International Journal of Cooperative Information Systems

DISCOVERING PATTERNS FOR INTER-ORGANIZATIONAL BUSINESS PROCESS COLLABORATION

ALEX NORTA

Department of Computer Science, University of Helsinki,
P.O. Box 68 (Gustaf Hällströmin katu 2b),
Helsinki, FI-00014, Finland

PAUL GREFEN

Department of Information Systems, Faculty of Technology and Management,
Eindhoven University of Technology, P.O. Box 513, NL-5600 MB,
Eindhoven, The Netherlands

In the area of business-to-business (B2B) collaboration for manufacturing, original equipment manufacturers (OEMs) are confronted with the problem of spending considerable time and effort on coordinating suppliers across multiple tiers of their supply chain. In tightly integrated supply chains the failure of providing services and goods on time leads to interruptions of the overall production and subsequently results in customer dissatisfaction. This paper proposes the concept of electronic Sourcing *eSourcing* as a consolidating approach for improving the coordination of service provision across several tiers of a supply chain. eSourcing allows for the harmonization of heterogenous system environments of collaborating parties without requiring a total disclosure of internal business details to the counterpart. Furthermore, with tool support in eSourcing it is possible to verify the correct termination of processes and the contractual adherence of service provision without imposing fixed standardized routing. In this paper features of the eSourcing concept are analyzed in a pattern-based way. This paper pursues an analysis of eSourcing patterns in a top-down way for constructing collaboration configurations. The discovered and specified eSourcing construction patterns of this paper are instrumental in the EU-FP6 project CrossWork for the conduction of case studies with industry partners from the automobile industry.

1. Introduction

After exploring and successfully applying workflow concepts for intra-organizational applications [1,2], enterprises in the B2B domain are faced with the next challenge of achieving increased efficiency and effectiveness in the area of inter-organizational collaboration for commercial exchanges. Traditionally, a business transaction starts with negotiating and defining a contract that contains essential elements like the clear specification of a service that needs to be provided by one contracting party and how much compensation is offered by another contracting party that consumes this service. The parties involved in the contract have their own internal processes that need to be aligned inter-organizationally for the duration of the enactment of a business transaction. However, collaborating companies are confronted with many problems. For example, the business processes of the respective parties are supported by systems that are based on concepts that differ extensively and therefore inter-organizational collaboration is too cumbersome or not possible at all. One party may use a process-formulation language that contains constructs for which no comparable constructs exist in the domain of the collaborating counterpart. The situation may occur that processes terminate correctly in-house but deadlock when they are connected because of a data-flow or control-flow problem. Furthermore, setting up an inter-organizational business process collaboration is time-consuming and costly. Thus, it is desirable to have web-based middleware and WSCLs available for an automated alignment of inter-organizational processes so that the B2B transactions are harmonized in an effective and efficient way. This need for middleware and languages that support inter-organizational collaboration is under investigation by different research communities [3,4,5,6]. Summing up these efforts, the research direction is on the one hand focusing on exploring workflow concepts and on the other hand including service-oriented business integration (SOBI). These two aspects are united in the framework of dynamic inter-organizational business process management (DIBPM) [7] where the need of organizations is addressed for dynamically bringing together the processes of a service consumer and provider over web-based infrastructures. In this case *dynamically* means that organizations are found and matched just-in-time in an instance-based way. In the domain of SOBI, web service composition languages (WSCL) have emerged for supporting process specifications, e.g., BPEL, WSFL, XLANG, and so on [8,9,10]. However, these WSCLs are primarily suitable for choreographing the execution of composite services.

Since existing applications and languages support DIBPM only partially, the question arises how to explore DIBPM features without being overwhelmed by the complexities resulting from inherent business issues, concepts, and heterogenous technologies. Thus, it is essential to find an approach that results in a separation of concerns so that a structured investigation takes place. Intra-organizational workflow systems have been explored [11] in various perspectives by using a pattern-based analysis approach. Accordingly, this paper also proposes a pattern-based exploration for DIBPM. However, in the latter case the challenge occurs that such an exploration can not be carried out in a bottom-up way by analyzing inter-organizational applications and WSCLs that have proven themselves over many years and been adjusted to real-world problems. For DIBPM such "reality-proven" systems and WSCLs are not yet available. Therefore, this paper pursues a pattern-based exploration of DIBPM in a top-down way, which leads to a catalog of patterns that are for harmonizing the business processes of parties that intend to engage in a business collaboration.

The structure of the paper is as follows. First, Section 2 presents related work and Section 3 briefly introduces the concept of eSourcing. Next, control-flow properties of eSourcing are informally described in Section 4 followed by a presentation of eSourcing dimensions in Section 5 that are relevant for deducting patterns in a top-down way. The eSourcing-pattern catalog commences in Subsection 5.4 with a focus on the eSourcing characteristic called *contractual visibility*. After that Subsection 5.5 and Subsection 5.6 contain pattern specifications belonging to the eSourcing characteristics described by the newly introduced terms *monitorability* and *conjoinment* respectively. In Section 6 a carried-out case study from the automobile industry demonstrates how the eSourcing concept and patterns are applied, followed by a discussion in Section 7 that detects the scope for extending the eSourcing concept. Finally, Section 8 gives a conclusion for this paper.

2. Related Work

There are two groups of related work that are discussed in this section. Firstly, various approaches to inter-organizational process collaboration emerged in research projects and publications. Secondly, as this paper tries to explore patterns, the second group of related work presents existing pattern-catalogues for intra-organizational business process perspectives such as control-flow, data-flow, resource, and so on that are a valuable foundation for DIBPM.

2.1. *Approaches to Inter-Organizational Process Collaboration*

With respect to research projects, the WISE project [12,13] resulted in a software platform for process-based B2B electronic commerce that focuses on support for a network of small and medium-sized enterprises. WISE relies on a central workflow engine to control inter-organizational processes that are termed virtual business processes. In WISE a virtual business process consists of a number of black-box services that are linked in a workflow process [12]. A service is offered by an involved organization and can be a business process that is controlled by a local workflow management system. Although the WISE project succeeds in orchestrating workflows of different collaborating organizations, it does not cater for a flexible degree of mutual visibility of business-process details. Furthermore, collaborating parties can not negotiate how much may be observed during the enactment phase.

In the CrossFlow project [14] inter-organizational business process collaboration was investigated. In the context of this project, the formation of virtual enterprises is realized by dynamically out-sourcing a part of the service consumer's process to a service provider. A service matchmaker matches a service offering and a service request. The service provider has adjustment flexibility as nodes of the assigned process can be internally refined on a lower process level. Based on the electronic contract, a service enactment infrastructure [15] is established dynamically, employing workflow technology. CrossFlow has an external level that spans across organizational domains where the process specification is part of a contract specification. The workflow specification language of the workflow management system IBM MQSeries Workflow [16] forms the internal process level. The shortcoming of CrossFlow is that the service provider is not able to insert tasks that are not a lower process-level refinement, but that extend the business process on the highest level. At the same time it should be ensured with verification-tool support that a refinement by inserted tasks does not violate the service provision behaviour a consumer demands.

More recently, in the EU research project called CrossWork [17], the objective is pursued to develop automated mechanisms for allowing dynamic workflow formation and enactment, enabling tight collaboration and strong synergies between different organizations. The CrossWork architecture involves the development of ontologies for goal decomposition and team formation, followed by an inter-organizational business-process setup-, verification, and enactment-environment that integrates legacy systems. Furthermore, this architecture is complemented by visualization tools for the setup and enactment phase of an eSourcing configuration. During the course of the project the eSourcing [18,19] construction patterns presented in this paper are incorporated in an XML-based language termed eSML (*e*lectronic *S*ourcing *M*arkup *L*anguage) for formulating inter-organizational business process collaboration.

The integrated EU research project called ATHENA [20] investigates enterprise inter-operability in a holistic way with a technology-based approach that is guided by user requirements. It is the strategic objective of the project to enable networked businesses and governments. Expected results are of a technical-, business-, content-, and community-building nature where the topics of inter-organizational enterprise and process modelling, correspondingly required ontologies, service-composition frameworks, and enactment infrastructures are investigated. In the area of inter-organizational business process modelling, a business-level framework is proposed with a B2B process spanning across organizations that is complemented by a public process and a private process in the domains of a collaborating organization. This business level is complemented with a technical and execution level that is responsible for enactment. For business-process modelling an extension of event-process chain (EPC) formalism with so-called process-modules is proposed to achieve process abstraction. However, opting for EPC poses the difficulty that a verification of correct termination before enactment is not achieved. Furthermore, it is not possible to verify the extent to which collaborating parties adhere to their internal processes to what is externally promised.

Additional approaches to inter-organizational process collaboration exist. In [21] the topic of B2B integration is dealt with in detail, ranging from differing integration concepts via required integration technology and their deployment to a discussion about integration standards, products, and ongoing research in the domain. These integration approaches deal predominantly with the technicalities involved and do not propose a collaboration model that is suitable for the way B2B collaboration between an OEM and suppliers unfolds. An advanced approach for enabling inter-organizational business collaboration [22] investigates the use of public and private processes that solves the problems of message-exchange implementation, message transformation, and business rules handling between opposing parties. Open issues of [22] are how collaborating parties can regulate the degree of exposing their business internals to each other during setup and enactment time, which detailed options exist for binding the nodes of collaborating business processes, what issues arise for message exchanges between collaborating domains, which process properties must be adhered to for ensuring a smooth enactment phase, how can those properties be independently verified and evaluated during the setup phase without forcing the collaborating parties into exposing their business internals, and so on.

An alternative way of modelling business processes [23] incorporates characteristics of B2B collaboration by taking into account concepts from semiotics, the language/action perspective, and ontologies. Building on this approach, in [24] the high-level complexity of modelling supplier networks is addressed from a business perspective by establishing a functional decomposition diagram for supply-chain development that results in the deduction of a business-component model. In [25] a process is presented for identifying business components based on an enterprise ontology that satisfies well defined quality criteria. For the purpose of this paper, the described alternative approach is less applicable because approaches are available that provide a more formal basis on the aspect of control-flow modelling for the purpose of structurally matching inter-organizationally exposed business processes.

