Grant Agreement N° 215483

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.

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

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

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:

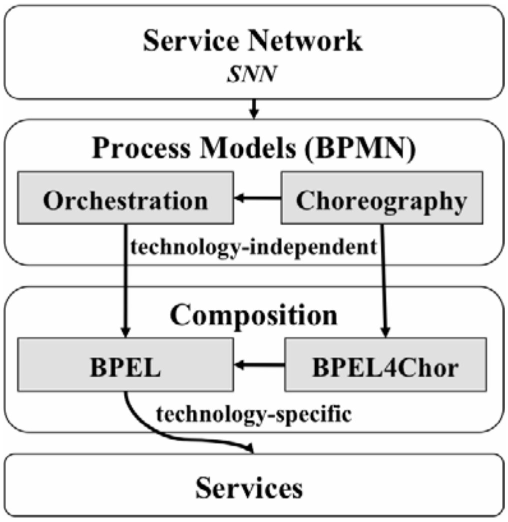

Figure 1: The Enhanced BPM Layering for the SN4BPM Architecture.

- 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.

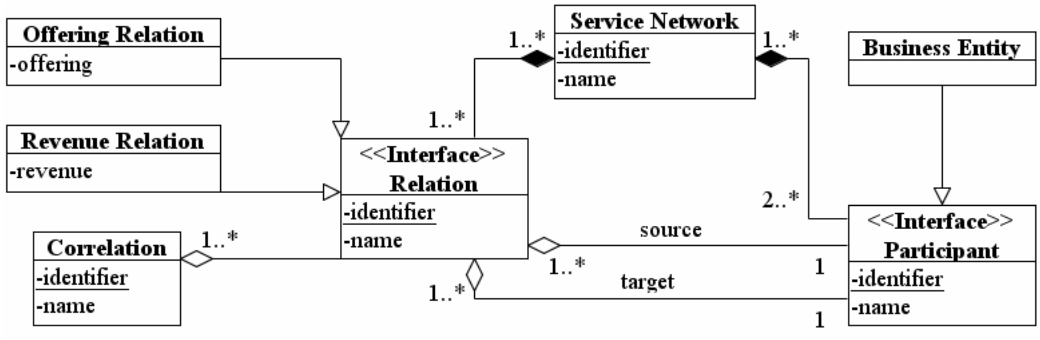

Figure 2: The SNN meta-model as a UML2 Class Diagram. [1]

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).

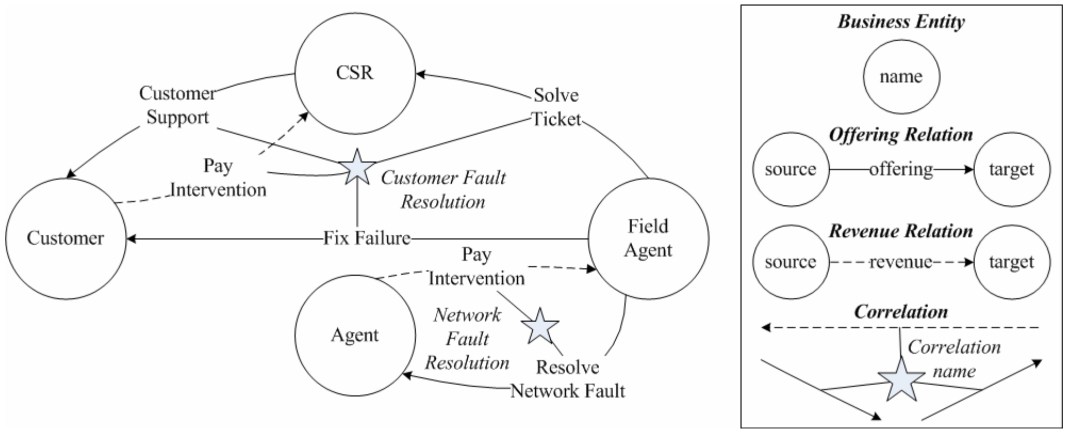

Figure 3: An example of SNN model based on the eTOM framework. [2]

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.

