Model Transformations to Leverage Service Networks*

Marina Bitsaki[1], Olha Danylevych[2], Willem-Jan A.M. van den Heuvel[3], George D. Koutras[1], Frank Leymann[2], Michele Mancioppi[3], Christos N. Nikolaou[1], and Mike P. Papazoglou[3]

[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,leymann}@iaas.uni-stuttgart.de
[3] INFOLAB, Dept. of Information Systems and Management, Tilburg University, The Netherlands
{wjheuvel,m..mancioppi,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, which provides 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 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 of service networks, assuming 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.

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) [2,3] leverage end-to-end service interactions between network partners that embody a succession of business processes typically cutting

* The research leading to these results has received funding from the European Community's Seventh Framework Programme under the Network of Excellence S-Cube - Grant Agreement n° 215483.

G. Feuerlicht and W. Lamersdorf (Eds.): ICSOC 2008, LNCS 5472, pp. 103–117, 2009.

across organizational boundaries and spanning various geographical locations. Service networks sequence service activities with the flow definitions of business process modelf 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) [4] 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, [2] 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. 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). This paper pursues new steps towards 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 [5]. Thus, 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) [6] and *Model Driven Architecture* (MDA) [7]. MDA is an effort of the Object Management Group providing the foundations and promoting the generation of programming code from models. Model transformations are used to generate new models (e.g. source code in some programming language) from other models (e.g. UML2 Class Diagrams) using repeatable (automatic) processes expressed as rules [8].

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, [9,10]). 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. Section 4 describes a staged transformation method to map SNs into business processes, 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

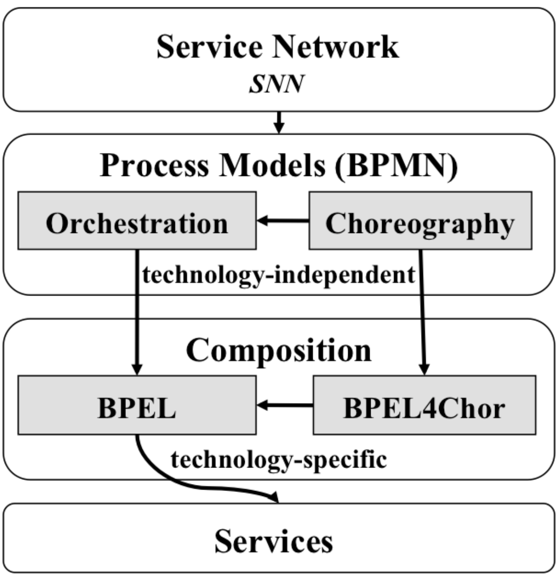

Fig. 1. The SN4BPM stack

The SN4BPM (Fig. 1) entails a stratified architecture that serves as the foundation for realizing business processes in service networks. This architecture unifies the BPM and SOC 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). 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) and *BPEL4Chor* [11], 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* [12].

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* [13]. For reasons of understandability we will now briefly explain the basic concepts in SNN.

A comprehensive overview of the SNN notation and its meta-model are provided in [2]. Fig. 2 depicts our running example as an SNN model. It focuses on two essential processes in fault resolution, the *Customer Fault Resolution* and the *Network Fault Resolution* business processes. The Customer Fault Resolution process conceptualizes the customers 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 Customers site to solve the issue. When the issue is solved, the Customer pays for the intervention.

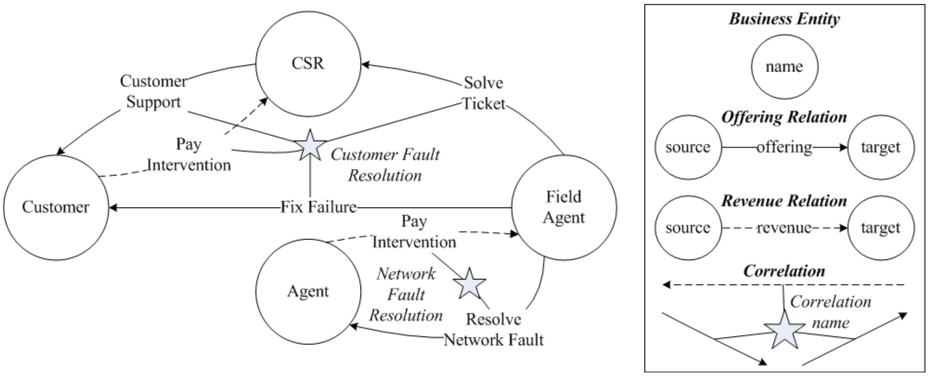

Fig. 2. An SNN model representing the eTOM example

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.