In [26] a Petri-net [27,28] based language is formally described for defining the interaction in a B2B environment. For this language several requirements are postulated that should be adhered to. It is particularly pointed out that current definition languages do not cater for ensuring the avoidance of deadlocks that can be verified before enactment and that they also lack formal execution semantics.

2.2. *Existent Pattern-Specification Catalogues*

Referencing patterns, Gamma et al. [29] first catalogued systematically some 23 design patterns that describe the smallest recurring interactions in object-oriented systems. Those patterns are formulated in a uniform specification template and grouped into categories. For the domain of intra-organizational business process collaboration patterns were discovered in various perspectives.

In the area of control flow, a set of patterns was generated [11,30,31] by investigating several intra-organizational workflow systems for commonalities. The resulting patterns are grouped into different categories. Basic patterns contain a sequence, basic splits and joins, and an exclusive split of parallel branches and their simple merge. Further patterns are grouped into the categories advanced branching and synchronization, structural patterns, patterns involving multiple instances, state-based patterns, and cancellation patterns. The resulting pattern catalog is instrumental for the evaluation [32,33] of WSCLs.

Following a similar approach as in the control-flow perspective, data-flow patterns [34] are grouped into various characteristics categories. One category focuses on different visibility levels of data elements by various components of a workflow system. The category called data interaction focuses on the way in which data is communicated between active elements within a workflow. Next, data-transfer patterns focus on the way data elements are transferred between workflow components and additionally describe mechanisms for passing data elements across the interfaces of workflow components. Patterns for data-based routing deal with the way data elements can influence the control-flow perspective.

Patterns for the resource perspective [35] are aligned to the lifecycle of a work item. A work item is created and either offered to a single or multiple resources. Alternatively, a work item can be allocated to a single resource before it is started. Once a work item is started it can be temporarily suspended by a system or it may fail. Eventually a work item completes. The transitions between those life-cycle stages of a work item either involve a workflow system or a resource. Characteristic categories for the resource perspective are deducted from those life-cycle transitions and group-specified patterns.

More recently, so-called service interaction patterns [36] are specified for the coordination of collaborating processes that are distributed in different, combined web services. Again, the patterns are categorized according to several dimensions. Based on the number of parties involved, an exchange between services is either bilateral or multilateral. The interaction between services is either of the nature single or multiple transmission. Finally, if the bilateral interaction between services is of the nature two ways, a round-trip interaction means the receiver of a response must be equal to the sender. Alternatively a routed interaction takes place.

After related work, the next section presents a specific concept for DIBPM that is the foundation for the pattern analysis of this paper, namely the concept of eSourcing [18,37]. In the CrossWork project, eSourcing is used as an integral concept for inter-organizationally harmonizing business processes.

3. The Concept of eSourcing

Structurally matching inter-organizational business processes in the framework of DIBPM requires a model for managing the involved business, conceptual, and technological complexity. A definition of DIBPM [7] is given as follows:

> *A dynamic inter-organizational business process is formed dynamically by the (automatic) integration of the subprocesses of the involved organizations. Here dynamically means that during process enactment collaborator organizations are found by searching business process market places and the subprocesses are integrated with the running process.*

Note that at least one organization involved in DIBPM must expose the explicit control-flow structure of its business process. Related issues to DIBPM are the definition and identification of processes, the way compatible business partners find each other efficiently, the dynamic establishment of inter-organizational processes, and the setup and coupling for process enactment. In order to manage such complex issues, a three-level framework [38] is a suitable model, which is explained in the next section.

Sourcing within the framework of DIBPM is essential for the automatic structural matching of parties who wish to collaborate. The following definition of eSourcing is used:

> *In the context of DIBPM, eSourcing is a framework for harmonizing on an external level the intra-organizational business processes of a service-consuming and one or many service-providing organizations into a B2B supply-chain collaboration. Important elements of eSourcing are the support of different visibility levels of corporate process details for the collaborating counterpart and flexible mechanisms for service monitoring and information exchange.*

The definition of eSourcing presented above establishes a differentiation to the research projects mentioned in Section 2 and takes into account requirements stemming from surveying the way industrial partners collaborate in the ongoing CrossWork [17] project.

The eSourcing definition includes the requirement of flexible service monitoring for which the newly introduced term *monitorability* is used (see Subsection 5.3). It is usually not economical or desirable for a consumer to monitor every step of service provision. Likewise, the service provider might not want all enactment progress to be monitored. The WISE project provides tools for evaluating the status of any process for monitoring that allows users to keep track and troubleshoot. Furthermore, WISE includes an awareness model [12] that allows the engine to make informed decisions about configuration, quality of service, resource reservation, load balancing, and so on.

In the CrossFlow project enactment shadowing is performed and explicit quality of service monitoring takes place. Shadowing means the shared view on the out-sourced process is equally reflected in the consumer workflow. Choosing what to monitor from a consumer perspective and what monitoring is permitted by a provider is negotiated and afterwards specified in an e-contract. However, it is not possible in CrossFlow to flexibly choose different mechanisms of service monitoring, e.g., based on messaging or polling mechanisms.

A crucial issue for collaborating business processes is the overall correct termination. Business processes that terminate correctly in isolation may, for example, contain a deadlock when linked together [26]. In B2B, if an inter-organizational business process collaboration fails, the consequence is penalty payments when a service is not provided as demanded. Such termination problems may also be caused by data-flow through an inter-organizational process. Neither the WISE nor the CrossFlow project offer the option of verifying correct termination before enactment of service provision. On the other hand, in the CrossWork project where eSourcing is used, inter-organizational collaboration needs to be put on a sound, formal foundation.

Summarized by the specifically introduced term *conjoinment* (see Subsection 5.3), it is the intention in eSourcing to evaluate data flow in an organization, which leads to the generation of control-flow for modelling a process. Before enactment this control-flow is verifiable for correct termination by using the tool Woflan [39]. Consequently, the data-flow perspective can be modelled in this initially created process so that it adheres to the correctly terminating control-flow. That way data-flow problems are reduced substantially. Thus, eSourcing needs to offer explicit control-flow constructs dedicated to data-flow across organizational domains.

The requirements above stress the importance of a sound control-flow foundation for the concept of eSourcing. Thus, the following section informally presents important control-flow properties so that a correct termination of an eSourcing configuration can subsequently be verified with tool support before enactment.

4. Control-flow Basis of eSourcing

The next subsection contains an eSourcing-configuration example based on which important control-flow properties for eSourcing are presented in the following subsection. A formal definition of the control-flow properties contained in the following sections is out of scope for this paper. Furthermore, since this paper focuses on eSourcing construction patterns, the example shows the end state of an established eSourcing configuration. The sequence of interactions between collaborating organizations during the setup phase are out of focus in this paper. However, in [18,40] so-called *interaction patterns* between a service consumer and a service provider are specified that focus on the setup phase of eSourcing configurations.

4.1. *An Example*

The example of Figure 1 is modelled using labelled Petri nets [27,28]. In Figure 1 the depicted eSourcing processes are distributed across three levels [38]. The very top and bottom show the internal levels of the service consumer and provider where processes are directly enactable by process management applications, e.g., by workflow management systems. Using internal levels caters towards a heterogenous system environment. In the middle of the service provider and consumer domains, processes are designed in a conceptual level independent from infrastructure and collaboration specifics.

In the centre of Figure 1, the external level stretches across the respective domains of eSourcing parties where process matching takes place. Parts of the respective conceptual-level processes are projected to the external level for performing matching to realize automated and dynamically forged collaboration between partners. A matching is successful when the projected processes of a service consumer and a service provider are equal, i.e., when similar nodes are embedded in identical control-flow constructs.

Furthermore, Figure 1 shows relevant parts of an eSourcing configuration. Starting with the domain of the service consumer in Figure 1, an in-house process is depicted on the conceptual level. The in-house process contains a subnet that is