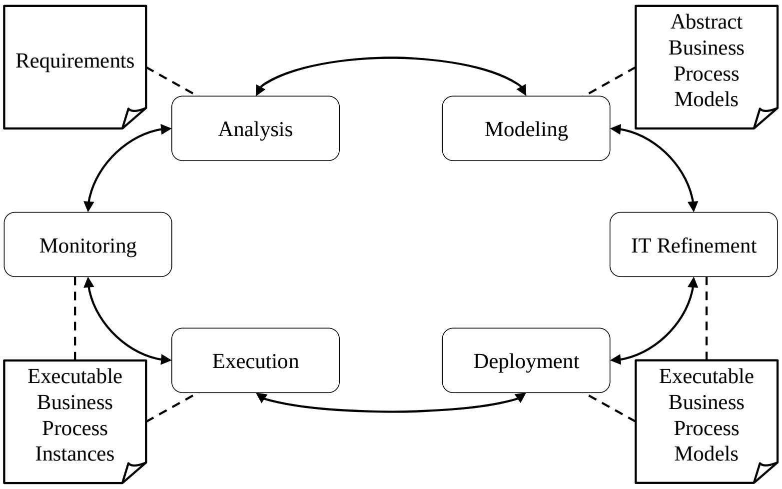

Figure 4: The established BPM Lifecycle. [1]

- **Modeling:** it covers the realization of abstract business process models (e.g. BPMN or Abstract BPEL models) that fulfill the requirements resulting from the analysis phase. The abstract business processes resulting from this phase need not to be detailed enough to be run on the infrastructure; they should have a level of detail suitable for humans (i.e. business analysts and business process modelers) to understand the overall structure of the final (executable) processes to be realized later in the lifecycle. It is common that inconsistencies in or incompleteness of the requirements emerge during the modeling of abstract business processes. If this is the case, the lifecycle reverts to the analysis phase to solve the issues.
- **IT Refinement:** the goal is to obtain executable business process models from the abstract ones resulting from the Modeling phase, a procedure known as "refinement". If, during the refinement, it turns out that some requirements can not be fulfilled (e.g. a certain QoS attribute can not be satisfied due to limitations of the current technology) or that the abstract business processes can not be refined into executable ones (e.g. in case there are constructs adopted in the abstract process models that have no correspondence with what is offered by the technologies adopted for the executable processes), the lifecycle reverts to the Modeling phase, and through it possibly to the Analysis phase (if there is no way to fix the issues by changing the abstract process models).
- **Deployment:** executable business process models are deployed on the infrastructure (e.g. a BPEL execution engine such as *Apache ODE* [10]), and are made ready for their execution. If technical difficulties emerge during the deployment (e.g. the engine does not support some features required to run the executable process models), the lifecycle reverts to the IT Refinement phase to solve the issues.
- **Execution:** the execution of an executable business process model results in a *business process instance* (e.g. a running WS-BPEL process). The Execution phase runs mostly in parallel with the Monitoring phase described below.
- **Monitoring:** business process instances produce events, such as the completion of a given activity or the incurring into a fault or exception, that are used in the Monitoring phase to assess the state of the instances. The Monitoring phase is complex and it involves a number of technologies and methodologies that are matter of current research in S-Cube, and more generally in the BPM community. The interested reader is referred to [11] for a comprehensive overview of the state of the art of monitoring approaches.

The established BPM Lifecycle presented in Figure 4 is turned into the Enhanced BPM Lifecycle presented in [1] by adding the **Rationalization** phase (see Figure 5):

- **Rationalization:** this phase deals with the modeling, analysis and optimization of SNN models that are modeled on the basis of the requirements elicited in the Analysis phase, or that are extracted from the abstract business process models resulting from the Modeling phase.