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 [2], which is depicted in Fig. 3.

This lifecycle extends the traditional BPM lifecycle by introducing a phase called *Rationalization* that deals with the design, optimization and simulation

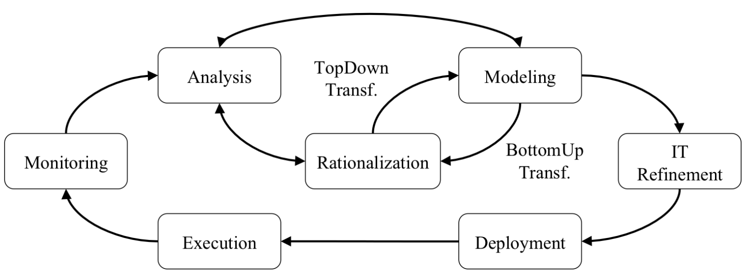

Fig. 3. The enhanced BPM lifecycle

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. 3. 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 transformation approach here proposed is an initial step towards this vision. The *TopDown* and *BottomUp* transformations, named after the relative directions in traversing the SN4BPM stack of Fig. 1, respectively allow to map service network models (belonging at the Service Networks layer) into abstract business process model (at Process Models layer in the stack) and vice-versa through a multi-step approach Mapping SNN models (belonging at the Service Networks layer) to abstract business process models (at Process Models layer in the stack).

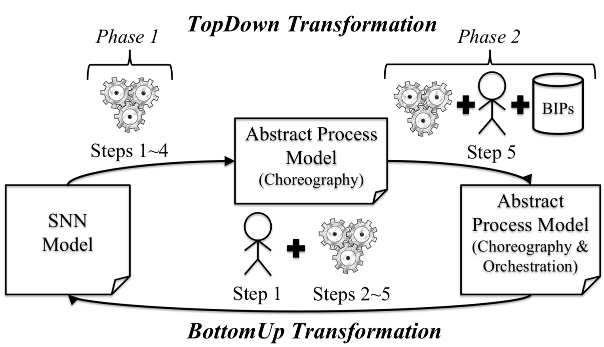

Fig. 4. Overview of the transformation approach

Fig. 4 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 fragment, workflows structuring the activities performed by the roles, and message-based interactions that occur among the different roles.

As depicted in this Fig. 4, 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 Section 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 BPMN[1] meta-models presented in Fig. 5 (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 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

[1] Extrapolated from the BPMN meta-model published on WSPER.ORG based on [14], and available at: `http://www.wsper.org/bpmn10.html`