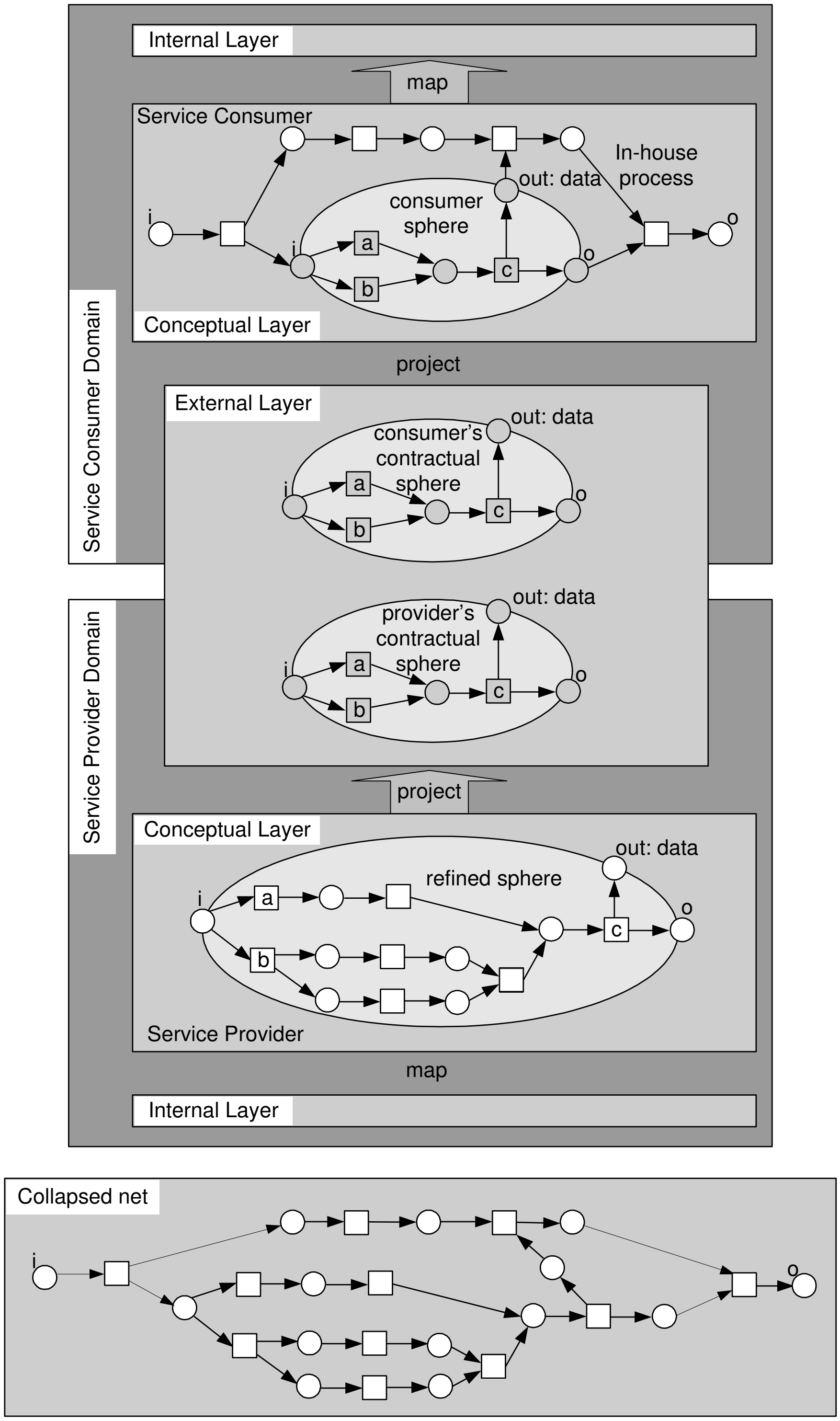

Fig. 1. An abstract eSourcing example in a three-level framework

termed a consumer sphere and visualized with a grey ellipse. On the border of the consumer sphere, labelled places are called interface places. Only one interface place is i-labelled and only one is o-labelled. The other interface places are either *in* or *out*-labelled to represent that exchange direction between the in-house process and its contained consumer sphere. Furthermore, the labelling implies whether an interface place has an input arc or an output arc in the sphere. If an interface place is i or *in*-labelled, it has one output arc to a transition in the sphere. If an interface place is o or *out*-labelled, it has one input arc from a transition in the spheres.

The in-house process is mapped to the internal level of Figure 1 where legacy systems are located. The consumer sphere is enacted by a different party and therefore projected to the external level to become the consumer's contractual sphere. From the opposite eSourcing domain a complementary provider's contractual sphere is projected to the external level. Since the respective contractual spheres in Figure 1 are equal, consensus is given between the eSourcing parties, which is the prerequisite for a contract. Note that this paper focuses on control-flow and abstracts from other relevant concepts [41] of electronic contracting.

Finally, the eSourcing configuration of Figure 1 includes one service consumer and one service provider. However, eSourcing may be extended for multi-party contracting where many providers offer their services to one consumer, which is demonstrated in the case study of Section 6. Such multi-party eSourcing is realized by nesting consumer spheres. Thus, a global consumer sphere contains nested consumer spheres that are embedded in control flow. The nested consumer spheres are filled with refined spheres from service providers The case study of Section 6 comprises one service consumer and several service providers.

4.2. *Control-Flow Properties of eSourcing*

With respect to control flow in eSourcing, a subclass of Petri nets is used, namely so-called *workflow nets* (WF-net) [42] that have been further explored [43,44]. Informally, a WF-net is a special subclass of Petri nets that has one unique input place (i) and one unique output place (o). There may be no 'dangling nodes', i.e., places and transitions that do not contribute to the processing of cases. Additionally, the requirement should be verified that for any case, the WF-net will eventually terminate and the moment the process terminates there is one token in output place o and all other places are empty. Moreover, it should be possible to execute an arbitrary transition by following the appropriate route through a workflow. The latter requirements are properties of *soundness* [45] and a strong argument for using WF-nets for eSourcing. Thus, WF-nets present an opportunity to verify the appealing notion of soundness of an overall process for ensuring smooth enactment, e.g. with the powerful tool Woflan [46].

Looking at the conceptual-level process of Figure 1, the in-house process with the contained consumer sphere is a sound WF-net. The consumer sphere of the service consumer in Figure 1 is a subnet contained in the in-house process. All nodes belonging to a consumer sphere are depicted as grey shaded when located as a subnet in the in-house process of a service consumer. A consumer sphere has an input place labelled i and an output place labelled o. All nodes belonging to the sphere are connected. When a consumer sphere is enabled, a token is put into the i-labelled input place produced from an active input node not belonging to the consumer sphere. After its enactment, only one token is left in its unique output place enabling one or many active nodes from outside of the consumer sphere belonging to the in-house process. Note that the previous statement focuses on control-flow in isolation and does not take other contexts of an eSourcing configuration into account.

Figure 1 shows an *out*-labelled interface place, which connects transitions that are located in a consumer sphere and the rest of the in-house process. The label specifies the nature of exchange between the in-house process and a consumer sphere. An *in*-labelled interface place denotes an exchange in the opposite direction. Exchange can only occur after a consumer sphere has begun with enactment being enabled by a token placed in its i-labelled input place. When a token is placed into an *in* or *out*-labelled interface place from an input transition, an exchange is attempted between the domains of a service consumer and provider.

Introducing these *in* or *out*-labelled interface places permits to set up the control-flow construct with which soundness can be verified before an actual exchange between eSourcing domains takes place during enactment. In this context, the term exchange is synonymous with data flow between a sphere and an in-house process. Enactment abnormalities like deadlocks may occur when data-flow deviates from process flow that is otherwise sound from a control-flow perspective. However, many data-flow problems occurring during enactment are avoidable when it is assumed that data ideally flows along the control flow of a process. That way data flow is emulated during build time with control-flow elements that include explicit control-flow constructs like *in*- and *out*-labelled interface places. A detailed investigation of data flow in eSourcing is not the focus of this paper.

A refined sphere of a service provider as depicted on the conceptual level of Figure 1 does not fulfill all properties required of a WF-net. As required, starting from the i-labelled input place the o-labelled output place is eventually reached once enactment is completed and all active nodes contribute to the processing of a case. However, the interface places of a refining sphere represent additional input and output places that are not permitted according to the definition of a WF-net. Thus, looking at the consumer's sphere in isolation, it represents a WF-net when the interface places are removed.

As the conceptual and external-level processes depicted in Figure 1 show, transitions depict labels. Comparing the consumer sphere and the refined sphere, one notes the labels of transitions are identically contained in both spheres. However, labels in a refined sphere must not always be in the consumer sphere and vice versa. A service provider is not aware of transition labels from the domain of a consumer that are not projected to the external level. A service consumer is not aware of labels contained in a refined sphere that are not projected to the external level.

4.3. *Quality Criteria of eSourcing*

Respecting control flow, the question arises how a service consumer ensures that a provider adheres to the contractual sphere in an eSourcing configuration while not imposing fixed routing in the provider's domain. Therefore a notion of *inheritance* is introduced that focuses on the dynamics rather than data and/or signatures of methods. Four notions of such dynamic inheritance have been identified in [47,48] addressing the usual aspects of substitutability, subclassing, and subtyping. Substitutability refers to replacing the superclass with a subclass without breaking the system. Subclassing asks if the subclass can use the implementation of the superclass. Finally, subtyping means a subclass can use or conform to the interface of the superclass. The four notions of inheritance are inspired by a mixture of these aspects. For eSourcing a restriction to so-called *projection inheritance* is considered of which the basic characterization can be described as follows:

> *If it is not possible to distinguish the behaviours of processes x and y when arbitrary active nodes of x are executed, but when only the effects of active nodes that are also present in y are considered, then x is a subclass of y.*

Thus, process x inherits the projection of the process definition y while process x conforms to the dynamic behaviour of its superclass by *hiding* active nodes new in x. Furthermore, such processes in an inheritance relation always have the same termination options. Mapped on the eSourcing configuration of Figure 1, hidden steps are transitions of a refined sphere where their labels are not contained in the corresponding contractual sphere. Let us assume that all the labels of active nodes of a consumer sphere are also contained in a refined sphere. When a consumer sphere is enacted and the service consumer only perceives the effects of active nodes with labels that are also contained in the consumer's sphere, the refined sphere is a subclass.

A second obligatory notion is the *well-directedness* of an eSourcing configuration. Looking at the eSourcing configuration of Figure 1, interface places carrying an *in* and *out* label that connect the consumer sphere with the rest of the in-house process are considered for this property. As mentioned before, they are part of what can be considered exchange channels between spheres and the remaining in-house process. If there exists one sending transition from a contractual sphere to an interface place, then there is one receiving transition located in the in-house process outside of the consumer sphere. Likewise, if there exists one sending transition from an in-house process outside of a sphere to an interface place, then there is one receiving transition located in a contractual sphere. Furthermore, if the sending transition is located in a consumer sphere, the refined sphere must equally have a sending transition to an equally labelled interface place. When a receiving transition is located in a consumer sphere, the refined sphere must also have a receiving transition to an equally labelled interface place.

4.4. *Collapsing an eSourcing configuration*

Finally, after presenting the notions of projection inheritance and well-directedness, the requirement of soundness [45] of an eSourcing configuration is covered. An evaluation is achieved by introducing the procedure of *collapsing*. Figure 1 shows the service consumer's and provider's conceptual-level processes at the top. A consumer sphere is contained in the in-house process of the service consumer. An interface place permits an exchange between the in-house process and the consumer sphere. Likewise, the corresponding refined sphere of the provider has an interface place where a sending transition serves as an input node. Thus, well-directedness of the consumer sphere and refined sphere are given. The collapsed net is shown at the bottom of Figure 1.

