with implementation-specific details [12], [13].

The service layer represents the set of available services that are exposed for use by the composition layer. The implementations of services are transparent, as well as the platforms on which they are deployed.

6 Enhanced BPM Lifecycle

In the BPM state of the art, the different techniques and technologies focusing on business processes are connected with each other by the *BPM lifecycle*, presented in Fig. 4 on the left. It comprises six phases: *analysis*, *modeling*, *IT refinement*, *deployment*, *execution* and *monitoring*.

The *analysis* phase consists of the elicitation of the requirements for the business processes. The *modeling* phase revolves around the design of abstract, high-level business processes (e.g., BPMN models, abstract BPEL processes) from the requirements gathered during the analysis phase. The abstract business process models, while not immediately executable, outline the overall structure of the final processes to a level of detail suitable to humans. Often during the modeling phase there are defects that emerge in the collected requirements. In such cases, the lifecycle reverts to the analysis phase in order to solve the issues. Abstract business processes models are transformed into executable process models during the *IT refinement* phase. The *deployment* phase deals with deploying on the enterprise information infrastructure the executable processes models produced in the IT refinement phase.

Once deployed, executable business process models enter the *execution* phase, where they are finally run. During their execution, processes instances produce events conveying information about executed activities, their performance, exceptions and faults that occur, and more. The events are collected and analyzed in the *monitoring* phase to adapt business process instances, measure KPIs, keep track of the overall state of the system, capture trends and patterns in the current usage of the processes, etc. The data processed in the monitoring phase are also taken into account in the analysis phase of the following iteration of the BPM lifecycle, providing feedback to evolve the business process models.

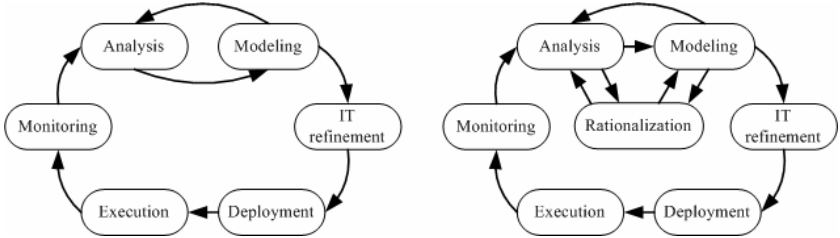


Fig. 4. The comparison between BPM lifecycle and enhanced BPM lifecycle

The canonical BPM lifecycle explained so far needs to be extended in order to benefit from the SNN and the analysis methods introduced in section 4. Fig. 4 (right side) presents the *Enhanced BPM Lifecycle*, obtained by adding a new phase, called *rationalization*, which deals with the modeling and analysis of SNN models.

The rationalization phase produces information which is used during either the modeling or analysis phase. We envision three ways of sequencing analysis, rationalization and modeling in the enhanced BPM lifecycle: *analysis–rationalization–analysis*, *modeling–rationalization–analysis* and *analysis–rationalization–modeling*. In the analysis–rationalization–analysis sequence (Fig. 5), the requirements resulting from the analysis phase are used in the rationalization one to create SNN models that represent the values flows among the participants. For example, the value calculation analysis described in section 4.1 is based on the requirements (e.g. an upper bound of the service delivery time) obtained from the analysis phase. The results are taken into account when modifying the abstract processes in order to maximize value. The new information on the desired characteristics of the process are then integrated with the previous set of requirements during another iteration of the analysis phase, during which takes place the resolution of conflicts that may arise between the original and new set of requirements.

In the modeling–rationalization–analysis sequence (Fig. 6), the existing abstract process resulting from the modeling phase is transformed into an SNN model through a *BottomUp transformation*. The value-maximizing analysis is then applied to the SNN model, producing a new set of requirements (e.g. a decreased upper bound of the service delivery time), which are integrated with the already existing ones in the upcoming iteration of the analysis phase. By analyzing SNN models extracted from abstract processes coming from outside the enterprise, it is possible to study the value flows from the point of view of the adopter of the processes and, for instance, take strategic decisions such as re-negotiate of the processes shared among participants.

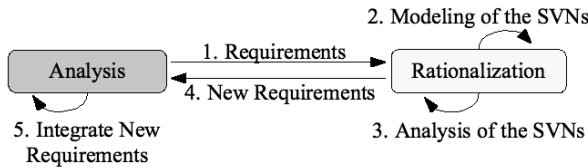


Fig. 5. The analysis-rationalization-analysis sequence

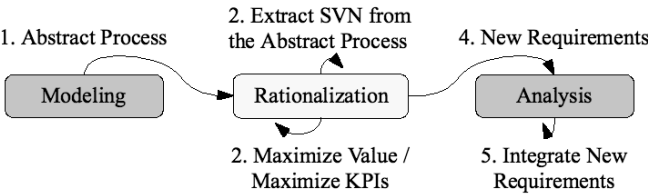


Fig. 6. The modeling-rationalization-analysis sequence

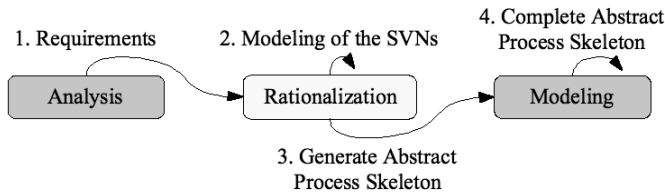


Fig. 7. The analysis-rationalization-modeling sequence

In the analysis–rationalization–modeling sequence (Fig. 7), the requirements resulting from the analysis phase are used in the rationalization phase to realize one or more SNN models. These models are transformed into *abstract process* models by applying *TopDown* transformations. The transformations use the correlations among offering and revenue relations to define the boundaries of the conversations involving the participants in the service networks.

The analysis–rationalization–modeling and modeling–rationalization–analysis sequences create a bond between SNN models and the abstract process models developed during the enhanced BPM lifecycle. Revenue and offering relations connecting parties in SNN models are translated into conversations and interactions in the abstract processes. Changes to SNN models (i.e., the removal of a revenue relation) can be mapped, through changes in the requirements, to changes to be applied to the abstract processes.

7 Conclusions and Future Work

Currently, we are witnessing an evolution in service oriented economies that need technological means to support them. In this paper we propose an architecture to coordinate business processes lifecycle and bridge existing gaps between technical and business perspectives. Our approach provides an abstract way to support business processes (in the SN level) and conversely a detailed description of the service network (in the BPM level). Next, we aim to formulate variations of optimization problems involving different kinds of KPIs and SLAs. The behavior of competing networks is also an open problem to be addressed possibly through means of game theoretic concepts. In this context, as interaction among different business roles in the process of providing a service is a key element in understanding and observing service systems, the field of game theory becomes a useful tool for identifying rules and strategies that optimize business objectives. As it was already done, all these studies have to be linked to the lifecycle management of business processes so that any progress made at the optimization level can be exploited by the business analysts.

References

1. Verna, A.: Reconfiguring the Value Network. *Journal of Business Strategy* 21(4) (July–August 2000)
2. Gordijn, J., Akkermans, H.: Designing and evaluating e-business models. *IEEE Intelligent Systems* 16(4), 11–17 (2001)

3. Caswell, N., Feldman, S., Nikolaou, C., Sairamesh, J., Bitsaki, M., Koutras, G.D., Iacovidis, G.: Estimating Value in Service Systems – A theory and an example. *IBM Systems Journal* 47(1) (2008)
4. Sampson, S.E.: *Understanding Service Businesses: Applying Principles of Unified Services Theory*. Wiley Press, Chichester (2001)
5. Spohrer, J., Maglio, P., Bailey, J., Gruhl, D.: Steps Towards a Science of Service Systems. *Computer* 40(1), 71–77 (2007)
6. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-Oriented Computing: a Research Roadmap. *Int. J. Cooperative Inf. Syst.* 17(2), 223–255 (2008)
7. Enhanced Telecom Operations Map The Business Process Framework For The Information and Communications Services Industry, TeleManagement Forum (2003), <http://www.tmforum.org>
8. Neely, A., Gregory, M., Platts, K.: Performance measurement system design: A literature review and research agenda. *International Journal of Operations & Production Management* 25(12), 1228–1263 (2005)
9. Business Process Modeling Notation (BPMN) Specification, Final Adopted Specification. Technical report, OMG (February 2006), <http://www.bpmn.org/>
10. Keller, G., Nüttgens, N., Scheer, A.-W.: Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK). Technical Report Heft 89, Universität des Saarlandes, Veröffentlichungen des Instituts für Wirtschaftsinformatik, IWi (1992)
11. Zaha, J.M., Barros, A., Dumas, M., ter Hofstede, A.: A Language for Service Behavior Modeling. In: CoopIS, Montpellier, France (November 2006)
12. Leymann, F.: Web Services Flow Language WSFL. IBM Corporation (2001), <http://www.ibm.com/software/solutions/webservices/resources.html>
13. Thatte, S.: XLANG: Web Services for Business Process Design. Microsoft Corporation (2001), http://www.getdotnet.com/team/xml_wsspecs/xlang-c/default.html

CD – JRA – 2.1.2

Paper [2]:

Bitsaki, M., Danylevych, O., Van den Heuvel, W.J., Koutras, G., Leymann, F., Mancioppi, M., Nikolaou, C., Papazoglou, M.:

Model Transformations to Leverage Service Networks.

In: *Fourth International Workshop on Engineering Service-Oriented Applications: Analysis and Design (WESOA 2008), Proceedings. Lecture Notes in Computer Science, Springer-Verlag (2008)*

Model Transformations to Leverage Service Networks

M. Bitsaki¹, O. Danylevych², W.J.A.M. van den Heuvel³, G.D. Koutras¹, F. Leymann²,
M. Mancipopi³, C.N. Nikolaou¹, M.P. Papazoglou³

¹ Computer Science Department, University of Crete, Greece
{bitsaki, koutras, nikolau}@tsi.gr

² Institute of Architecture of Application Systems, University of Stuttgart, Germany
{olha.danylevych, leymann}@iaas.uni-stuttgart.de

³ INFOLAB, Dept. of Information Systems and Management, Tilburg University, The Netherlands
{wjheuvel, m.mancipopi, mikep}@uvt.nl

Abstract: The Internet has catered for the transformation of traditional “stovepiped” service companies into global service networks fostering co-production of value to more effectively and efficiently satisfy the ever-growing demands of mundane customers. The catalyst of this change is the happenstance of Service Oriented Computing, providing a natural distributed computing technology paradigm for implementing and evolving such highly distributed networks of autonomous trading partners with coordinate and cooperative actions. However, how to faithfully (re-)map service networks, including value flows and inter-party interactions, to business processes and service realizations and vice-versa is still partly terra incognita.

In this paper, we introduce a semi-automatic model transformation approach for creating the abstract business processes that take place between trading partners from models representing the service networks, assuming extremely limited human-involvement focused on selecting reusable transformation patterns. This approach is explored and validated using a realistic case study reflecting best practices in the telecommunications industry.