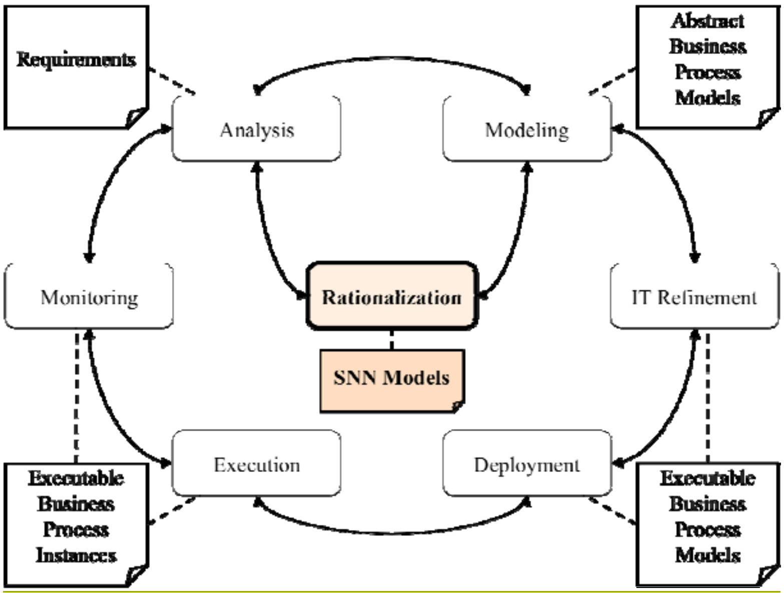

Figure 5: The Enhanced BPM Lifecycle (in highlight the additions with respect to the established BPM Lifecycle of Figure 4).

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.

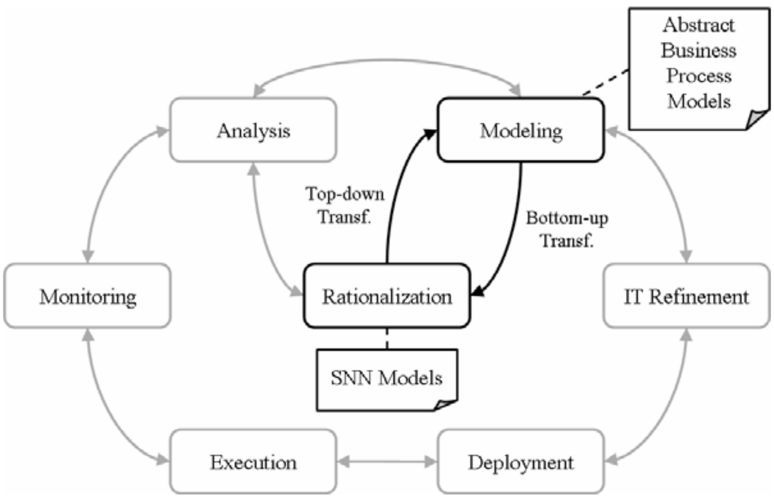

Figure 6. The role of Top-down and Bottom-up transformations in the Enhanced BPM Lifecycle.

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(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

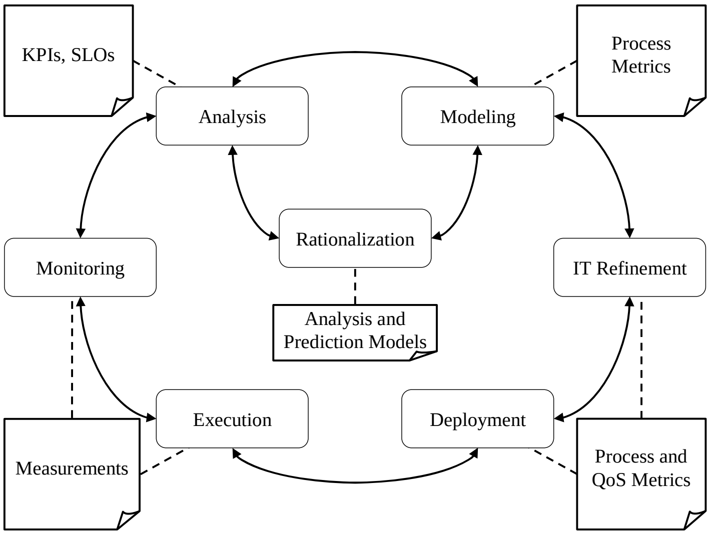

Figure 7: The Enhanced BPM Lifecycle in relation with the metrics used in the different phases.