Compared to the top depiction, the bottom process shows that the consumer sphere is removed and replaced with the refined sphere in the in-house process. As a result, the collapsed net must be a sound WF-net. Independent of what type of soundness is adhered to, a collapsed eSourcing configuration must always be a sound WF-net. In the case of *eSourcing soundness*, the collapsed net containing the refined sphere must be a subclass net of the consumer's in-house process according to the projection inheritance requirement. *Basic soundness* is given when the collapsing procedure of an eSourcing instance results in a sound WF-net. However, in this case the collapsed net is not a projection-inheritance subclass of the consumer's in-house process.

Basic soundness of an eSourcing configuration is a required minimum that must always be adhered to with respect to control flow. Otherwise the enactment of a corresponding eSourcing configuration fails because of control-flow abnormalities, e.g., a contained deadlock. Basic soundness occurs, for example, when the contractual spheres only contain the same interface places as the consumer sphere but when no other nodes are contained. In that case the consumer does not mind how the service is provided as long as the interface places are dealt with correctly in the refined sphere.

On the other hand, when eSourcing soundness is given in a configuration, the full content of the consumer's conceptual-level sphere is projected to the external level. In the example of Figure 1 this situation is given. Accordingly, the provider must demonstrate projection-inheritance adherence in its refined sphere. Flexibility is given as the provider is permitted to add refinement nodes that do not violate projection inheritance.

After proposing the three-layer framework for complexity management and informally discussing the control-flow perspective in eSourcing, a top-down, pattern-based evaluation of eSourcing is performed. This top-down approach results in a multi-dimensional, logical space of dimensions that is instrumental for deducting eSourcing configuration-pattern specifications. Furthermore, the multi-dimensional, logical space creates a taxonomy for these patterns.

5. Exploring eSourcing Top-Down

As the previous section demonstrates, control-flow is one fundamental perspective of eSourcing. In this section other related perspectives are discussed. Informally, a perspective is a particular angle from which a certain domain is regarded.

5.1. *Relating DIBPM Perspectives*

Figure 2 relates eSourcing to other essential perspectives. To the very left and right of the figure, factory symbols represent a service consumer and provider where internal and conceptual-level processes are located. eSourcing rests on other relevant perspectives depicted as pillars in the centre of Figure 2 where external-level processes are located. The listed pillar perspectives are considered the most significant for DIBPM without claiming completeness.

The importance of the control-flow pillar is identified in the previous section. In an eSourcing configuration, data flow is essential as input and output of information during the enactment of service provision steps. Earlier, it has been stated that soundness can be verified with the tool Woflan [39]. By using control-flow elements that emulate data flow, this verification capability is exploited. As a result many enactment problems are preventable when data follows sound control flow. The provision and consumption of services involves human or non-human resources. Thus, the resource pillar deals with the way the involvement of such resources is defined. Finally, the enactment of an eSourcing configuration must offer a degree of certainty. By including a transaction pillar, enacted eSourcing steps are secured and exceptional situations are handled and compensated if required.

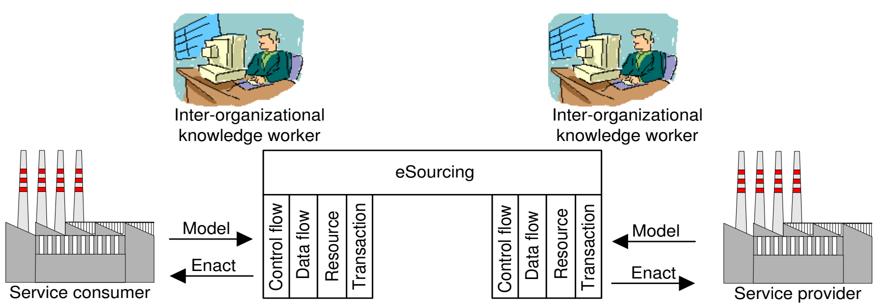

Fig. 2. Relating perspectives for DIBPM

By employing the perspective of eSourcing, business processes bridge organizational domains for realizing B2B collaboration. Developers of such eSourcing configurations must understand the concept well. In Figure 2 inter-organizational knowledge workers (IKW) are depicted who need to perform a harmonization of intra-organizational business processes. Such IKWs are likely to be repeatedly confronted with particular questions. How should a problem be solved and what are the pros and cons of a particular solution? Which solution should be chosen and how can the solution be realized? How does the selected solution relate to other potential solutions? Since such questions occur repetitively, IKWs should not have to consistently reinvent the solutions. Instead, developers of eSourcing configurations should have a set of patterns available that offer standard solutions to problems that keep reoccurring.

Patterns for the control-flow, data-flow, and resource perspective were discovered empirically after evaluating several workflow management systems and WSCLs. However, as the perspective of eSourcing is novel and no supporting systems exist yet, a special approach needs to be taken for discovering patterns. Thus, a particular pattern definition is proposed in line with Section 3 where the eSourcing definition is contained:

> *A pattern for eSourcing represents a technology-independent structure that exists across multiple applications and that has the purpose of offering a predefined, conceptual solution to recurring problems which inter-organizational business process developers are confronted with.*

To explore the eSourcing perspective in a pattern-based way, the following method is chosen. DIBPM contains several feature dimensions in the form of axes that create a multi-dimensional, logical space. On every axis, dimension values are located that detail the DIBPM feature an axis represents. By taking a subset of axes, a logical space is created that represents a particular DIBPM perspective. Consequently, the axes and their contained values serve as a taxonomy for ordering and relating to each other a set of perspective-relevant patterns. By pursuing a pattern-based approach for perspective exploration, an abstraction is achieved from technological and business complexity.

By evaluating the features used in a particular eSourcing scenario, it is possible to perform a positioning in the logical space. The following subsection creates such a logical space for the eSourcing perspective and gives a detailed description of dimension values that are located on the perspective axes.

5.2. *eSourcing Dimensions and Values*

As depicted in Figure 3, several characteristic dimensions of Sourcing exist in the form of axes that create a multi-dimensional, logical space. In line with the definition of Sourcing contained in Section 3 which is based on observations of industry partners in the CrossWork [17] project, the three dimensions of Figure 3 are contractual visibility, and the newly introduced terms monitorability, and conjoinment. On every axis further refining dimension values are located. The axis and their contained values serve as a taxonomy for discovering, ordering, and relating a set of patterns to each other. Furthermore, by evaluating the sort and amount of patterns used, it is possible to position an eSourcing configuration in the multi-dimensional, logical space. In Figure 3 the black circles on the walls of the cube stand for pattern instances that create eSourcing configurations. The connecting lines show the positioning of two concrete configurations where both configuration use four patterns belonging to particular axis values.

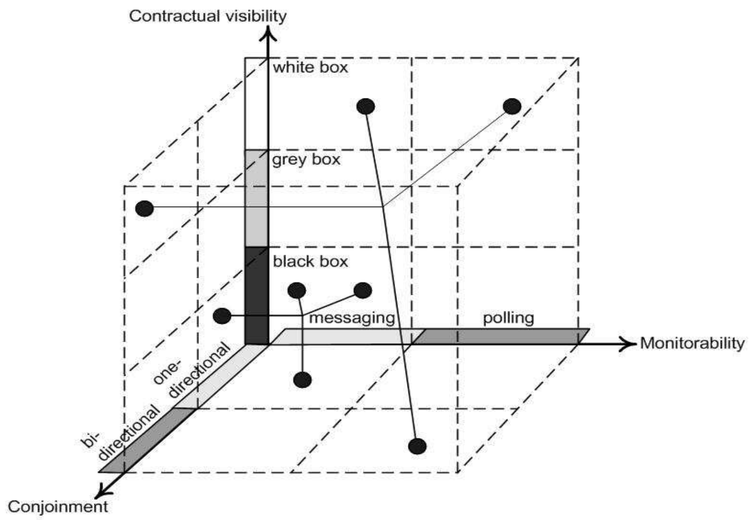

Fig. 3. Dimensions and values of the eSourcing perspective

The first dimension of *contractual visibility* addresses the issue of keeping business secrets by disclosing different amounts of internal process details to the collaborating counterpart. Thus, the depicted dimension values are called white box, grey box, and black box that are presented in Section 5.4. The *monitorability* dimension focuses on linking equivalent nodes of the consumer sphere and the refined sphere so that their enactment is coordinated and that it is possible for one party to observe in a flexible way the enactment progress of the eSourcing counterpart. The given values in Figure 3 are messaging and polling from which patterns are deducted in Section 5.5. Finally, the *conjoinment* dimension addresses the issue of modelling the exchange of commercially relevant information between the collaborating domains while ensuring correct termination. Such information exchange is either one-directional or bi-directional and Section 5.6 specifies patterns that are deducted from the given values.

5.3. *A Pattern Specification Template*

For specifying discovered eSourcing-construction patterns in the following sections, a description template is used containing a name, problem statement, pattern description, several forces, and one or several examples:

- The *name* is an identifier of a pattern that needs to be meaningful and representative for its main ideas.
- The *problem* of a pattern is a statement describing the context of pattern application. In this context, conflicting environmental objectives and their constraints are described. The application of a pattern in that context should result in an alignment of the given objectives.
- The *description* of a pattern mentions the inherent, differing pattern properties and describes the relationship between them.
- The *forces* describe obstacles that occur during pattern application and that may prevent an alignment of objectives of a pattern context.
- Finally, the *example* of a pattern either describes a concrete instance in a real-world setting where the pattern is used or an abstract application scenario.

Next, the Sections 5.4, 5.5, and 5.6 specify eSourcing-construction patterns deducted from the dimensions and values of Figure 3. These patterns are specified in the mentioned pattern-description template. Note, that for the sake of brevity not all eSourcing-construction patterns can be specified in this paper. Instead, only those construction patterns are specified that are used in the eSourcing case study of Section 6 eSourcing construction patterns. However, in [18,19] all patterns mentioned in Subsections 5.4, 5.5, and 5.6 are fully specified.