Keywords: service networks, SOC, BPMN, business process management

1 Introduction

The services industry has become the leading contributor to business activities in developed economies, encompassing sectors such as logistics, education, publishing, finance, healthcare, telecom and government. The digitally networked service economy, driven by distributed computing technologies such as *Service Oriented Computing* (SOC) [1], is believed to revolutionize the way in which these companies conduct business, enabling exiting new business models such as service networks.

Service Networks (SNs) [5,15] leverage end-to-end service interactions between network partners that embody a succession of business processes typically cutting across organizational boundaries and spanning various geographical locations. Service networks properly sequence service activities according to the flow definitions in a business process model into end-to-end service constellations, assign work items to the appropriate human actors or groups, and ensure that both human- and systems-based activities are performed within agreed-upon timeframes and QoS criteria.

SOC is touted as the de-facto distributed enterprise-computing technology for developing and evolving SNs. In a SOC-based environment, business processes can be implemented as networks of choreographed services between- and orchestrated services within- network partners, relying on global standards including BPEL and WSDL. *Business Process Management* (BPM) [2] is a natural supplement to SOC through which business activities can be monitored and measured across business processes and services, while maximizing business value in service networks. In a nutshell, BPM has been evolved into a comprehensive lifecycle model that encompasses (graphical) process analysis & design, process execution, and process monitoring and reporting capabilities.

SOC-based design & development in tandem with BPM-based management of SNs should be grounded on a methodology, offering a consistent body of methods, notations and tools. As a first fundamental step, [5] proposed the *Service Network Notation* (SNN) as a novel modeling language enabling quantitative economic analysis of SNs to ascertain the optimal constellation of collaborative economic agents resulting in maximum economic value. This notation is equipped with concepts to model participants in service networks and their interactions in terms of *offerings*, i.e. services and/or functionalities provided to other participants, and *revenues*, the compensation for fulfillment of an offering to another participant, such as payments. Service networks described as SNN models can be analyzed to optimize the value generated in the network using financial metrics like cost, revenues and customer satisfaction. Further evolution of the SNN notation and its related analysis techniques will concern the simulation of service networks to

discover value anomalies (e.g., services or partnerships that do not produce as much value as expected) before the actual services are realized and deployed, and to perform different types of “what-if” analysis, such as study the changes of value flows upon replacements of services and partners, broadening or shrinking of the market (i.e., more participants join or live the service networks). It is the purpose of this paper to pursue new steps in the direction of the alignment of service networks and the underpinning business processes to realize such a comprehensive service network analysis, development and management methodology.

The development of business processes and services are already part of the current state of the art of subsequently BPM and SOC, and is well understood [12]. Therefore, we will not any further consider them in this work. Also, mapping business processes and services has been extensively scrutinized in the field of *Model Driven Engineering* (MDE) [11] and *Model Driven Architecture* (MDA) [10]. MDA is an effort of the Object Management Group providing the foundations and promoting the generation of programming code from models. Model transformations are subject of research since the 80s and are very relevant in for software development. Transformations are used to generate new models (for instance, source code in some programming language) from other models (e.g. UML2 Class Diagrams) using repeatable (automatic) processes expressed in the shape of rules [9].

Similarly to their applications to software engineering, MDE and MDA harbor huge potential benefits for BPM and service networks. In BPM, a wide variety of transformations have been devised to facilitate the generation, for instance, of executable business processes from abstract ones (such as, but not limited to, [17,18]). In the ambit of service networks, one of the links currently missing is how to exploit the information about value flows among participants contained in SNN models to streamline the generation of those business processes that are the backbone of the service networks, and vice-versa how to extract service network representations from existing abstract business processes. In this paper we bridge this gap by introducing a transformation approach for constructing business processes in from models of service networks and the other way around, which is accelerated through the usage of process interaction patterns that can be injected during the transformation processes. This results into an approach that is one of continuous (re-)design, scoping, refinement and adjustment of service network- and abstract business process models.

The paper is organized as follows: Section 2 outlines the SN4BPM architecture on which our transformation is grounded. Section 3 then introduces a realistic running example of a service network for customer- and network fault handling. In Section 4 a staged transformation method to map SNs into business processes is subsequently described, after which Section 5 elaborates the transformation mechanisms in detail. Finally, Section 6 concludes the paper with conclusions and directions future work.

2 The SN4BPM Architecture

The SN4BPM (Fig. 1) entails a stratified architecture that serves as the foundation for realizing business

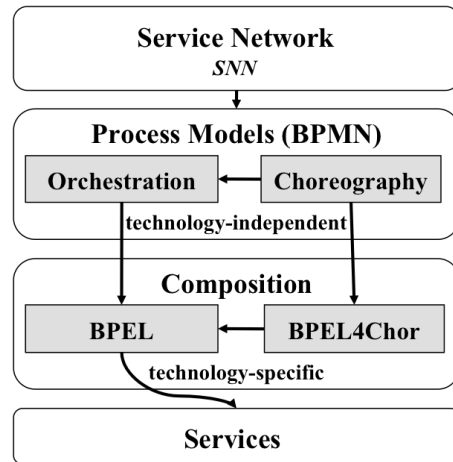


Fig. 1. The SN4BPM stack.

processes in service networks. This architecture unifies the BPM [11] and SOC [1] standards stacks for realizing service networks, fostering a clean separation of concerns among the devising of the strategies, partnerships and their effects, the abstract business processes realizing them, the executable business processes and the underpinning IT infrastructure.

At the top layer of this architecture, the *Service Network* layer encompasses service network models that serve as the basis of analyzing, simulating and optimizing networked constellations of business partners, each of which contributes to the network processes, while adding value.

The *Process models* layer deals with modeling abstract business processes with languages such as the *Business Process Modeling Notation* (BPMN) [4]. Abstract business process models are implementation agnostic, and harness processes as orchestrations of services, each of which realizes an activity and is executed according to a control flow that

may be private or public to the partners in the SN. Abstract business processes thus omit implementation details that are necessary to the execution of the business processes (e.g., the endpoint of the services to

interact with). Abstract business processes can be partitioned into *business process fragments* that group independent and cohesive subsets of interactions among the participants. For example, an abstract business process *CustomerProblemHandling* in a telecommunications service network can be partitioned into two cohesive business process fragments, viz., *ProblemDiagnostics* and *ProblemFixing*.

In particular, abstract processes are realized as executable processes in the *Composition* layer, where they can be rendered as *choreographies*, which provide a global view on the (inter- and intra-organizational) multiparty collaborations focusing on the message-based communication among partners, or *orchestrations*, which specify and connect into executable workflows the activities performed and the message exchanges performed by a participant or a service. Executable business processes are technology-dependent, and are usually modeled using languages such as the *Web Services Business Process Execution Language* (WS-BPEL) [7] and BPEL4Chor [8], respectively focusing on orchestrations and choreographies. The *Services* layer provides the set of discrete services available in the service network, relying on open standards based message backbone, enabled by SOC infrastructural plumbing technologies such as an *Enterprise Service Bus* [16].

3 The Service Network Notation

The transformation approach will be illustrated and explored with a simple and realistic running example concerning a service network for resolving resource and service problems that are reported by Telco clients, e.g., connection problems, in a telecommunications service network comprised of consumers, intermediaries, telco service providers and suppliers. This case study is based on a description of standard, end-to-end business processes in the *eTOM Business Process Framework* [3]. For reasons of understandability we will now briefly explain the basic concepts in SNN.

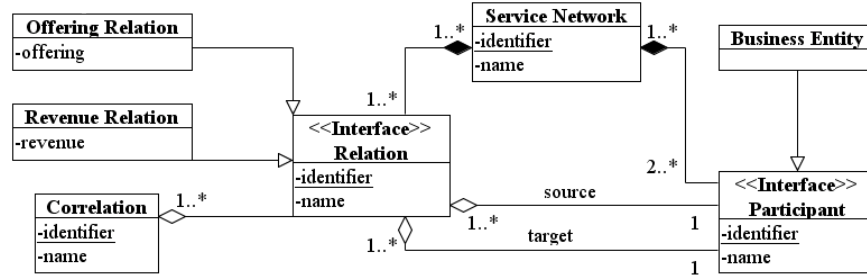


Fig. 2. The SNN meta-model.

The Service Network Notation meta-model is depicted in Fig. 2 using a UML2 Class Diagram. Participants in a service network are modeled as *Business Entities*. Interactions among network participants are modeled in terms of *offering relations* and *revenue relations* connecting the corresponding business entities. Each offering and revenue relation has a *source* and *target* participant: the source participant of an offering relation provides a commodity to the target. Similarly, the source of a revenue relation transfers economic value to the target in return for a value offering. A *Correlation* aggregates all the offering and revenue relations that represent interactions grouped in the same business process fragment.

Fig. 3 depicts our running example as an SNN model based on the meta-model of Fig. 2. It focuses on two essential processes in fault resolution, the *Customer Fault Resolution* and the *Network Fault Resolution* business process. The Customer Fault Resolution process conceptualizes the customer's procedure for reporting a fault to a Customer Service Representative (CSR): after the reception of a trouble ticket, the CSR delegates the resolution job to a Field Agent, after which the Field Agent intervenes at the Customer's site to solve the issue. When the issue is solved, the Customer pays for the intervention.

The SNN model captures this scenario among the three business entities, *Customer*, *CSR* and *Field Agent* as follows. The provisioning of the resolution service from the CSR to the Customer is represented as a directed arrow labeled as the *Customer Support* offering. The Field Agent supplies the Customer with the *Fix Failure* service and the CSR with the *Solve Ticket* service. The Customer pays for the intervention by generating revenue for the CSR, which is modeled with the dotted directed arrow labeled *Pay Intervention* pointing to the CSR. The *Customer Support*, *Solve Ticket*, *Fix Failure* and *Pay Intervention* relations are correlated (the star-like symbol connecting them) because they all take place within the

context a single business process fragment, called *Customer Fault Resolution*, with a clearly demarcated start and end.

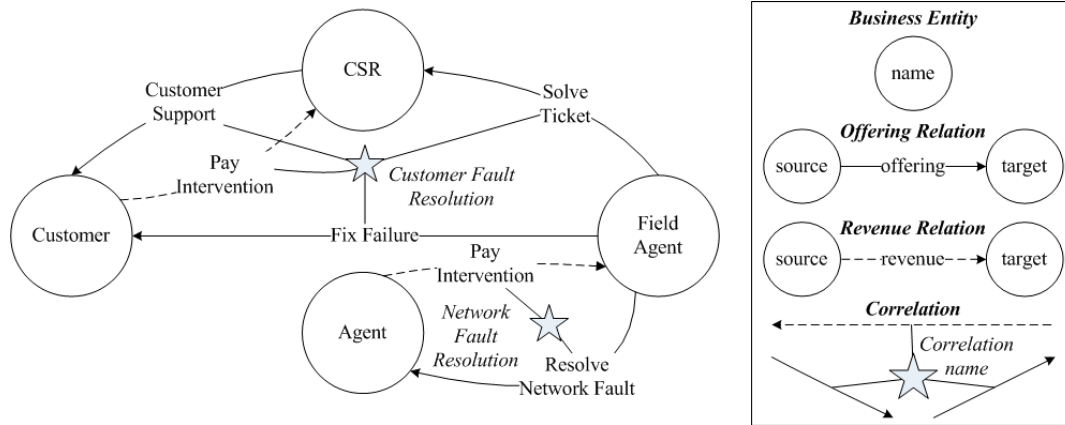


Fig. 3. An SNN model representing the eTOM example.