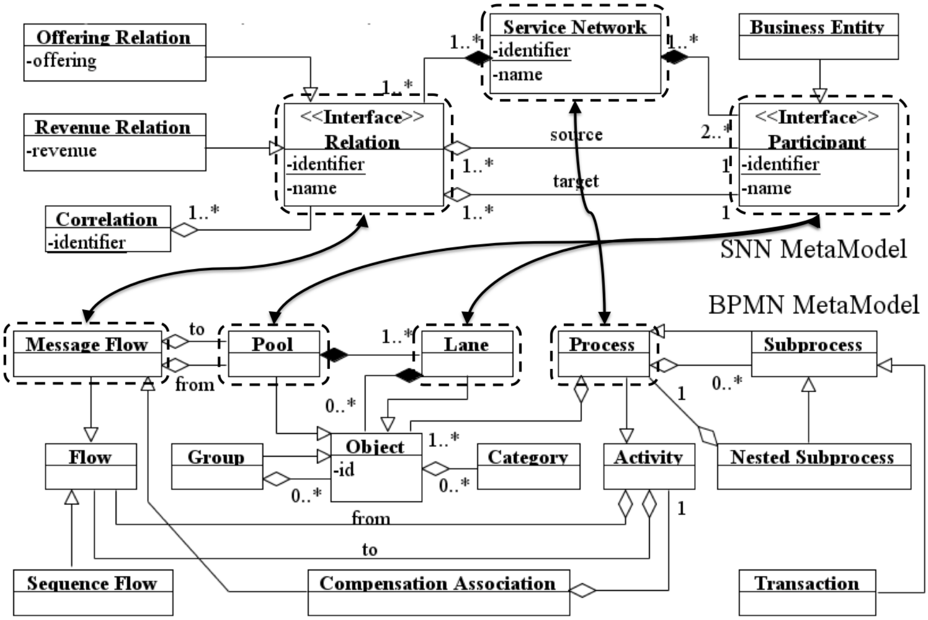

Fig. 5. The mappings between the SNN and BPMN meta-models

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.

These mapping 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 Top-Down transformation BPMN models that are immediately usable in a BPM lifecycle to, for instance, automate the generation of executable BPEL processes as proposed in [9]. 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) [15], 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 to this as a workflow. Workflows in different pools are logically interconnected via message-flows.

Fig. 6 presents an example of BIPs that we have developed, called "On Behalf Of" BIP. It 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 *CustomerFault-Resolution* 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 [3]).

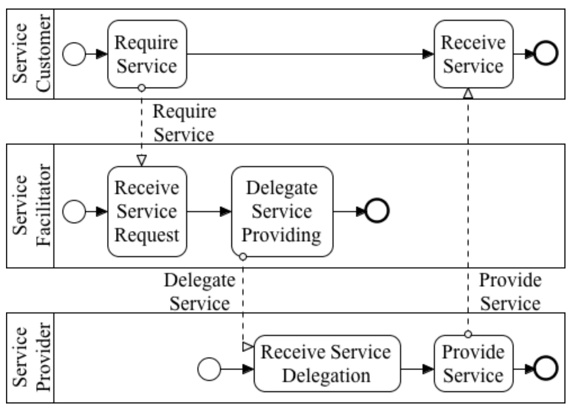

Fig. 6. The "On Behalf Of" Business Interaction Pattern

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. 4):

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. 7. 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 [14].
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 $\mathcal{A}$ with an activity in the pool of participant $\mathcal{B}$, do as follows:
   (a) 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 $\mathcal{A}$ to participant $\mathcal{B}$ and using the name of the message-flow as the revenue offering associated with the newly created revenue relation $\mathcal{R}$. Participant $\mathcal{A}$ and $\mathcal{B}$ are respectively source and target of $\mathcal{R}$.
   (b) If the message flow is labeled as an offering relation, then create a new offering relation in the SNN model connecting participant $\mathcal{A}$ to participant $\mathcal{B}$ and using the name of the message-flow as the offering associated with the newly created offering relation $\mathcal{O}$. Participant $\mathcal{A}$ and $\mathcal{B}$ are respectively source and target of $\mathcal{O}$.
   (c) If neither Step 4a nor Step 4b 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:
   (a) 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.

[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.

(b) 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.
(c) 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. 7 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. 7. Fig. 8 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. 8, Step 3). The revenue and offering relations in the SNN model are created in the fourth step (outcome presented in Fig. 8, step 4). Finally, the correlations are added to the SNN model during the fifth step (result in Fig. 8, 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).