5.4. *Contractual Visibility*

The three-level model of Figure 1 shows two identical contractual spheres contained in the external level of an eSourcing configuration resulting from a consensus between service consumer and provider. The contractual visibility is variable depending on the amount of process content that is projected from the consumer sphere and the refined sphere to the contractual sphere. In Figure 3 three values are contained on the contractual-visibility dimension for which one corresponding pattern is deductable:

- *white-box pattern*: the consumer sphere and the contractual spheres contain identical nodes and control-flow constructs. Such an example is depicted in Figure 1.
- *black-box pattern*: only the interfaces of the consumer sphere are projected to the contractual sphere, which grants the service provider no visibility of further process details in the consumer sphere.
- *grey-box pattern*: the consumer sphere's interfaces and a subset of the remaining process content is projected to the contractual sphere.

The figure below depicts a grey-box pattern containing only two levels, namely the service provider's and consumer's conceptual level, and the external level. Since the consumer's and provider's contractual spheres always need to be similar for representing a consensus, depicting both contractual spheres does not add to the depiction. Furthermore, in the consumer's contractual level only the sphere is depicted and the rest of the in-house process omitted as the content of the in-house process has no influence on the type of contractual-visibility pattern. Finally, when a transition is depicted with a τ label [49] in a consumer sphere or a refined sphere, it means the eSourcing counterpart is not aware of this transition during the build time and the enactment of an eSourcing configuration. During build time the eSourcing counterpart is not aware because a τ-labelled transition is not projected to the external level and during enactment the counterpart is not aware since the effects of enacting a τ-labelled transitions are hidden from the counterpart.

In [18,19] the full specifications of Pattern 1 and Pattern 2, namely the *black-box pattern* and the *grey-box pattern* are contained. Only the grey-box pattern is specified in this section since it is used in the case study of Section 6.

Pattern 3 (Grey Box)

Problem: The service consumer does not mind in large parts how service provision is realized. However, the consumer wants to ensure that particular steps contained in the provided service are mandatory and carried out in certain control-flows.

Description: The pattern does not permit deviating interface places in the consumer sphere, the contractual sphere, and the refined sphere. Furthermore, a subset of nodes different from interface places contained in the consumer sphere is projected to the contractual sphere. As a result many nodes in the consumer sphere are therefore not visible to the service provider. The subset of nodes contained in the contractual sphere must be equally present in the refined sphere. To complete the demanded service provision, the provider fills the sphere adopted from the external level with additional nodes that are opaque for the service consumer. The resulting refined sphere must be a WF-net.

Forces: The service consumer can not demand total adherence to projection inheritance from the provider since only parts of the overall consumer sphere are disclosed.

Examples:

- An OEM in the automobile industry demands the delivery of leather-coated car seats from a provider. However, the OEM demands that the provider purchases the leather from a special certified seller following an agreed-upon flow of tasks.
- An example of grey-box contractual visibility is depicted in Figure 4. The contractual sphere depicts interface places with labels that are identical compared to the consumer sphere and the refined sphere. The contractual sphere shows two process places and two labelled, transitions that are equally contained in the consumer sphere and the refined sphere. However, the latter two spheres fill the gap depicted in the contractual sphere with differing nodes embedded in different control-flow constructs. Since transitions carry τ labels, the eSourcing counterparts are not aware how the

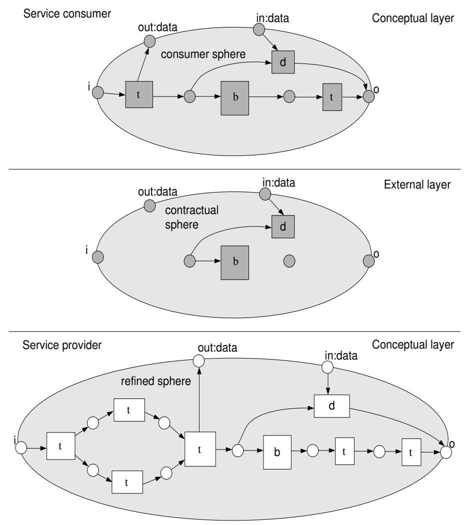

Fig. 4. Grey-box pattern example

sphere in the opposing domain is completed.

In the next section, patterns belonging to values of the dimension named monitorability are presented, which is a newly introduced eSourcing term. These monitorability patterns ensure flexibel degrees of enactment awareness for the service consumer.

5.5. *Monitorability*

Revisiting the initial eSourcing example of Figure 1, there is no linking depicted between the contractual sphere, refined sphere, and the contractual spheres of the respective eSourcing domains. However, during enactment of an eSourcing configuration, the collaborating parties must be able to coordinate and overview the service provision and consumption in a flexibel way. Thus, the monitorability patterns of this section offer ways of establishing such links between spheres in different domains.

Communication across organizational domains may take place in two ways. Accordingly, in Figure 3 two values are contained in the dimension called monitorability, namely polling and messaging. Based on those generalized communication concepts, patterns for eSourcing are deducted:

- In the case of value *polling* one eSourcing domain periodically asks if a change has taken place to a linked node of the opposing eSourcing domain. Detected changes are duplicated by the linked node in the polling eSourcing domain.

- The second monitorability classifier called *messaging* reverses the signalling direction. When a linked node experiences a change, a signal is sent to the other linked node in the opposing eSourcing domain.

Investigating such opposing monitorability patterns is relevant to cater to a heterogenous enactment environment that does not support similar monitorability options in the opposing eSourcing domains.

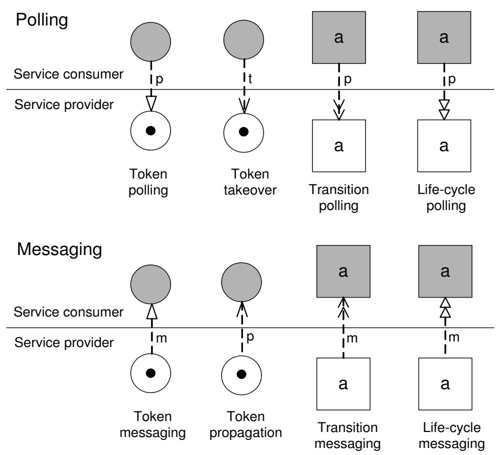

Fig. 5. Linking options for pursuing run-time visibility

The degree of monitorability for a service consumer during the enactment of an eSourcing configuration is determined by the level of contractual visibility (see Section 5.4). Only nodes from the consumer sphere and the refined sphere of an eSourcing configuration that are projected to the respective contractual spheres on the external level are considered for monitorability. Figure 5 depicts different messaging and polling patterns for linking such passive and active nodes without claiming completeness. However, before messaging-monitorability patterns are specified, an investigation of issues related to the monitorability of equally labelled active nodes with a life cycle is carried out.

Introduction of Life-Cycle Monitorablity

To apply patterns of life-cycle monitorability, it is necessary to explore whether it is possible to achieve a mapping of life-cycle stages that equally labelled transitions use in opposing eSourcing domains. To present an analysis of the problem in this subsection, a life cycle of a transition is refined to a labelled WF-net when an isolated transition has only one input place and one output place. As Figure 6 shows, there is only one life-cycle transition serving as an output node of that transition's input place and only one life-cycle transition serving as an input node of the transition's passive output node. Those life-cycle transitions propel a transition's life-cycle states, which are represented by labelled places.

If a transition in a process has multiple input places, they are at the same time multiple input places of the first labelled life-cycle transition that is starting to propel the life cycle. Equally, when the transition with a life cycle has multiple output places, they are at the same time the output places of the last unique, labelled life-cycle transition.

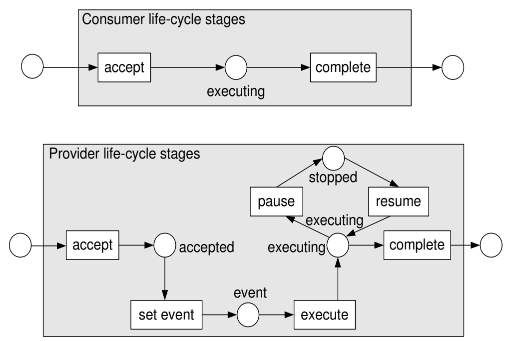

Fig. 6. Mapping of life-cycle stages

In Figure 6 the respective transitions' life cycles are WF-nets as both have one passive input and one passive output node. The transitions are depicted by grey shaded boxes. All other nodes are connected, i.e., they have at least one input and one output arc. In both cases the life-cycle nets terminate after enactment so that only one token is left in their output places. Furthermore, Figure 6 contains a loop for pausing and resuming a transition that is in the life-cycle stage *executing*. The transition contains an additional life-cycle transition and state for event setting, which is not present in the consumer's life cycle. When a transition is accepted by a provider, this may serve as an event that other transitions might need to wait for before they can be enabled.

Revisiting the notion of projection inheritance in Section 3, the life-cycle of the provider's isolated transition in Figure 6 is a subclass of the consumer's transition life cycle. Thus, in this case, when all transitions in a refined sphere and an in-house process are replaced with life-cycle transitions and life-cycle places then an in-house process containing a refined sphere instead of the consumer sphere is still a sound WF-net. Next, the *token-propagation pattern* is described in further detail as it is used in the case study of Section 6 that also uses the *token-messaging pattern*. However, for brevity the specification of the latter pattern is not given in this paper. Instead in [18,19], this pattern is fully specified together with the remaining monitorability patterns depicted in Figure 5.