The Network Fault Resolution business process involves the *Field Agent* and *Agent* business entities. The Agent notifies the Field Agent of a failure in the network that requires fixing, and the Field Agent solves the problem. In the SNN, this is modeled by having the Field Agent business entity providing the Agent with the *Resolve Network Fault* offering, while the Agent remunerates the Field Agent through the *Pay Intervention* revenue. Since the *Resolve Network Fault* offering relation and the *Pay Intervention* revenue relation take place in the Network Fault Resolution business process, they are associated by a correlation called *Network Fault Resolution*.

4 Transformations in the Enhanced BPM Lifecycle

The development practices and activities for developing service-enabled processes in SNs are organized and integrated by the *enhanced BPM lifecycle* introduced in [5], which is depicted in Fig. 4.

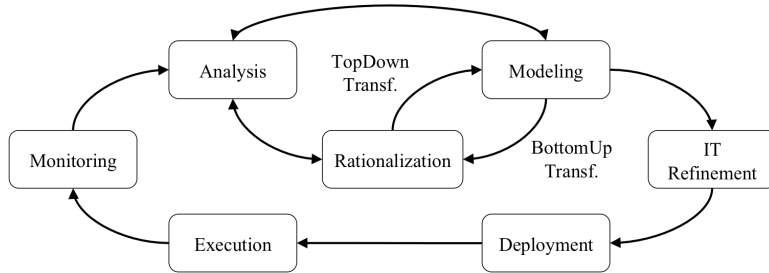


Fig. 4. The enhanced BPM lifecycle.

This lifecycle extends the traditional BPM lifecycle by introducing a phase called *Rationalization* that deals with the design, optimization and simulation of SNN models. The *Analysis* phase elicits and collects business process requirements, and resolves requirement inconsistencies and incompleteness. The *Modeling* phase addresses design, maintenance and evolution of abstract business process models. The *IT refinement* phase centers on realizing executable process models, which are then deployed on the IT infrastructure during the *Deployment* phase. The *Execution* phase enables the execution of processes, and generating execution trails which are used in the *Monitoring* phase to adapt particular process instances while still running, detect trends and patterns in the current usage of the processes, keep track of the overall state of the system, etc.

Transformations are the mortar that bind the new elements of the enhanced BPM lifecycle with the well-established practices of the standard BPM lifecycle, enabling the analyst to move from the Modeling phase to the Rationalization phase and vice-versa as shown in Fig. 4. The connections between the Analysis and Rationalization phases are based on one direction on modeling SNN models that capture the requirements specified on the Analysis phase, and on the other on extracting new requirements from optimized SNN models. The transformations that are scrutinized in this paper foster in the lifecycle the bi-directional synchronization of artifacts designed during the Modeling- and Rationalization phases, viz. the SNN model and the abstract business process model that are logically associated through the stratified SN4BPM architecture.

5 The Transformation Approach

Enhanced business process management is facilitated when the business analyst may easily progress from one phase to the next and back to iteratively develop service-enabled processes for new service networks, and to incrementally deal with changes in existing ones. While not incremental and iterative, the SN2BPM transformation approach represents an initial step towards this vision, predefining the overall process of mapping service network models into abstract business process model through a multi-step approach, involving *TopDown* and *BottomUp* transformations, named after the relative directions in traversing the SN4BPM stack of Fig. 1. Mapping SNN models (belonging at the Service Networks layer) to abstract business process models (at Process Models layer in the stack) is achieved through TopDown transformations, while changes in business process models can be propagated to the service network through BottomUp transformations.

The BottomUp and TopDown transformations are combined to enable round-trip engineering of service-enabled processes in service networks, keeping the overall design synchronized after modifications to abstract business processes or the service network. Fig. 5 provides an overview of the basic workings of the transformation approach indicating whether steps are completely automatic (gears), or that require some decisions taken by an analyst (sticky figure), or involve *Business Interactions Patterns* (BIPs). A BIP is a generic and reusable template of a business process fragment that can be applied to concepts in the SNN model (e.g., correlation). In particular, a BIP summarizes *roles* played by participants in a business process fragment, workflows structuring the activities performed by the roles, and message-based interactions that occur among the different roles.

As depicted in this Fig. 5, the BottomUp transformation is composed of five steps, one requiring human intervention and the other automatic, that produce a well-formed SNN model from a BPMN model. The first step requires the analyst to label the message-flows at BPMN level that represent revenue and offering relations between participants at SNN level. From then on, the BottomUp transformation is fully automated. The TopDown transformation creates a BPMN model from the information embedded in a SNN model. The transformation is divided in two main phases, the first required and the second optional. The steps from 1 to 4 (phase 1) are completely automatic and start from a SNN model to result in a BPMN model that describes pools, lanes that divide the pool in independent business process fragment, sub-processes captured in the lanes, and also defines interrelationships between sub-processes through message-flows that mirror interactions at SNN level. Step 5 (phase 2) is semi-automatic, as it requires human involvement for the selection of the BIPs to be applied.

Both transformations are based on the mapping between the SNN and BPMN meta-models that is presented in Section 5.1. The business interaction patterns are examined more in depth in Section 5.2. The BottomUp and TopDown transformations are respectively described in detail in Section 5.3 and 5.4.

5.1 Model Mappings for SNN and BPMN model transformation

Both the TopDown and BottomUp transformations are defined on the basis of the mapping between the SNN and the BPMN¹ meta-models presented in Fig. 6 (see the bold bi-directional arrows in this figure).

Each SNN *Service Network* corresponds to a BPMN *process*. *Participants* in SNN models are mapped to *pools* in BPMN. SNN correlations group interactions among participant in different business process fragments. A participant involved in interactions spread over multiple business process fragments has multiple lanes in its pool, one per fragment. For instance, if a participant takes part in interactions that are

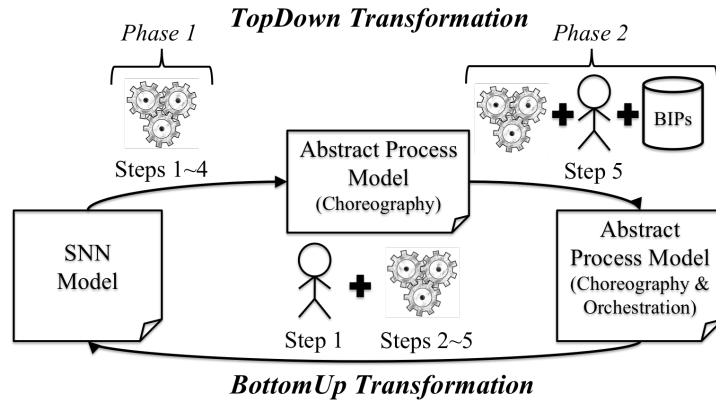


Fig. 5. Overview of the transformation approach.

¹ Extrapolated from the BPMN meta-model published on WSPER.ORG based on [4], and available at: <http://www.wsper.org/bpmn10.html>

divided into three different business process fragments, its respective pool will contain three lanes. In BPMN, for all practical purposes, pools with a single lane and pools without lane are equivalent. Pools and lanes contain *workflows* (i.e., *activities* connected by control flow constructs). *Sub-processes* are special activities that abstract entire workflows. Workflows and sub-processes can be recursively nested into each other. Activities communicate with other activities in different pools through *message-flows*, which represent message exchanges. Both *offering-* and *revenue relations* in SNN are mapped to message-flows in BPMN.

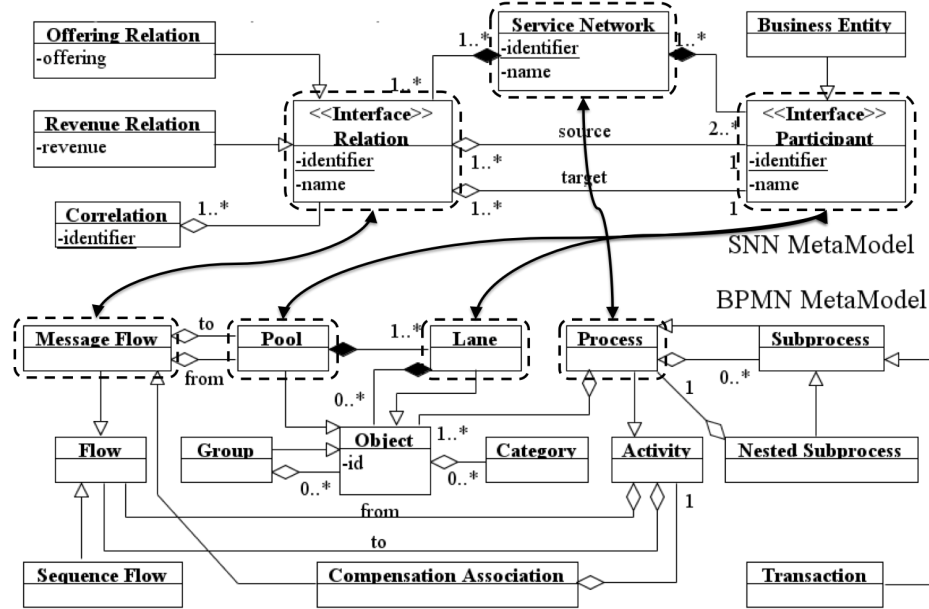


Fig. 6. The mappings between the SNN and BPMN meta-models.

These mappings are based on two critical assumptions about the structure of the BPMN. Interactions among participants in the SNN are represented by message-flows in BPMN (i.e., the participants in the service network carry out their interactions over message-based conversations). Secondly, each pool only comprises business logic (workflows) of one particular business fragment. Workflows in different lanes within the same pool are independent, i.e., they are not connected through control flows. The first assumption allows the BottomUp transformation to derive the offering and revenue relations to be represented in the resulting service networks. The second assumption enables the BottomUp transformations to cluster message exchanges between participants in different abstract business processes as correlations in the SNN model.

5.2 Business Interaction Patterns Under the Lens

The Business Interaction Patterns play a central role in creating via the TopDown transformation BPMN models that are immediately usable in a BPM lifecycle to, for instance, automate the generation of executable BPEL processes as proposed in [17]. SNN models describe which interactions take place between the participants, but not how these interactions are structured (e.g., in terms of message exchanges and activities performed by the involved participants). It is not possible for an automatic transformation to “guess” how participants will communicate with each other to carry out the interactions described at SNN level, and this is mirrored by the coarse granularity of the intermediate BPMN models resulting from the first four steps of the TopDown transformation. BIPs are meant to improve the level of detail of the business processes resulting from the transformations by providing a structure (based on message-exchange communication) for the abstract interactions at SNN level in the shape of business process fragments that are selected by humans and are automatically “plugged” in the BPMN models resulting from the first transformation phase.