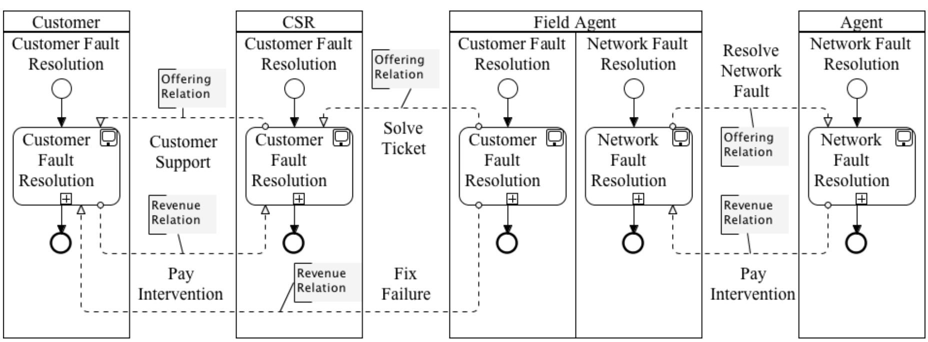

Fig. 7. A BPMN model based on the eTOM example presented in Section 3

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.

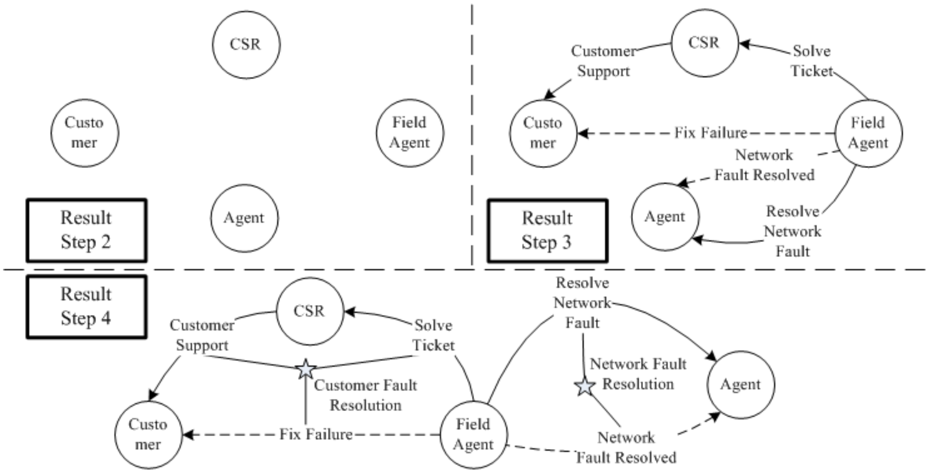

Fig. 8. The results of Step 3, Step 4 and Step 5 of the BottomUp transf. on the BPMN model in Fig. 7

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 participants 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 $\mathcal{R}$ in the correlation $\mathcal{C}$ connecting the source participant $\mathcal{A}$ and the target participant $\mathcal{B}$, create a message-flow from the sub-process in the lane $\mathcal{C}$ of the pool $\mathcal{A}$ to the sub-process in the lane named $\mathcal{C}$ of the pool $\mathcal{B}$. Let the revenue of the relation $\mathcal{R}$ be revenue. The newly created message-flow is labeled as: "[Revenue] revenue".
4. *Create offering message-flows:* for each offering relation $\mathcal{O}$ in the correlation $\mathcal{C}$ connecting the source participant $\mathcal{A}$ and the target participant $\mathcal{B}$, create a message-flow from the sub-process in the lane $\mathcal{C}$ of the pool $\mathcal{A}$ to the sub-process in the lane named $\mathcal{C}$ of the pool $\mathcal{B}$. Let the offering associated to the relation $\mathcal{O}$ 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:

1. *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 roles 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. 7 visualizes the result of applying the first four steps to the SNN model in Fig. 2. The Field Agent participant is transformed in the Field Agent pool (Step 1). Since 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 (Step 2). The revenue relation Resolve Network Fault in the SNN model is transformed into a message-flow, labeled "Resolve Network Fault"