Pattern 4 (Token Propagation)
Problem: During the enactment of an eSourcing configuration, tokens enter *out*-labelled interface places in the refined sphere that should trigger a message exchange with the in-house process. Since all places of the refined sphere must be empty after enactment, these tokens should be removed while the exchange between the refined sphere and the in-house process takes place. Additionally, the final token left in the o-labelled output place of the refined sphere needs to be removed to complete enactment.
Description: This pattern links two equally labelled places from spheres that are not part of the same level in an eSourcing configuration. Thus, linked are *out*- and o-labelled interface places of the refined sphere, contractual sphere, and the consumer sphere. The propagation starts when a token arrives in the linked source place. In that case the token is passed on as a message to the linked target place in the different level. As a result the source place has that token removed and placed in the target place.
Forces: A token already resides in the *out*-labelled interface place of the consumer sphere before token propagation takes place. Thus, this token may be ahead of the refined sphere's *out*-labelled interface place and as a result it still takes more time until a token-propagation message is triggered. Such an 'early' token can not result in enabling a transition.
Examples:

- For o-labelled interface places, the eSourcing configuration depicted in Figure 7 shows an example of an 'early' token occurring in combination with token propagation. A small e shows which transitions are enabled. The enactment state of Figure 7 shows a token in the consumer sphere's o-labelled interface place that is early. The reason is an additional transition with attached τ label in the refined sphere at the left bottom of Figure 7 that is enabled but has not fired yet. The transition of the in-house process that follows the consumer sphere's o-labelled interface place can not be enabled as the contained token is produced too early after the firing of the b-labelled transition. When the τ-labelled transition of the refined sphere completes firing, a token is placed in the output place. Consequently, the triggered token-propagation message results in having the token removed from the o-labelled interface place of the refined sphere. Next, the early token in the consumer sphere turns 'current' and enables the active output node of the in-house process.
- The *out*-labelled interface place of the refined sphere in Figure 7 is connected with a token-propagation arc. When a token is placed in the interface place, a message is sent to the equally labelled interface place of the consumer sphere. As a result the token is passed on across the eSourcing

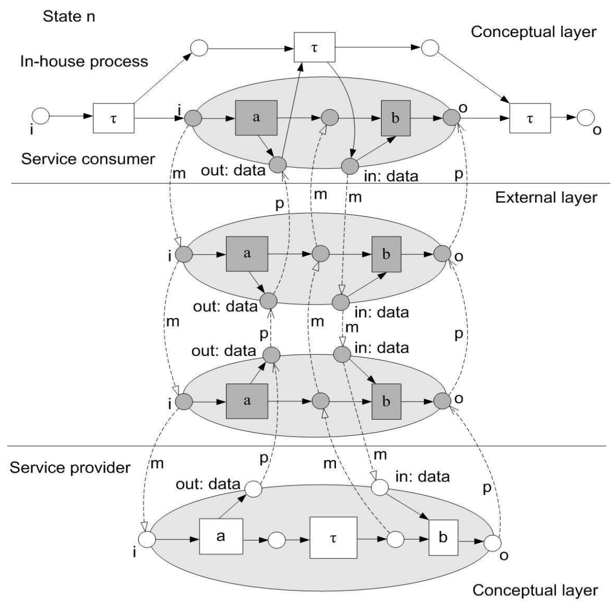

Fig. 7. Example of token propagation and token messaging

domains from one interface place to the other.

5.6. *Conjoinment*

Conjoinment focuses on the way a service provider and consumer establish exchange channels with each other. In the example of Figure 1, the refined sphere, respective contractual spheres, and the consumer sphere contain *out*-labelled interface places through which such conjoinment between the two eSourcing domains takes place.

In Figure 8 an extra notation for transitions is presented that is used for further discussing the conjoinment dimension. By using this notation, particular features of conjoinment patterns become apparent. The transitions of Figure 8 either receive a message, send a message, or receive and send a message consecutively. Such nodes can either be transitions that fire immediately when enabled or they contain more elaborate life cycles (see Subsection 5.5).

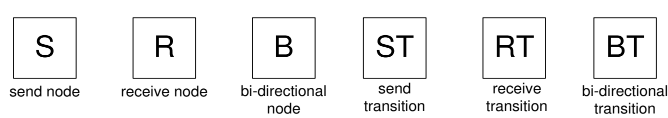

Fig. 8. Active conjoinment node notation

In order to make those distinctions visible, additional labels are depicted in Figure 8 that are carried by the shown nodes. The labels *S*, *R*, *B* are for transitions with lower level life cycles where the first is sending or initiating an exchange, the second receives or accepts an exchange, and the latter is bidirectional, i.e., the node needs to accept an exchange before it is enabled to initiate a counter exchange after firing. The labels *ST*, *RT*, *BT* are for transitions without any contained life cycle where the first is for an exchange-initiating transition, the second an exchange-accepting transition, and the latter node is a bi-directional transition that needs to accept an exchange before an exchange is initiated in response.

In accordance with the values depicted in Figure 3 on the conjoinment dimension, there are two values. In [18,19] all patterns deducted from the conjoinment-dimension values are fully specified.

- *One-directional* conjoining implies that there is one *out*- or *in*-labelled interface place present that is either complemented by a send transition or a receive transition respectively. Those send and receive transitions are contained in the spheres of an eSourcing configuration and handle the exchange between the domains of a service consumer and provider. There are two one-directional patterns termed *provider-initiated one-directional* and *consumer-initiated one-directional*.
- *Bi-directional* conjoining is initiated by a sending transition to the domain of the eSourcing counterpart that returns the communication exchange immediately to the initiating eSourcing party. There are two bi-directional patterns termed *provider-initiated bi-directional* and *consumer-initiated bi-directional*.

Below, the provider-initiated bi-directional pattern that is mentioned in the case study of Section 6 in a faulty application is presented. The pattern specification includes a visualization that consists of two parts. The top shows the conceptual-level in the consumer domain consisting of an in-house process with a contained consumer sphere. Due to the requirement of well-directedness (see Section 3) the conjoinment pattern of the top figures is replicated in the contractual spheres and the refined sphere. Thus, for the sake of brevity a depiction of the external level and the provider's conceptual level are omitted. If the labels of conjoinment nodes in the top depictions are in brackets, then the service provider is not aware of them as they are part of the in-house process. The same holds for transitions that carry a τ-label. The bottom of the pattern figures show a pattern variations that employ different types of conjoinment nodes from Figure 8. The specified conjoinment pattern of this subsection does not contain a description of related forces as they are similar for all conjoinment patterns. Instead the forces are given towards the end of this section. Finally, the *provider-initiated bi-directional* pattern is replaced in Subsection 6.3 with one-directional conjoinment patterns that are fully specified in [18,19].

<u>Pattern</u> 14 (Provider-Initiated Bi-Directional)
Problem: The service provider has to forward information to the consumer domain that should immediately be answered by a response. Such an exchange should not violate the correct termination of an eSourcing configuration.

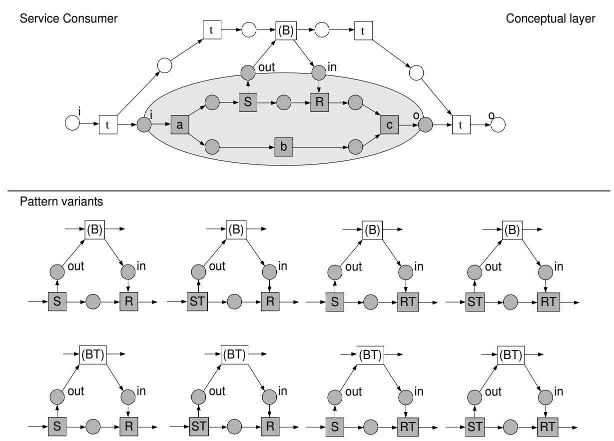

Fig. 9. Provider-initiated bi-directional conjoinment

Description: A sending conjoinment node that initiates information exchange to the domain of the service consumer is part of the consumer sphere, the contractual spheres, and the consumer. Such a sending conjoinment node is connected to a port at the border of the sphere through which information is exchanged to the domain of the service consumer. The port is connected to a bi-directional conjoinment node that is part of an in-house process outside of the consumer sphere. This bi-directional node receives the delivered information and responds with sending information back to the domain of the service provider through another sphere port. Finally, a receiving conjoinment node that is replicated in all spheres of an eSourcing configuration serves as a recipient for the information sent through the response port.
Examples:

- A local service provider is booking a hotel room according to special requirements of a customer. Those requirements are delivered by a travel agency that sources in the service of booking a hotel room. Since the payment format should be kept flexible, the provider must be able to ask the travel agency during enactment how the customer would like to pay for a particular hotel room. The response from the travel agency should be received immediately by the service provider in order to finalize the hotel booking.

- The top of Figure 9 the depicted consumer sphere contains a sending and receiving conjoinment node in the sphere. The first node carries an S label and the latter an R label. The sending conjoinment node has an output place that is an *out*-labelled interface place, which serves as an input node for the bi-directional node in the in-house process. This B-labelled conjoinment node produces an immediate response during the exchange. For this reason one output place of the bi-directional conjoinment node is an *in*-labelled interface place. Consequently, that interface place is an input place for the R-labelled receive node contained in the consumer sphere.

It needs to be stated that all four conjoinment patterns are exposed to the same forces. When a conjoinment construct is chosen, the performance characteristics of an exchange during enactment is influenced by incorporated monitorability patterns. Depending on the conjoinment direction, the *out*- and *in*-labelled interface places need to be linked with monitorability arcs. If messaging is applied, token propagation (see Pattern 4) is suitable and if polling is applied, token takeover (see Pattern 8 in [18,19]) is appropriate. However, such a monitorability choice influences the speed of conjoinment enactment. If token takeover is used, the defined polling period potentially postpones the conjoinment until the next polling interval starts. If token propagation is chosen, the conjoinment across eSourcing domains is started immediately. However, it is possible that the R-, RT-, or B-labelled conjoinment node is not yet enabled for responding to the conjoinment exchange.

Let us assume equally labelled conjoinment nodes are part of the consumer sphere, the respective contractual spheres, and the refined sphere. If these transitions are linked for life-cycle monitorability (see Pattern 7 and Pattern 11 in [18,19]) then conjoinment construction fails during build time if the lower level life-cycles can not be semantically matched. Furthermore, just as in the case of linking *out*- and *in*-labelled interface places, the speed of enacting conjoinment nodes varies depending on the type of employed monitorability links.