BIPs are based on existing business process standards in industry. Such as RosettaNet *Partner Interface Process* (PIPs) [13] and *Architecture of Integrated Information Systems Value Chain Reference Model* (ARIS VCRM) [14], which are widely adopted reference models for standard, multi-party collaborative business process models. In the remainder of this paper we assume that BIPs are modeled as context-

independent BPMN models where each role is represented by a pool. Each pool captures a well-structured sequence of internal processing steps; we refer this as a workflow. Workflows in different pools are logically interconnected via message-flows. Fig. 7 and Fig. 8 present two examples of BIPs that we have developed, respectively the “On Behalf Of” and “Provide Service Within Deadline” BIPs.

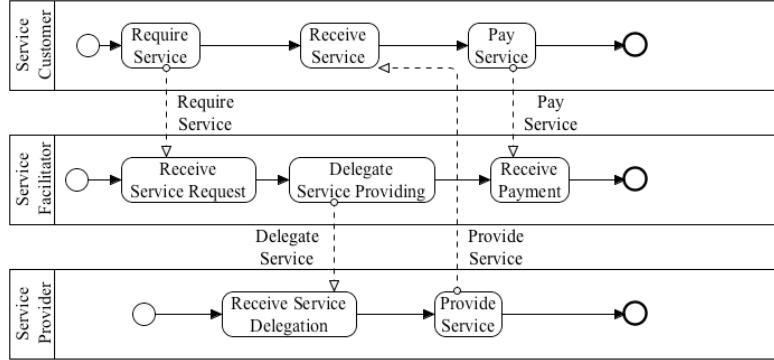


Fig. 7. The “On Behalf Of” Business Interaction Pattern.

The “On Behalf Of” BIP presented in Fig. 7 defines a BPMN template for a chain of value-relationships between multiple partners participating in a correlation within the service network, and is computed as a transitive relation between a *Provider* and a *Customer* comprising a cohesive path of message-exchanges that involve the sub-contractor (*Service Facilitator*) as “man in the middle”. A series of delegated service offerings are subsequently traversed during the execution of a particular business process fragment, e.g., the *CustomerFaultResolution* process. Note that for reasons of brevity, we concentrate on a simple “On Behalf Of” correlation involving three participants; however, in practice we have already encountered correlations involving larger chains for which we have designed more complicated BPMN templates (e.g., for the automotive repair service scenario introduced in [15]).

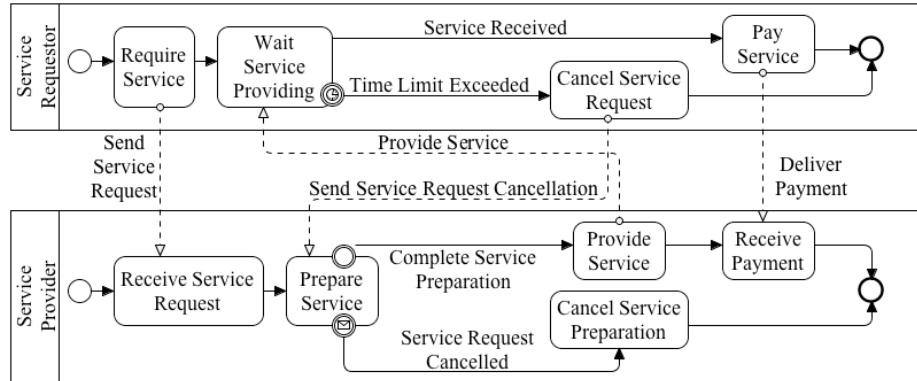


Fig. 8. The “Provide Service Within Deadline” Business Interaction Pattern.

The “Provide Service Within Deadline” BIP of Fig. 8 defines a BPMN template prescribing a prototypical business process between a Service Provider and a Service Request that is operated under temporal constraints (e.g., deadline), specifying how the process behaves when the constraint (deadline) is met, how it deals with deadline violation, and the remuneration of the service, if it is delivered in time.

In practice, by applying a BIP the human analyst “explodes” the offering- and revenue-based interactions grouped at SNN level by a correlation into a message-based conversation at the abstract business process level. Typically, such message-based conversations are expressed as Message Exchange Patterns [11]. For the sake of simplicity, in our approach we rely on the assumption that a single BIP suffices to describe the structure of a whole business process fragment. Without this assumption, the user should specify a set of BIPs to be applied to a given fragment, and how to connect and merge the workflows that each BIP creates in a participant’s pool (but this is already on the list as future work).

5.3 The BottomUp Transformation: Extracting SNN Models from Abstract Process Models

The BottomUp transformation extracts a service network model from an abstract business process. As explained in Section 5.1, the correspondences between business entities in SNN and participants in BPMN

are rather straightforward. On the other hand, it is much harder to extrapolate from an abstract business process, in which the participants interact over message exchanges, what kind of revenue and offering relations occur. The first step of the BottomUp transformation, requiring human intervention, tackles this issue by having an analyst label the message-flows in the source BPMN model that represent interactions between participants that have to be represented at SNN level in the shape of revenue or offering relations. This approach relies on the assumption that each relation at SNN level is represented by (at least) one message flow in the source BPMN model.

The BottomUp transformation produces an SNN model from a *source* BPMN model using the following five steps (see also Fig. 5):

1. *Label message-flows that represent revenue- or offering-based interactions:* the analyst labels the message-flows that represent offering or revenue relations at SNN level as shown in Fig. 10. For instance, the message-flow “Solve Ticket” represents an offering relation with the Field Agent as source and the CSR as target, while “Pay Intervention” represents a revenue relation from the Customer to the CSR.
2. *Collapse sub-processes in the source BPMN model:* expanded sub-processes (i.e., sub-processes that show their internal workflow) are transformed in *collapsed* sub-processes as explained in [4].
3. *Create the participants:* for each pool in the source, create a participant in the SNN model, and name the participant after the pool.
4. *Create offering and revenue relations:* for each message-flow connecting an activity in the pool of participant A with an activity in the pool of participant B, do as follows:
 - 4.1. If the a group of message flows is labeled as a revenue relation, then create a new revenue relation in the SNN model connecting participant A to participant B and using the name of the message-flow as the revenue offering associated with the newly created revenue relation R. Participant A and B are respectively source and target of R.
 - 4.2. If the message flow is labeled as an offering relation, then create a new offering relation in the SNN model connecting participant A to participant B and using the name of the message-flow as the offering associated with the newly created offering relation O. Participant A and B are respectively source and target of O.
 - 4.3. If neither 4.1 nor 4.2 apply, then ignore the message-flow.
5. *Create correlations:* group the message-flows in the source BPMN model according to the business process fragments they belong to. This is obtained by:
 - 5.1. Grouping the workflows in the participants’ lanes in *workflow-groups*. Two workflows belong to the same workflow-group if they are connected by a message-flow. Namely, a workflow-group is the *transitive set*² of workflows that are connected by a message-flow.
 - 5.2. Grouping message-flows in *message-flow-groups*. Two message-flows belong to the same message-flow-groups if they originate from or end in workflows grouped in the same workflow-group. Alternatively, a message-flow-group is the transitive set of message-flows that connect workflows in the same workflow-group.
 - 5.3. For each message-flow-group, create a correlation connecting all the offering- and revenue relations that have been created starting from the message-flows in the message-flow-group.

Consider the BPMN model in Fig. 10 that models the key abstract business processes in our running example. The first step in the transformation is to label the message-flows as revenue- or offering-relations. The second step is to collapse the sub-processes. After these first two steps, the resulting BPMN model looks like the one presented in Fig. 11. Fig. 9 exemplifies the three remaining steps in the BottomUp transformation. The third step of the transformation creates the participants in the SNN model (result shown in Fig. 9, step 3). The revenue and offering relations in the SNN model are created in the fourth step (outcome presented in Fig. 9, step 4). Finally, the correlations are added to the SNN model during the fifth step (result in Fig. 9, step 5). The workflow groups are two: the workflows in the “Network Fault Resolution” sub-processes in the Agent and Field Agent pools (they are connected by the message-flows “Resolve Network Fault” and “Pay Intervention”), and the workflows named “Customer Fault Resolution” in the pools Field Agent, CSR and Customer (transitively connected by the “Customer Support”, “Solve Ticket”, “Fix Failure” and “Pay Intervention” message-flows).

² In other words, conversations at SNN level are identified in step 5 by calculating the transitive sets of the message-flows on the basis of the workflows they connect, and creating a new correlation for every transitive set, connecting all the revenue and offering relations originated by the message-flows in that transitive set.

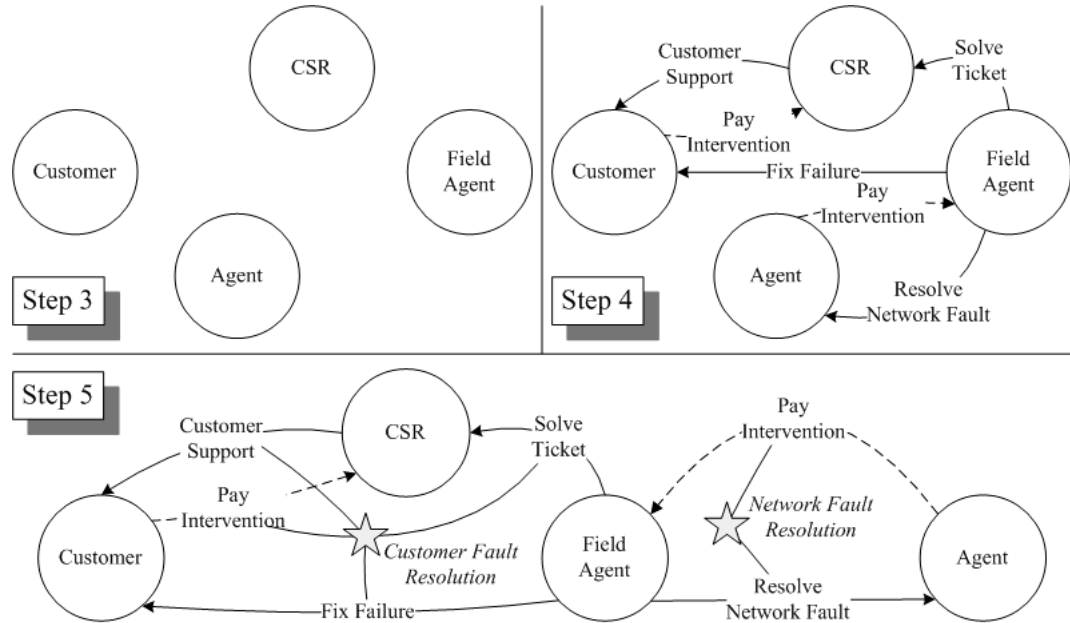


Fig. 9. The results of the 3rd, 4th and 5th step of the BottomUp transf. on the BPMN model in Fig. 10.