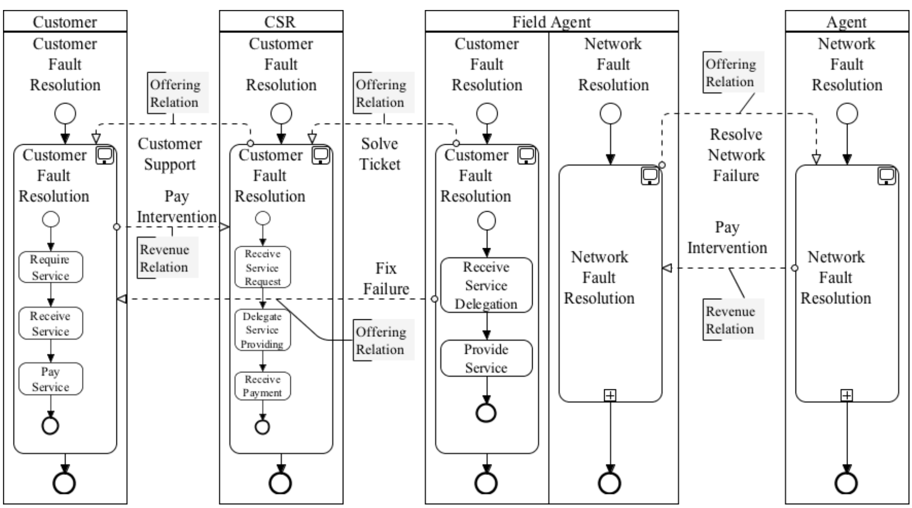

Fig. 9. Result of application of the "On Behalf Of" BIPs to the business process fragment "Customer Fault Resolution"

(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. 9 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. 7 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. 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. 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 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.

References

1. Papazoglou, M.P.: Service-oriented computing: Concepts, characteristics and directions. In: WISE, pp. 3–12. IEEE Computer Society, Los Alamitos (2003)
2. Bitsaki, M., Danylevych, O., van den Heuvel, W.J., Koutras, G., Leymann, F., Mancioppi, M., Nikolaou, C., Papazoglou, M.P.: An architecture for managing the lifecycle of business goals for partners in a service network. In: Mähönen, P., Pohl, K., Priol, T. (eds.) ServiceWave. LNCS, vol. 5377, pp. 196–207. Springer, Heidelberg (2008)
3. Caswell, N.S., Nikolaou, C.N., Sairamesh, J., Bitsaki, M., Koutras, G.D., Iacovidis, G.: Estimating value in service systems: a case study of a repair service system. IBM System Journal 47(1), 87–100 (2008)
4. McGovernan, D.: An introduction to BPM and BPMS. Business Integration Journal (April 2004)
5. Papazoglou, M.P., van den Heuvel, W.J.: Service oriented architectures: approaches, technologies and research issues. VLDB J. 16(3), 389–415 (2007)
6. Havey, M.: Essential Process Modeling. O'Really (2005)
7. Object Management Group (OMG): MDA Guide Version 1.0.1, Document Nr: omg/2003-06-01 (June 2003)
8. Sendall, S., Kozaczynski, W.: Model transformation: The heart and soul of modeldriven software development. IEEE Software 20(5), 42–45 (2003)
9. Ouyang, C., Dumas, M., ter Hofstede, A.H.M., van der Aalst, W.M.P.: From BPMN process models to BPEL web services. In: ICWS, pp. 285–292. IEEE Computer Society, Los Alamitos (2006)
10. Hornung, T., Koschmider, A., Mendling, J.: Integration of heterogeneous BPM schemas: The case of XPDL and BPEL. In: Boudjlida, N., Cheng, D., Guelfi, N. (eds.) CAiSE Forum. CEUR Workshop Proceedings, vol. 231 (2006) CEUR-WS.org.
11. Decker, G., Kopp, O., Leymann, F., Weske, M.: BPEL4Chor: Extending BPEL for modeling choreographies. In: ICWS, pp. 296–303. IEEE Computer Society, Los Alamitos (2007)
12. Chapell, D.A.: Enteprise Service Bus. O'Really (2004)
13. TeleManagement Forum: Enhanced Telecom Operations Map The Business Process Framework For The Information and Communications Services Industry, `http://www.tmforum.org`
14. Object Management Group (OMG): Business Process Modeling Notation Specification (February 2006)
15. RosettaNet Consortium: Partner Interaction Processes (PIPs), `http://www.rosettanet.org`