After presenting pattern specifications for the eSourcing dimensions contractual visibility, monitorability, and conjoinment, the following section presents a case study for eSourcing where the patterns are applied. The case study is input for the development of a proof-of-concept prototype in the EU research project called CrossWork [17], which is developed together with project partners from the automobile industry. Note, as this paper focuses on *eSourcing-construction patterns*, the example shows the end state of an established eSourcing configuration ready for enactment. The sequence of interactions between collaborating organizations during the setup phase are out of focus in this paper. However, in [18,40] so-called *interaction patterns* are specified that focus on the setup phase of eSourcing configurations.

6. Applying the eSourcing Patterns in a Case Study

In the automobile industry, OEM's have several tiers of suppliers that agree to deliver parts collaboratively. Such a constellation resembles a pyramid where the

OEM at the top spends considerable time and effort on aligning first- and second-tier suppliers for achieving the desired service provision. Thus, in the automobile industry there exists a need for a more efficient establishment of service provision across multiple tiers of supply chain.

In this case study, a truck-producing company is a service consumer that does not build all parts of the trucks by itself. Instead it has several suppliers of components that are united in a regional automobile cluster of service providers. Within that cluster the service provider A functions as a unique communication party for the entire cluster to the truck producer. Once an order for service provision is given to party A, a decomposition of the service takes place within the automobile cluster. Thus, with the agreement of the truck producer, several members of the cluster commit themselves to provide parts of the overall service. Henceforth, the suppliers of the automobile cluster are termed service providers.

The water tank itself consists of a water-tank body, a grommet, a motor pump, a dispenser, and a sealing ring. Provider A receives the order for the entire automobile cluster and organizes the distribution of the production of water-tank parts to partners of the automobile cluster. Thus, it is decided the water-tank body and the grommet are produced by service provider B, the motor pump and the sealing ring are produced by service provider C, and the dispenser is produced by service provider D. Furthermore, provider A takes over the tasks of initially preparing the order and assembling the finished parts into the water tank before it is delivered to the OEM.

After presenting the context of the case study, the remaining subsections are organized as follows. First, the case study is translated into an overall eSourcing configuration where the collaborating parties of the automobile cluster are organized into a virtual-enterprise network for the OEM. Next, for the sake of brevity only one sphere is chosen for a detailed depiction that shows how the subset of all eSourcing construction patterns specified in Subsections 5.4, 5.5, and 5.6 are applied to improve the design time and the analysis of an eSourcing configuration. This pattern application is split into two parts. First, it is demonstrated how patterns are instrumental IKWs during the design phase of eSourcing spheres followed by an analysis phase for which the eSourcing-construction patterns of this paper are equally instrumental. Finally, the technology employed in the proof-of-concept prototype for this water-tank case study is presented in the last subsection.

6.1. *The Overall eSourcing Configuration*

On the external level of Figure 10, several contractual spheres are depicted, which the consumer intends to source. From the consumer's perspective, first the order must be prepared before all respective parts of the water tank are produced in parallel branches. After the parallel branches are completed, the assembly takes place. The contractual spheres of the external level's virtual-enterprise network are translated to nested consumer spheres in one encapsulating consumer sphere. This

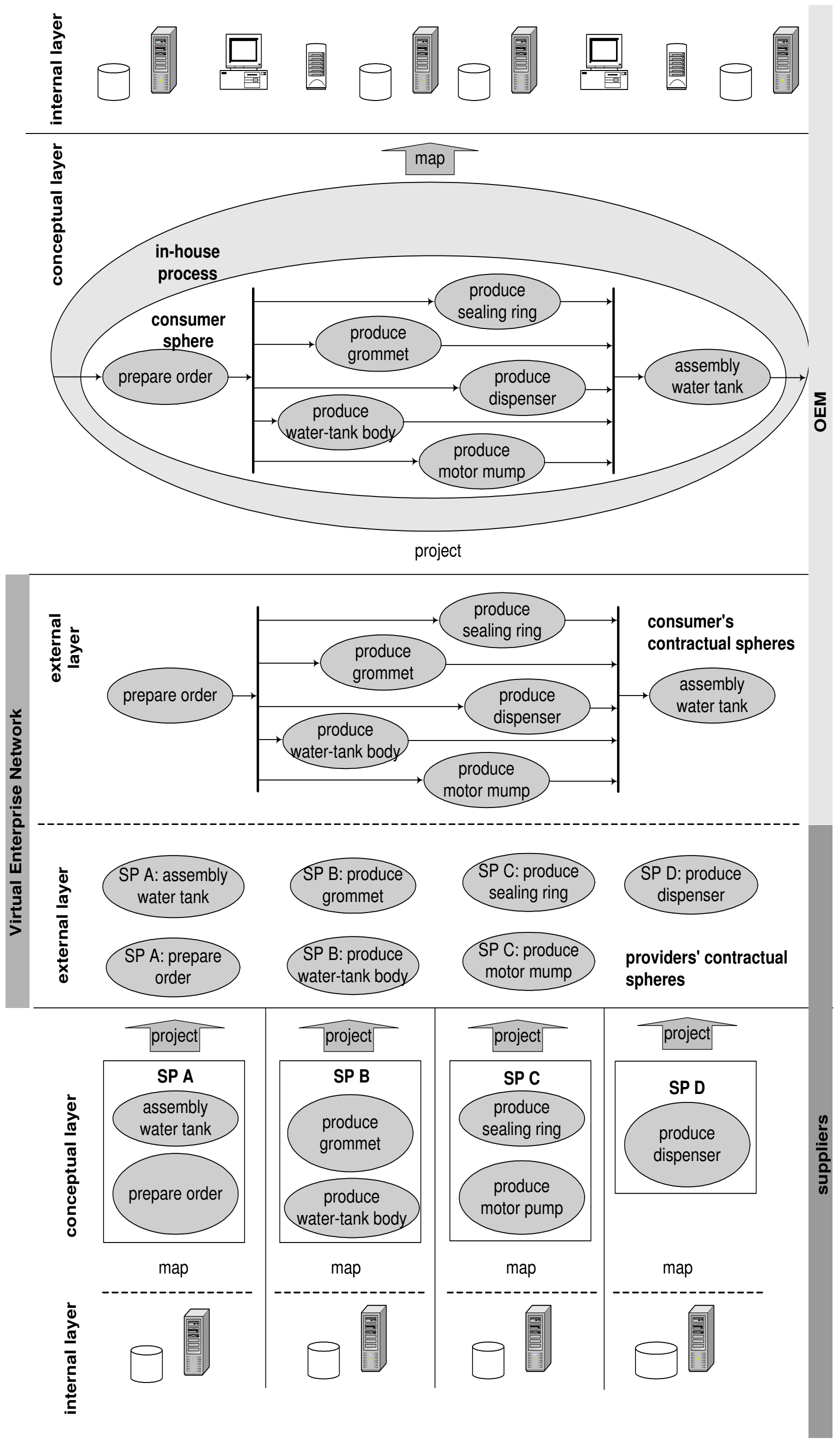

Fig. 10. The overall eSourcing configuration for eSourcing a water-tank production process

consumer sphere is embedded as a subnet of the in-house process on the truck-producer's conceptual level. Thus, the in-house process is designed for building the entire truck of which the consumer sphere is a demarcation that focuses on building the water tank.

The external level of Figure 10 also shows the providers' contractual spheres. In the grey ellipses, labels indicate which collaborating party takes over what service provision. In order to achieve consensus on the external level, it is required that the contractual spheres of the virtual-enterprise network and the providers' contractual spheres match in content, i.e. contain equal tasks embedded in the same control-flow structure. Furthermore, a consensus must exist about conjoinment and the extent of monitorability. The providers have refined spheres on the respective conceptual levels, which is indicated in Figure 10 by bigger sized ellipses than their respective contractual spheres on the external level. Finally, for the OEM and the service providers of the automobile cluster, Figure 10 depicts internal levels that contain icons representing legacy systems. In an eSourcing configuration such legacy systems may be extended with a service-oriented architecture so that parts of it are represented in wrapping services that are addressable by the conceptual-level processes.

For the sake of brevity it is not possible in this paper to offer a detailed presentation of all eSourcing spheres that Figure 10 depicts as grey shaded ellipses. Still, one subset of eSourcing spheres is chosen for a detailed study of pattern application. Concretely, the spheres of the conceptual and external levels that are related to the production of a motor pump are chosen for a deeper exploration of pattern application. To remain consistent with the modelling notation of the pattern specifications of this paper and point out the patterns used, the detailed spheres are depicted in a Petri-net format.
After providing an overall description of the water-tank eSourcing configuration, the following subsections describe how patterns are used for the design and analysis phases of an eSourcing configuration.

6.2. *Using Patterns in eSourcing-Configuration Design*

As stated before, one subset of related spheres depicted in Figure 10 is taken for discussing pattern application. The conceptual level of the service consumer in Figure 11 shows a consumer sphere that is a subnet of an in-house process. The depicted clouds mean the figure abstracts from details of the in-house process. There is only one node depicted that interacts with the ports of the consumer sphere. This interacting node is a bi-directional conjoinment node (see Pattern 14) that delivers a motor-pump specification to the *in*-labelled interface place of the consumer sphere after withdrawing such information from the web service of the internal level.