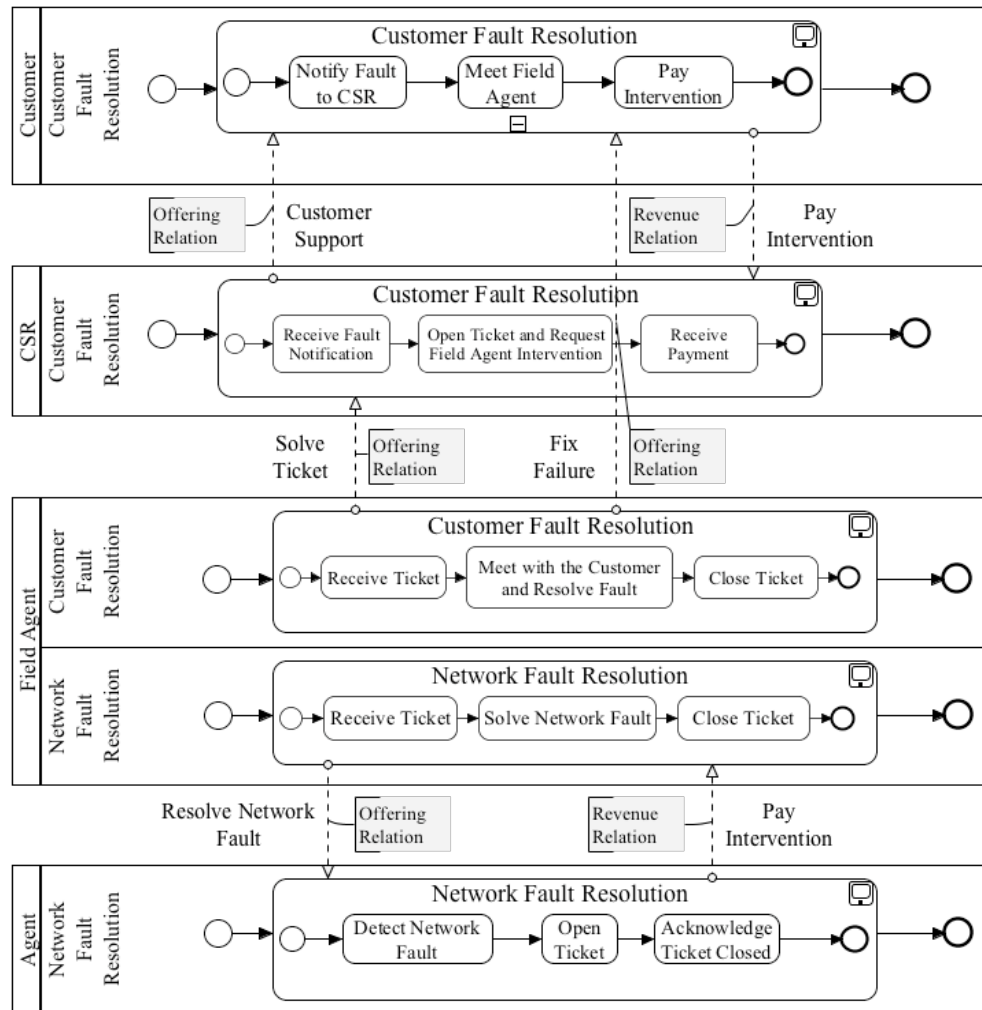


Fig. 10. A BPMN model based on the eTOM example presented in Section 3.

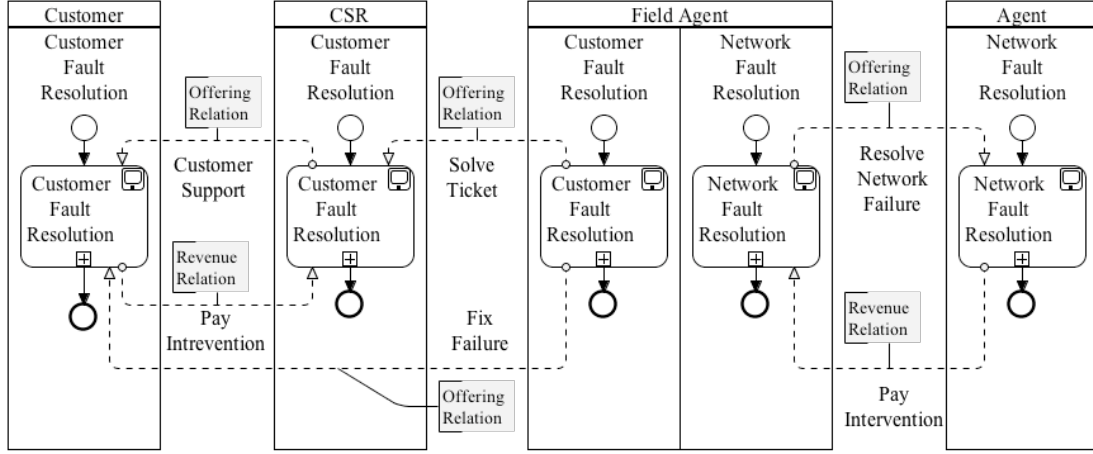


Fig. 11. The BPMN model resulting from applying the transformation to the SNN model of Fig. 3.

5.4 The TopDown Transformation: Creating Abstract Process Models from SNN Model

In the following we introduce the TopDown transformation to refine a SNN into a BPMN model as a two-staged process.

PHASE 1 (Required): Produce an Abstract Business Process Choreography Model

This abstract business process model is rendered in BPMN, and defines the process choreography tracking the globally visible message flows between network partners. It is automatically generated in the following manner:

1. *Create the pools:* for each participant in the SNN model, create a pool in the BPMN model with the same name.
2. *Create the lanes and simple workflows:* for each correlation in the SNN model a participant is involved in, create a lane (named after the correlation) in the participant's respective pool. If a relation in the SNN model does not belong to any correlation, it is treated as it belonged to a correlation comprising only itself. Each lane thus created this way is filled with a workflow made of a *start event*, a sub-process named as the correlation, and an *end event* sequentially connected in this order.
3. *Create revenue message-flows:* for each revenue relation R in the correlation C connecting the source participant A and the target participant B , create a message-flow from the sub-process in the lane C of the pool A to the sub-process in the lane named C of the pool B . Let the revenue of the relation R be *revenue*. The newly created message-flow is labeled as: "[Revenue] revenue".
4. *Create offering message-flows:* for each offering relation O in the correlation C connecting the source participant A and the target participant B , create a message-flow from the sub-process in the lane C of the pool A to the sub-process in the lane named C of the pool B . Let the offering associated to the relation O be *offering*. The newly created message-flow is labeled as: "[Offering] offering".

PHASE 2 (Optional): Produce an extended Abstract Business Process Model

The resulting model not only defines the process choreography, but also captures the private workflows of network partners. This phase is optional if the level of detail in the BPMN model resulting from the first phase is deemed insufficient, and thus requires further refinement through the application of one of more business interaction patterns. The phase is composed of the following semi-automatic step:

5. *Apply BIPs:* the application of a BIP requires the user select a correlation (*selected correlation*) in the SNN model (again, relations not involved in any correlation are treated as if they belonged in a correlation comprising only themselves). The participants that are source or target of offering and revenue relations that are comprised in the selected correlation are called *involved participants*. The user provides a mapping from the roles in the BIP to the involved participants. The BIP must define as many roles as the involved participants. For each involved participant, the workflow in

the mapped role's pool is copied inside the lane created at step 2 for the selected correlation in the pool corresponding to that participant. Finally, all the message-flows in the BIP are automatically copied into the BPMN model and connected to the same activities as they are in the BIP.

Fig. 11 visualizes the result of applying the first four steps to the SNN model in Fig. 3. The Field Agent participant in the SNN model has offering and revenue relations grouped in different correlations (“Customer Fault Resolution” and “Network Fault Resolution”), the Field Agent pool has two lanes, one per correlation (second step). The revenue relation Resolve Network Fault in the SNN model is transformed into a message-flow, labeled “Resolve Network Fault” (step 3) which connects the two sub-processes in the lanes of the Agent and Field Agent pools that are created because of the correlation.

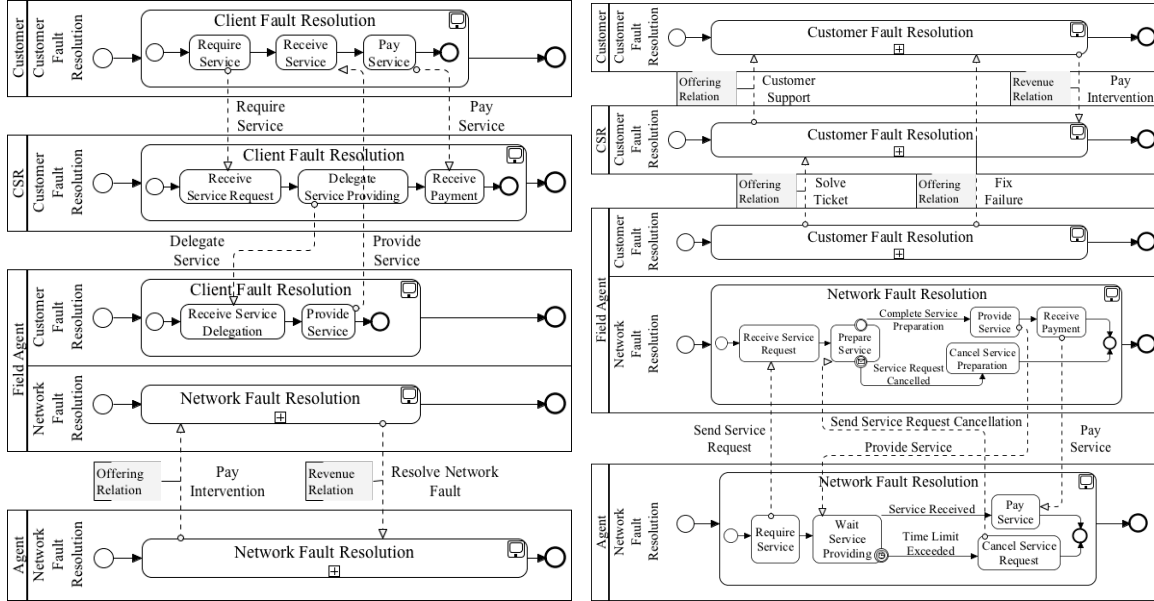


Fig. 12. Result of application of the “Provide Service Within Deadline” and “On Behalf Of” BIPs.

Fig. 12 shows the results of the application of the fifth transformation step (“Apply BIPs”) to our running examples. At the left hand side in the result of applying at the fifth step the “Provide Service Within Deadline” BIP to the correlation “Network Fault Resolution” by mapping the role “Service Provider” to the Field Agent, and “Service Requestor” to the Agent, is exemplified. At the right hand side of this figure, it demonstrates the application of the “On Behalf Of” BIP to the BPMN model in Fig. 11 by respectively mapping the “Service Requestor”, “Service Facilitator” and “Service Provider” roles to the Customer, CSR and Field Agent pools.