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.

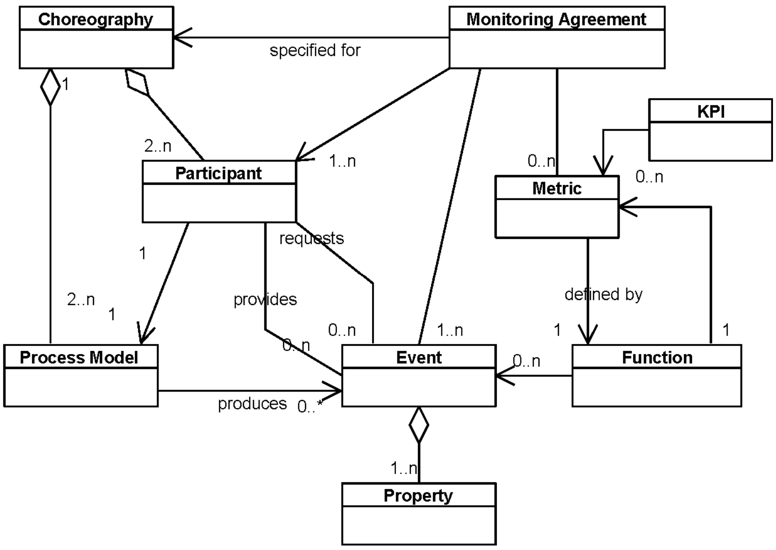

Figure 8: Main concepts of the Monitoring Agreement.

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.

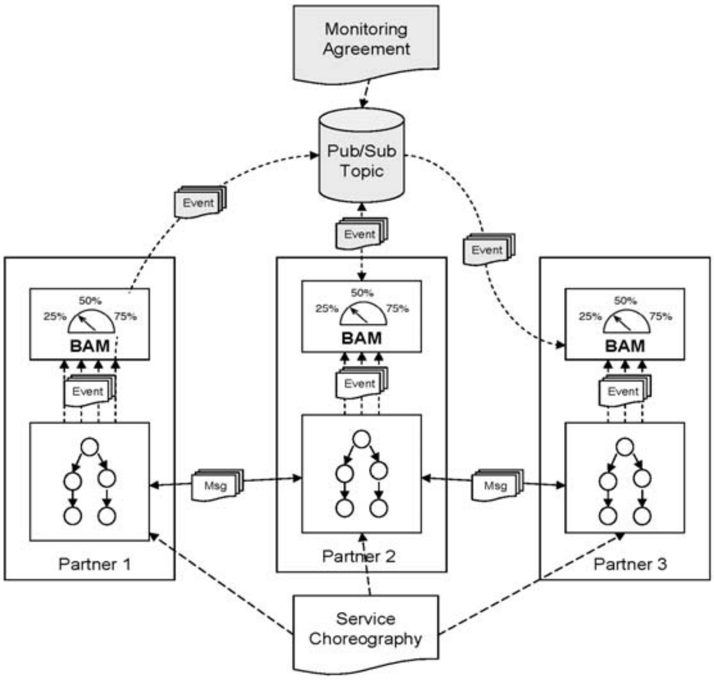

Figure 9: Monitoring Architecture. [3]

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 $\vec{x}_i = (x_{i1}, \ldots, x_{iK}), i = 1, \ldots n$ 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 $\vec{x}_i$ 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 $f_i(\vec{x}_i)$ 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.

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:

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

where $\vec{C} = (C_1, \ldots, C_K)$ 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:

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

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:

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

where $0 \le \gamma_i, \delta_i \le 1$ 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:

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

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:

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

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 n1 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:

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

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., Mancioppi, 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., 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)

[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)