6 Future work and Conclusions

The service industry, the leading contributor to developed economies, is quickly transitioning towards the digital networked economy that is leveraged through distributed computing technologies including Service Oriented Computing. In conjunction with its natural complement Business Process Management, SOC is touted as the ideal paradigm to develop, evolve and manage sophisticated service networks that enact successions of automated end-to-end business processes that traverse several enterprises and geographical locations. However, this vision is far from a reality and many organizations are still fixated on orchestration of internal processes, witnessing the popularity of languages such as BPEL.

Service networks promise to effectively leverage and bridge between business-like requirements such as value and revenues, and the IT enactment through Service Oriented Architecture and Business Process Management. Service networks have recently catered a wide interest, which resulted, among other proposals, in the Service Network Notation that describes the interactions among participants in a service network in lieu of offering- and revenue relations.

In this paper, we have proposed and explored a semi-automatic approach for constructing business processes in service networks, or redefining service networks after changes to business processes. This approach is almost completely automated, and assumes restricted involvement of human experts restricted

to the selection of the business interaction patterns that best capture recurrent skeletons of interactions between and within processes partners in the service network, and to the labeling in the business processes of message exchanges that need to be represented at service network level. The proposed transformation approach is grounded on a series of mappings between the meta-models of SNN and BPMN models, and is formalized through procedural transformation algorithms. Note that in this article we have concentrated on the conceptual underpinnings of the solution, e.g., we have restricted ourselves to defining pseudo-code for the procedural transformations, which can be straightforwardly mapped into code or transformation rules using MDA technologies such as the *Query/View/Transformation* (QVT) standard from the Object Management Group.

The results presented in this paper are core results in nature. Extensions and refinements are needed in various directions. Firstly, we intend to further elaborate the transformation approach to make it incremental and iterative, and to improve the BottomUp transformation to use pattern recognition mechanisms to automatically extract the revenue and offering relations at SNN level by applying “backwards” the BIPs (i.e., recognize process fragments that fit BIPs and generate the corresponding relations at SNN level). In addition, we intend to further explore and elaborate the transformation approach in several real case studies. Thirdly, we wish to further extend the library of business process interaction patterns. The BIP library, currently populated with a handful of patterns, will be extended with existing patterns that can be easily extracted from industrial reference models, standard protocols and industrial best practices. Moreover, we intend to investigate more complex transformation scenarios where multiple business interaction patterns occur in the same business process. Lastly, we are in the process of implementing the transformations in the *Value Network Tool* (<http://vnt.tsl.gr/>), the integrated development environment that supports the design of SNN models.

7 References

1. Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. In: Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE 2003). IEEE Press, New York (2003) 3—12
2. Mc Goveran, D.: An Introduction to BPM and BPMS. *Business Integration Journal* (April 2004)
3. TeleManagement Forum. Enhanced Telecom Operations Map The Business Process Framework For The Information and Communications Services Industry. <http://www.tmforum.org>
4. Object Management Group (OMG). Business Process Modeling Notation Specification. OMG Standard (February 2006)
5. Bitsaki, M., Danylevych, O., van den Heuvel, W.J., Koutras, G.D., Leymann, F., Mancipopi, M., Nikolaou, C., Papazoglou, M.P.: An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network. To appear in: *ServiceWave* (2008).
6. Nitzsche, J., van Lessen, T., Karastoyanova, D., Leymann, F.: BPEL^{light}. In: *BPM*. Volume 4714 of LNCS., Springer (2007) 214—229
7. Organization for the Advancement of Structured Information Standards (OASIS). Web Services Business Process Execution Language Version 2.0. OASIS Standard (April 2007)
8. Decker, G., Kopp, O., Leymann, F., Weske, M.: BPEL4Chor: Extending BPEL for Modeling Choreographies. In: *IEEE International Conference on Web Services (ICWS'07)*. IEEE Press, New York (2007) 296—303
9. Sendall, S., Kozaczynski, W.: Model Transformation - the Heart and Soul of Model-driven Software Development. *IEEE Software* **20**(5) (September/October 2003) 42—45
10. MDA Guide Version 1.0.1. Object Management Group, Document Nr: omg/2003-06-01 (June 2003)
11. Havey, M.: *Essential Process Modeling*. O'Reilly (2005)
12. Papazoglou, M.P., van den Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal* **16**(3) (2007) 389—415
13. RosettaNet Consortium. Partner Interaction Processes (PIPs). <http://www.rosettanet.org>
14. Scheer, A.W., Nüttgens, M.: ARIS Architecture and Reference Models for Business Process Management. In: *BPM*. Volume 1806 of LNCS., Springer (2000) 301—304
15. NS Caswell, C. Nikolai, J. Sairamesh and M. Bitsaki: Estimating value in service systems: a case study of a repair service system, *IBM System Journal* **47**(1) (2008) 87—100
16. D.A. Chapell: *Enterprise Service Bus*. O'Reilly (2004)
17. C. Ouyang, M. Dumas, A.H.M. ter Hofstede, W.M.P. van der Aalst: From BPMN process models to BPEL web services. In: *IEEE International Conference on Web Services (ICWS'06)*. IEEE Press, New York (2007) 285—292
18. T. Hornung, A. Koschmider, J. Mendling: Integration of heterogeneous BPM Schemas - The Case of XPDL and BPEL. In: *CAiSE'06 Forum Proceedings, Theme: Trusted Information Systems* (2006)

CD – JRA – 2.1.2

Paper [3]:

Wetzstein, B., Danylevych, O., Leymann, F., Bitsaki, M., Nikolaou, C., van den Heuvel, W.-J., Papazoglou, M.:

Towards Monitoring of Key Performance Indicators Across Partners in Service Networks.

In: *Proceedings of the International Workshop on Service Monitoring, Adaptation and Beyond (MONA+ 2008).*(2008)

Towards Monitoring of Key Performance Indicators Across Partners in Service Networks

Branimir Wetzstein¹, Olha Danylevych¹, Frank Leymann¹, Marina Bitsaki², Christos Nikolaou², Willem-Jan van den Heuvel³, and Mike Papazoglou³

¹ Institute of Architecture of Application Systems, University of Stuttgart, Germany
lastname@iaas.uni-stuttgart.de

² Computer Science Department, University of Crete, Greece
lastname@ts1.gr

³ Dep. of Information Systems and Management, Tilburg University, Netherlands
W.J.A.M.vdnHeuvel, mikep@uvt.nl

Abstract. In an ever increasing dynamic environment, companies are forced to co-operate in order to meet customer needs effectively. They set up Service Networks (SN) trying to create a win-win situation for all participants of the network. The calculation of value in an SN is based on key performance indicators (KPIs) which measure the performance of underlying cross-organizational business processes. As for the calculation of KPIs of these processes monitoring information from several participants is needed, in an SN it is no more sufficient for the participants to monitor just KPIs of their internal processes, e.g., by using Business Activity Monitoring technology. The participants now have to provide a set of monitoring events to the other partners in the SN. In this paper, we describe an approach to monitoring of KPIs across partners in a service network. An SN is mapped to a service choreography and a monitoring agreement is created which specifies how KPIs are decomposed to events that participants in the choreography have to provide. We present our approach based on a case study from the telecommunications domain.

1 Introduction

In today's networked economy, companies are not independent, isolated entities, but they must act in a concerted manner to survive in an ever increasing dynamic environment. Thereby, interacting companies build networks to serve their joint customers in a dynamic manner, focusing on optimizing their financial benefits at the individual and network level. Recently, Service Networks (SNs) have been proposed to model such networks and analyze and optimize company's business collaborations [1]. SN is a graph-based approach to model a business environment as a set of business partners and their relations. SNs reside on a high abstraction business level depicting partners as nodes and their offering and revenues as edges. Modeling a business landscape as SN, allows, on the one hand, calculating the value gained by a single partner when joining the collaboration network. On the other hand, an SN perspective gives the possibility to measure the value of the whole network. The value calculation is used for measuring the profitability of the SN, which can lead to adaptation of SNs, for example, through outsourcing.

Service Networks focus on cooperations between partners in terms of offerings and revenues and don't detail the concrete interactions between the partners. In addition, the dependencies between the actors in an SN don't necessarily express the temporal dependencies between the partner interactions. Each offering - each single edge - in the SN is realized through a set of complex interactions between the partners. The partner interactions are not of interest on SN level but represent one of the main concerns of the level of business processes and choreographies as part of business process management [2]. The refinement from SNs to executable processes has been motivated in [3] and first steps towards mapping of SNs to service choreographies are described in [4].

The value calculations in an SN are based on a set of Key Performance Indicators (KPIs). KPIs are business metrics which are used for measuring the performance of underlying business processes of the SN. Traditionally, companies have measured the performance of their internal processes using established concepts such as Business Performance Management and technologies like Business Activity Monitoring [5]. In the setting of an SN, this is no more sufficient. Partners now have and want to share SN relevant information of their internal processes with other partners. In order to do so the partners have to provide monitoring events or already measured metrics to the "outside", so that the overall performance of the SN can be evaluated.

In this paper, we propose a method of how to model and monitor KPIs across partners in a service network. We assume that the SN is mapped to service choreography descriptions, as described in [4]. Based on the choreography description, we describe how KPIs are decomposed to events each partner has to provide for the overall KPIs to be calculated. We introduce in this context the concept of a *monitoring agreement* which specifies the monitoring events each partner has to provide. The monitoring agreement includes partner descriptions, the events which each partner provides, and how these events are aggregated to calculate the overall KPIs of the SN. We describe the concepts based on a case study from the telecommunications domain.

This paper is organized as follows: Section 2 describes the case study we have chosen for evaluating the concepts of this work. Section 3 gives an overview of our approach. The definition of monitoring agreements is described in Section 4. Section 5 sketches aspects related to runtime monitoring. Section 6 positions our approach among existing work and finally, Section 7 concludes the paper and presents future work.

2 Case Study

The case study discussed in the following is based on the "Enhanced Telecom Operation Map" (eTOM), which is a reference model for business processes of the telecommunications industry [6]. In particular, we describe a service network that is formed in order to set up a new DSL service.

Figure 1 depicts the involved parties of the DSL installation process and the offering flows between the parties. The main actors are the Subscriber (customer) and the Service Provider, which offers DSL products to its customer. The functionalities of the Service Provider relevant to this part of the model are distributed among the Call Center, Service Agent and Field Agent. The Subscriber interacts with the Call Center to order the DSL line and to report problems or complaints. In order to make the installation, the Call

Center performs some actions (like checking availability) and forwards the request to the Service Agent. The Service Agent is responsible for the setup and configuration of the order. He finally contacts the Field Agent to perform the service installations at the customer site. The Billing Agent is responsible for setting up the monthly billing procedure for the customer. In our case study, we further assume that Call Center, Service Agent, and Field Agent are not part of the SP company, but separate organizations that have been outsourced.

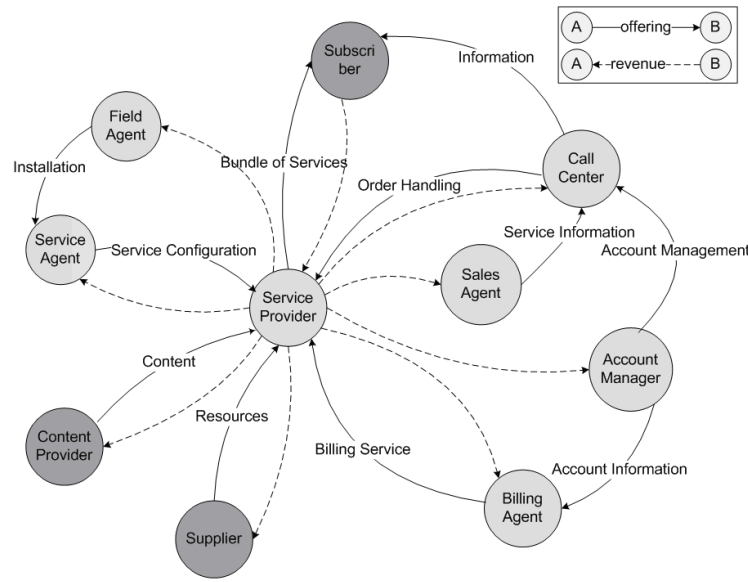


Fig. 1. The Service Network for a new DSL Service Set-Up

The value of Service Provider (SP) in the presented service network can be calculated as described in [3]. The calculation takes into account the following KPIs: the revenues, the costs the SP pays for the needed resources to provide the service, and the satisfaction of customers (subscribers). These three "top-level" KPIs are aggregations of other finer-grained KPIs. The revenues are calculated based on the prices of the service and the number of customer orders in a certain period. The costs are a function over labor and contract rates with the Call Center and Field Agents and internal process costs and the number of employees together with some additional costs (material costs). The customer satisfaction is a function over the order fulfillment lead time, deadline adherence, price of service, number of customers that cancelled their order, number of customers that complained during the time period of order processing, and perfect order fulfilment (order processed in full and in time as requested by customer without customer complaints).

Some of the described KPIs can be calculated with information available already on the level of service network. For example labor rates and the service price are already fixed in service level agreements between the service provider and customer.

Other KPIs, the ones we focus on in this paper, are measured on BPM layer based on underlying business processes, such as order fulfillment time, deadline adherence, number of complaints etc. The Service Provider wants to measure these KPIs in a timely manner in order to be able to calculate its value in this service network. For the calculation of these KPI, he depends on information from its business partners. Assume e.g. the KPI perfect order fulfilment. In order to calculate this KPI, we need information from the Call Center (time of order receipt, and number of customer complaints), Service Agent (information on whether we could install and configure the requested product as wished by the customer), Field Agent (time of product delivery and installation at customer site). That means for the calculation of the KPI information from more than one business partner is needed. This information should also be provided in a timely manner, i.e. based on events as they happen in the process.

3 Overview of the Approach

In this paper, we take a top-down approach in which service networks are mapped to service choreographies and further refined to executable business processes [3]. We distinguish between three layers and between a functional and non-functional view, as shown in Figure 2.

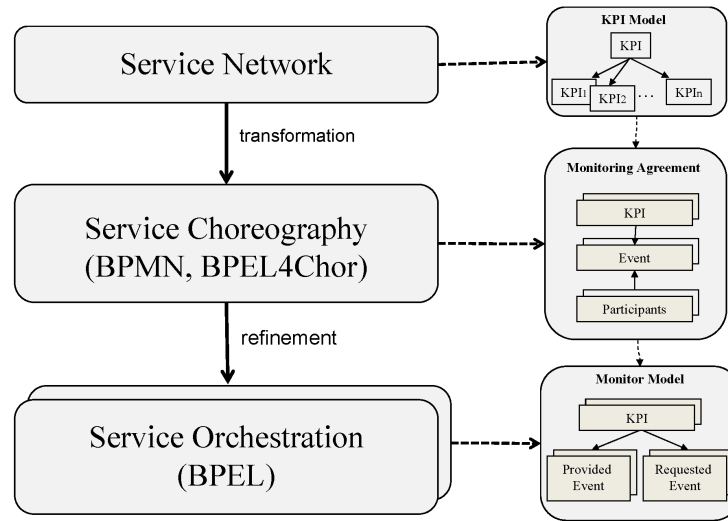


Fig. 2. Overview of the Approach

In the functional view, on the topmost layer, SNs are modeled. An SN specifies the interactions between partners on a very high level in terms of offerings and revenues. In the next step, the SN is mapped to a service choreography, which specifies the

message exchanges between the partners in the SN. This mapping can be performed semi-automatically [4]. The choreography can be first modeled in a technology-independent notation such as BPMN, and then later transformed to a technology specific choreography specification, such as BPEL4Chor [7]. On the orchestration level, each participant in the choreography implements his part of the process and exposes it to the outside as a Web service. This implementation of the process can be done in WS-BPEL.

In the non-functional view, on SN level, value calculations of the SN as a whole and of the participants in the SN are performed. Value is calculated based on KPIs, such as revenues, costs, and customer satisfaction. KPIs can again be defined based on other KPIs. For example, customer satisfaction can be defined based on the customer satisfaction index, the number of customer complaints, deadline adherence, and average order fulfillment lead time. All KPIs of the SN and their calculations are part of the *KPI model*.

In order to measure the KPIs of the SN in a timely manner, we have to specify how they are to be monitored based on operational business processes. On choreography level, the public processes involving message exchanges are modeled, serving as an agreement between partners on how they communicate together. We argue, that on this level one should also *agree* on which events each partner has to provide in order to calculate the KPIs of the KPI model. This is because on choreography level no private process information is modeled, and thus events based on public process models also should not lead to privacy issues. For example, in order to calculate the KPI *Order Fulfillment Lead Time*, the Call Center has to provide an *OrderReceived* event which contains an *order receipt date*, while the Field Agent provides the *ProductInstalled* event with the *product installation date*. The monitoring agreement specifies KPIs which are to be evaluated for the service choreography, and how they are decomposed to events each partner has to provide. The agreement involves also the definition of event formats, and monitoring mechanisms which define how the events can be retrieved at process runtime.

The monitoring agreement has two purposes for each participant in the choreography: (i) it serves as a requirements specification, on which events the participant has to provide; (ii) the monitoring agreement also specifies how KPIs are calculated based on all events provided by all partners. Based on this information, a participant can subscribe for the events of other partners, in order to be able to calculate the KPIs internally, if he wants to. Therefore, a participant creates a *monitor model*, which defines how the provided events are to be created and how the needed events from other partners are requested, and how the KPIs are calculated based on those events.

4 Definition of Monitoring Agreements

In this section, we describe how monitoring agreements between partners in an SN are created. A monitoring agreement specifies which information partners have to provide in order to enable monitoring of KPIs.

4.1 Service Choreography in the Case Study

Figure 3 shows an excerpt of a choreography description in BPMN for the case study. It models the business process for setting up the DSL line from customer request to

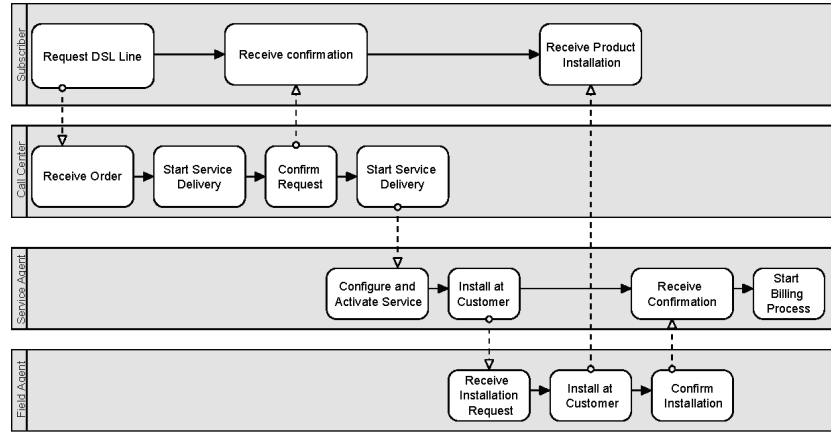


Fig. 3. BPMN Choreography Description for the Case Study SN

installing of the product at customer's site. The choreography involves four partners. The subscriber places an order in the call center. After checking its availability, the call center sends the order to the service agent, which configures the package. The field agent finally installs the product at customer site. The process is distributed between several business units and business partners. For example, we assume that Call Center, Service Agent, and Field Agent are all separate companies (business partners) which were outsourced by Service Provider.

In Table 1 we have listed a set of KPIs which are interesting in this Service Network. For each KPI, we specify its calculation, and define the events which have to be provided by the partners in the choreography. As shown in the Table, the first KPI can be provided by only one partner, the other two need events from more than one partner. Note that events have to be correlated when used in the calculation functions. In the above cases this can happen based on an *Order Identifier* which is set when a new order is received, and which is then used until shipment. This *Order Identifier* has to be part of each event and also of the messages which are exchanged between the partners. Correlation of events is an important concept when specifying the calculation of KPIs, and is well-known from Complex Event Processing (CEP) [8] which is mostly used for implementation of BAM solutions.

4.2 Specification of Monitoring Agreements

As shown in the last section, for the calculation of KPIs events from different partners are needed. In intra-organizational BAM, one also needs to model events and instrument different information systems which emit these events at runtime. Typically, these events are published to a publish/subscribe eventing infrastructure (a topic) which the BAM tool subscribes to. The difference in the SN setting is that monitoring is performed based on cross-organizational processes.

In the case of cross-partner monitoring, where different organizations are involved, one has to *agree* on events (and their content) which are provided by different partners.

KPI	Metric Calculation	Partner. Provided Event
# of Received Orders	count(OrderReceived)	Call Center.OrderReceived
Order Fulfillment Lead Time	t(ProductInstalled) - t(OrderReceived)	Call Center.OrderReceived Field Agent.ProductInstalled
Perfect Order Fulfillment	Order Fulfilment Time < 14 days & ServiceConfigured.status = "in full" & not(ReceivedCustomerComplaint)	Call Center.OrderReceived Field Agent.ProductInstalled ServiceAgent.ServiceConfigured Call Center.ReceivedComplaint

Table 1. KPIs and their Calculation

Note that in this case there are also several non-technical aspects involved, such as privacy issues. Companies want to restrict insight into their internal processes as much as possible. On the other hand, a certain degree of openness and monitoring support, might be part of a service offering. Discussion of these non-technical issues is out of scope of this paper. As we base our monitoring agreement, and therein specified events, on service choreographies which consist of public process descriptions, privacy issues are minimized. However, we do not restrict ourselves to choreographies; if participants want (or need) to provide events which go beyond information contained in the choreography, they are free to do so.

Figure 4 shows the main concepts needed for the specification of a monitoring agreement. A *monitoring agreement* is specified for a *service choreography* description which contains a set of *participants*, and for each participant a public *process model* which defines its behavior as part of the choreography. If BPEL4Chor is used as a service choreography language, the process model is specified in (an abstract profile of) WS-BPEL. The *monitoring agreement* defines a set of *indicators*. An *indicator* is defined based on a *function* which calculates an indicator value based on already defined indicators or based on *events* which are provided by *participants*. Note that we support the special case that no indicators but only events are specified in an agreement. That case is needed if one only wants to track the progress of a process instance (which is signaled by events). Functions can contain boolean, arithmetic and aggregate operators, among others. An *event* contains a set of *properties* which can be arbitrary data items, consisting of a name and a type. An *event* definition can contain a reference (not shown in the Figure) to a process element (process, activity, variable) thus specifying where in the process the event is emitted. In addition to the specified concepts, one needs to specify how the events can be obtained at process runtime, e.g. by publishing and subscribing to a topic (see Section 5).

Listing 5 shows an excerpt of the monitoring agreement for the KPI Order Fulfilment Lead Time as defined in Table 1. It is calculated based on two events provided by the Call Center and Field Agent. The monitoring agreement document references the

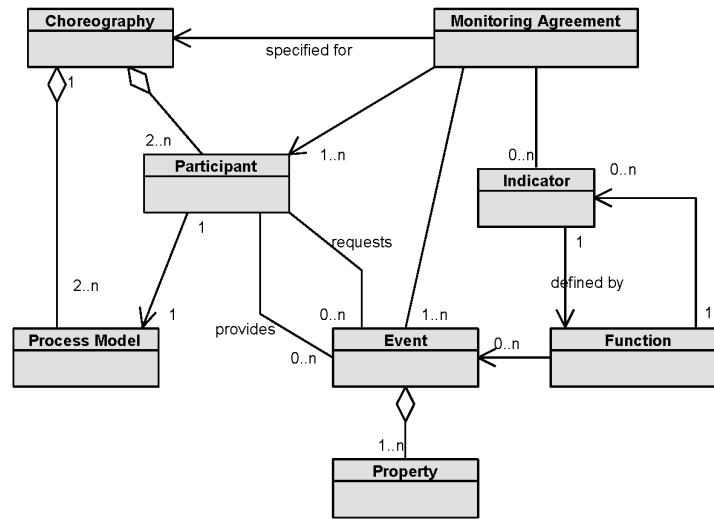


Fig. 4. Overview of the Main Monitoring Agreement Concepts

choreography description and the participants defined in the choreography description. The calculation specification uses a predefined function *duration* and correlates on the *orderId* which is a property of both events.

5 Monitoring Architecture

After creation of the choreography description and the monitoring agreement, each partner implements its internal process according to the choreography, and implements his part of the monitoring agreement. Figure 6 shows a high-level view of the monitoring architecture.

Each partner implements its role in the choreography description, e.g., by refining the abstract BPEL description from the BPEL4Chor description to an executable BPEL service orchestration. This service orchestration implements its role in the choreography. In the same manner, each partner has to implement its part of the monitoring agreement, thus providing events to the other partners and receiving events from the other partners.

As we focus on BAM as performance measurement technology, we assume that each partner has an internal BAM implementation. Based on the monitoring agreement, partners have to make sure that they provide events to the outside. As shown in Figure 6, a possible solution is to establish a publish/subscribe topic (a.k.a. publish/subscribe channel), e.g. based on WS-Notification, which is used by all partners in the choreography. The partner thus has to generate events, which might involve prior instrumentation of services and systems, and publishes it to the topic. All partners which are interested in this event and have accordingly subscribed to it, receive this event.

Note that the implementation of the publish/subscribe channel can be hosted by one of the partners or also by a third party. The channel is also only responsible for routing

```

1 <monitoringAgreement>
2   <serviceChoreography name="eTomChor:eTom-Choreography"/>
3   <participants>
4     <participant name="eTomChor:CallCenter">
5       <providedEvents>...</providedEvents>
6       <requestedEvents>...</requestedEvents>
7     </participant>
8     <participant name="eTomChor:FieldAgent">
9       <providedEvents>...</providedEvents>
10    </participant>
11  </participants>
12  <indicators>
13    <indicator name="Order Fulfillment Lead Time" unit="hours">
14      <duration>
15        <event name="ProductInstalled" property="installationDate" />
16        <event name="OrderReceived" property="receiptDate" />
17        <correlation>
18          <equal>
19            <event name="ProductInstalled" property="orderId" />
20            <event name="OrderReceived" property="orderId" />
21          </equal>
22        </correlation></duration></indicator>
23    </indicators>
24    <events>
25      <event name="OrderReceived">
26        <property name="orderId" type="xsd:string"/>
27        <property name="receiptDate" type="xsd:date"/>
28      </event>
29      <event name="ProductInstalled">...</event>
30    </events>
31  </monitoringAgreement>

```

Fig. 5. Monitoring Agreement for Order Fulfillment Lead Time (simplified)

of events, but not for correlation and aggregation of events for the calculation of KPIs. In this architecture, this is performed by each partner on its own.

6 Related Work

Business activity monitoring approaches in the context of monitoring of KPIs of business processes focus on intra-organizational processes. There exist several research approaches [5, 9] and products [10] which deal with evaluation of process metrics in near real time and their presentation in dashboards. They all have in common that events are emitted as the process is executed, collected by a process monitor and evaluated in near real time. Some solutions focus on monitoring of BPEL processes [9], while others are more general and support an extensible architecture via event adapters [10]. To the best of our knowledge there is no work yet which considers monitoring of KPIs of service networks or service choreographies in a cross-organizational scenario.

Service Level Agreements (SLA) are similar to our problem in that they involve monitoring in a cross-organizational setting. Thereby mostly two partners, the service consumer and the service provider, agree on certain service QoS, typically technical characteristics such as availability and response time. The SLA specifies also how the agreed QoS levels are to be monitored and what happens in case of violations. Web Service Level Agreement (WSLA) [11] is an approach to modeling and monitoring

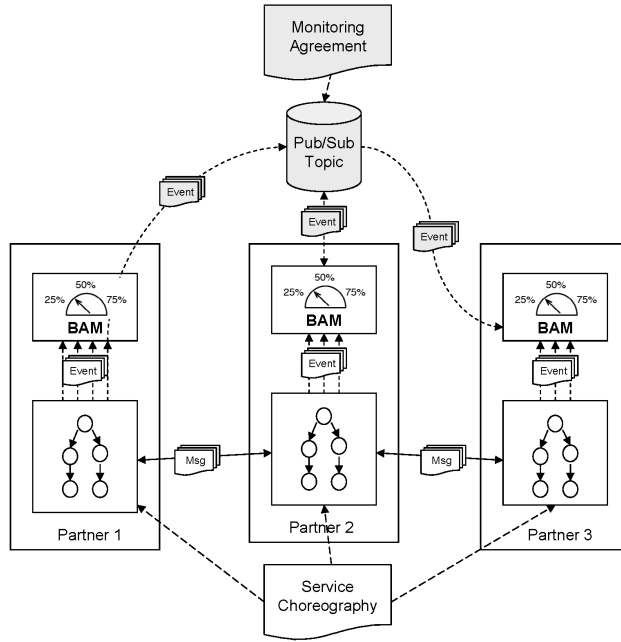


Fig. 6. Monitoring Architecture

of SLAs in the context of Web services. A WSLA-based agreement specifies involved parties, SLA parameters and objectives they agree on, the underlying metrics including measurement directives, and penalties in case of violations. The commonalities with monitoring in our context are that in an SLA partners also *agree* on metrics and how they are to be monitored. In our case, we also need an agreement on metrics and measurements between partners. However, in our case the focus is on monitoring (of potentially more than two partners) and not on guaranteeing certain KPI values. We are also mostly interested in monitoring of process metrics, not in low level QoS metrics, and in particular deal also with cross-partner metrics, which require event correlation, and which are not being dealt with in frameworks such as WSLA.

7 Conclusions and Future Work

In this paper we have presented an approach to monitoring of KPIs in service networks. We have motivated the approach based on a case study, showing that partners have to provide monitoring events to the outside for calculation of KPIs when participating in service networks. Thereby, KPIs are decomposed to events each partner has to provide. We have introduced the concept of a monitoring agreement which is specified on the level of service choreographies. The monitoring agreement describes the calculation of the KPIs based on monitoring events and the obligations of each partner concerning the

provision of those events. Finally, we have sketched a possible monitoring architecture, which is based on a publish-subscribe infrastructure used by all partners.

Our future work includes refining and implementing the framework presented in this paper. We want to base the implementation of the framework on BPEL4Chor for the specification of choreography descriptions and WS-BPEL for implementing orchestrations. The realization includes specifying the monitoring agreement metamodel in detail, including its linkage to BPEL4Chor choreography descriptions, and the semi-automatic generation of a WS-BPEL based monitoring implementation. The monitoring solution should provide both near real time monitoring in BAM fashion and retrieval of monitoring information on demand.

Acknowledgments The research leading to these results has received funding from the European Community's 7th Framework Programme under the Network of Excellence S-Cube Grant Agreement no. 215483.

References

1. Caswell, N.S., Nikolaou, C., Sairamesh, J., Bitsaki, M., Koutras, G.D., Iacovidis, G.: Estimating Value in Service Systems: A Case Study of a Repair Service System. *IBM Systems Journal* **47**(1) (2008) 87–100
2. Weske, M.: *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)
3. Bitsaki, M., Danylevych, O., Van den Heuvel, W.J., Koutras, G., Leymann, F., Mancipopi, M., Nikolaou, C., Papazoglou, M.: An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network. In: *ServiceWave2008*. (2008)
4. Bitsaki, M., Danylevych, O., Van den Heuvel, W.J., Koutras, G., Leymann, F., Mancipopi, M., Nikolaou, C., Papazoglou, M.: Model Transformations to Leverage Service Networks. In: *Proceedings of the 4th International Workshop on Engineering Service-Oriented Applications (WESOA 2008)*, Springer-Verlag (2008)
5. Jeng, J.J., Schiefer, J., Chang, H.: An Agent-based Architecture for Analyzing Business Processes of Real-Time Enterprises. In: *Proceedings of the 7th International Conference on Enterprise Distributed Object Computing (EDOC '03)*, Washington, DC, USA, IEEE Computer Society (2003) 86
6. TeleManagement-Forum: *Enhanced Telecom Operations Map (eTOM) - The Business Process Framework For The Information and Communications Services Industry* (2003)
7. Decker, G., Kopp, O., Leymann, F., Weske, M.: BPEL4Chor: Extending BPEL for Modeling Choreographies. In: *ICWS, Salt Lake City, USA* (2007)
8. Luckham, D.: *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Professional (2002)
9. Roth, H., Schiefer, J., Schatten, A.: Probing and Monitoring of WSBPEL Processes with Web Services. *Proceedings of the The 8th IEEE International Conference on E-Commerce Technology (CEC-EEE'06)* (2006)
10. Wahli, U., Avula, V., Macleod, H., Saeed, M., Vinther, A.: *Business Process Management: Modeling Through Monitoring Using WebSphere V6.0.2 Products*. IBM, International Technical Support Organization (2007)
11. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *J. Netw. Syst. Manage.* **11**(1) (2003) 57–81