In the consumer sphere a receive task accepts the motor-pump specification. A receive task (see Figure 8) contains a lower level life cycle and requires resource involvement, e.g., for accepting and completing. Next, a task is contained for for-

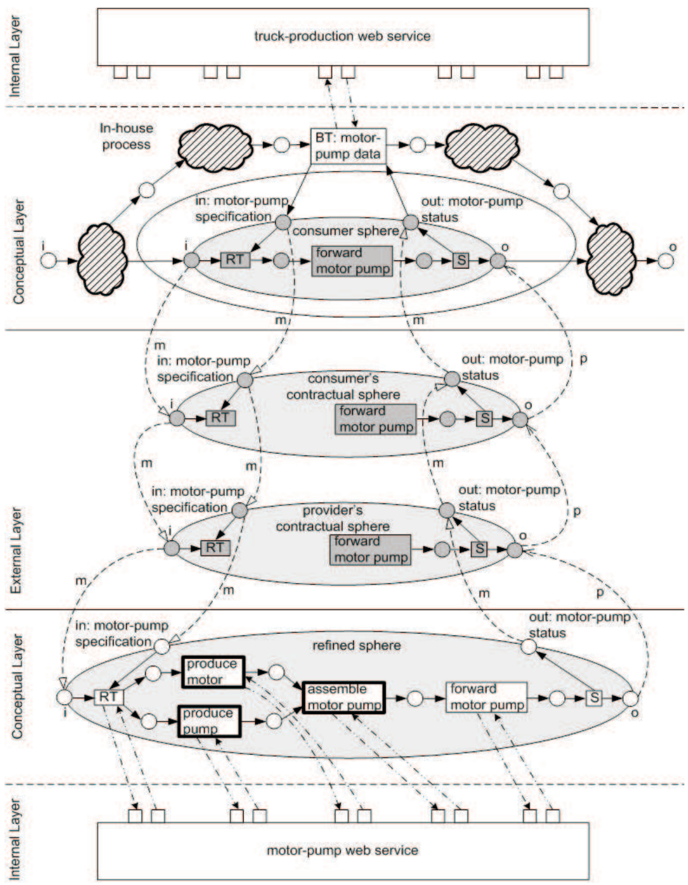

Fig. 11. Related eSourcing spheres in detail

warding the finished motor pump. That way the truck producer determines which forwarding company the service provider needs to use if it is assumed the consumer sphere exists before the refined sphere and the contractual spheres. Finally, a send transition returns the status of the motor-pump to the bi-directional conjoinment node via the *out*-labelled interface place. Note, a send transition (see Figure 8) does not involve resource involvement and it does not have a lower-level life cycle. Instead the node fires immediately when enabled. The service consumer chooses to use a bi-directional conjoinment pattern because the in-house process is not supposed to carry on when the returned quality data is not satisfactory. The consumer sphere only defines nodes for exchanging business information and for determining who needs to forward the produced motor pump. While the service consumer intends to project this process content to the external level, the consumer is aware that more process nodes are required for producing the motor pump. At the same time the consumer can indicate where such corresponding nodes may be inserted. Thus, a grey-box pattern (see Pattern 3) is applied where a subset of the consumer sphere is projected to the contractual sphere on the external level.

The service provider can fill the void in the consumer spheres in the refined sphere with the required additional motor-pump production steps. This scenario is depicted on the external level of Figure 11 where the consumer's contractual sphere shows only a subset of nodes contained in the consumer sphere. A void exists between the receive task for the motor-pump specification and the task that specifies which forwarding company the service provider needs to use. Furthermore, the external level of Figure 11 also shows the provider's contractual sphere that contains the same set of equally labelled nodes embedded in the same control-flow constructs. Thus, a consensus is depicted that may be preceded by a possibly complex negotiation procedure between the collaborating parties. An investigation of such negotiation procedures is not in the focus of this paper.

On the conceptual level of the service provider, the contractual sphere is refined by additional nodes. Since a grey box contractual-visibility pattern is used, the provider has freedom to complete the content of the contractual-sphere process in the refined sphere. Differently to a white-box contractual pattern (see Pattern 1 in [18,19]), the grey-box contractual pattern does not require adherence to projection inheritance (see Subsection 4.2). In the motor-pump spheres the OEM does not dictate in which way the production steps should be defined. Figure 11 shows boldly lined tasks that represent inserted nodes in the refined sphere. Thus, after receiving the motor-pump specification, the motor and the pump are produced in parallel branches before they are assembled in a joining task. Next, the ready motor pump is forwarded as defined by the truck producer and subsequently quality data is transferred in a document about the motor-pump status to the domain of the service consumer. The tasks focusing on producing and forwarding the motor-pump interact with a web service of the provider.

For transitively linking the refined sphere with the consumer sphere via the external level, two monitorability patterns are used, namely token messaging (see Pattern 5) and token propagation (see Pattern 4). In the water-tank case study it is assumed that selecting such monitorability patterns is determined by the heterogenous system environments that are used in the domains of the collaborating parties. Token messaging is used for connecting the i-labelled interface places. For an enactment application of the eSourcing configuration token messaging means once the enactment of the in-house process has reached the consumer sphere, such a state is messaged across the organizational domains and the enactment of the provider's refined sphere commences. Token messaging is also used for connecting *in*- and *out*-labelled interface places for exchanging the motor-pump specification and the status report across organizational domains. Finally, given the system infrastructure of the collaborating parties, token propagation is employed for connecting the o-labelled interface places of the refined sphere and the consumer sphere via the external level. In the eSourcing configuration of the water-tank case this means the enactment of the refined sphere is terminated and this event is communicated to the domain of the truck producer where the enactment of the next consumer sphere is starting.

6.3. *Using Patterns in eSourcing-Configuration Analysis*

The created eSourcing configuration of Figure 11 must be verified for correct termination before enactment. By doing so, the likelihood of penalty payments because of failed service provision is reduced. Thus, the parties independently submit their respective processes to a trusted third party. The consumer submits the entire in-house process and the providers submit their refined spheres without having to disclose internal business details to the collaborating counterpart. The trusted third party "collapses" the eSourcing configuration by replacing all nested consumer spheres of the in-house process with the refined spheres. Next, the correct termination of the collapsed net is verified. For the eSourcing configuration of Figure 11 this verification fails. The reason is a deadlock contained in the processes that is caused by the bi-directional conjoinment node. As this node needs to wait for the S-labelled conjoinment node to fire, it can never be enabled. In fact, revisiting the specification of Pattern 14 makes it apparent that a provider-initiated bi-directional conjoinment can not be applied for the nested consumer sphere of Figure 11 as the RT-labelled node is modelled before the S-labelled node.

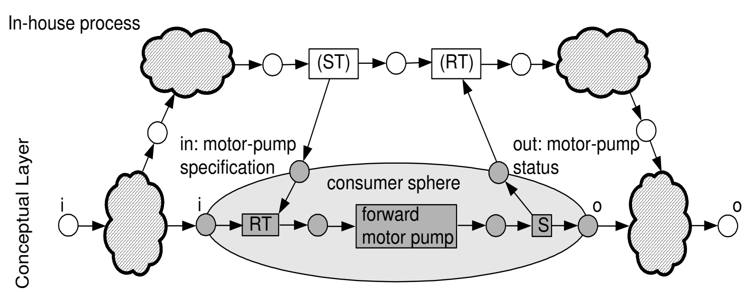

Fig. 12. Corrected in-house process

After the evaluation of correct termination, the trusted third party sends the deadlock information back to the service consumer who has to remodel the in-house process. The changed in-house process depicted in Figure 12, has the bi-directional conjoinment node replaced with a send task and a receive task. Since such conjoinment nodes require resource involvement, it is ensured that the in-house process can be halted if the quality feedback is not satisfactory. Consequently, a repeated termination verification by the trusted third party is successful, provided no deadlocks are caused by other parts of the eSourcing configuration.

The case study of this section that applies the concept of eSourcing and patterns specified in this paper, shows how inter-organizational business process collaboration can be realized. However, the case study raises the question of which scope exists for extending and enhancing the concept of eSourcing. The following section attempts to deal with this question.

7. Extending the eSourcing Concept

Based on the concept of eSourcing, scope for follow-up research is given. Other relevant perspectives for further research are data flow, resource, and exception and compensation handling. While patterns for an intra-organizational data-flow and resource perspective have been proposed [34,35], the concept of eSourcing is instrumental for exploring inter-organizational patterns for the data-flow and resource perspective on an inter-organizational level. The same is true for managing exception and compensation handling for eSourcing configurations that span across organizational boundaries.

Moreover, the utilization of eSourcing spheres in a three-level framework forms a basis for exploring exception handling and compensation, which is part of a transaction concept that needs to reflect e-business requirements. Since eSourcing is intended for commercial inter-organizational collaboration, it is essential to lay a foundation for a concept that not only supports a negotiation-process during the formation of an eSourcing configuration but also ensures enactment safety. By using spheres for providing and consuming services, it is feasible to equip an eSourcing configuration with an inter-organizational business-transaction concept. For example, a transaction model for inter-organizational business collaboration termed X-transaction [50] is adaptable for eSourcing. The use of spheres permits the introduction of necessary e-business specific atomicities [51], e.g., contractual atomicity, non-repudiation atomicity, payment atomicity, delivery atomicity. Furthermore, eSourcing can be integrated with the ongoing project called XTC [52,53] (e*X*ecution of *T*ransactional *C*ontracted e-services), where the objective is pursued to develop a business-transaction framework that utilizes abstract transactional constructs to provide a generic foundation for the support of complex transactional services in contract-driven inter-organizational business interactions that rely on dynamically composed web services.

Finally, while the conceptual sections of this paper focus on bi-lateral contracting, the case study demonstrates that the concept of eSourcing is sufficiently flexible to cater for multi-lateral contracting. Therefore, future research shall investigate the scenario of multiple eSourcing spheres in one service consumer's conceptual-level process that are projected in one connected process to the external level. Thus, each eSourcing sphere of the service consumer's contractual process is related to a separate provider sphere contained on the external level. Accordingly, several conceptual levels for each respective service provider contain the corresponding refined spheres of an eSourcing configuration. However, in future research such bi-lateral contracting requires adjustments with respect to the control-flow properties (see Subsection 4) of the service consumer's processes on the conceptual and external level.

8. Conclusion

For an exploration of eSourcing, this paper pursues a top-down method of construction-pattern discovery. eSourcing manages the conceptual, business, and technological complexity involved in inter-organizational business by using a three-level business process framework collaboration for commercial purposes. Furthermore, the paper informally presents control-flow properties of an eSourcing configuration that are a prerequisite for structurally matching processes of a service consumer and provider. These properties include provisions that allow a service provider to extend a business process with additional nodes that are opaque for the service consumer without violating the desired enactment behaviour of the overall service provision.

The characteristics of eSourcing are explored in a top-down way based on a multi-dimensional, logical space with the dimensions contractual visibility, monitorability, and conjoinment. Further refining dimension values create a taxonomy for eSourcing construction-pattern specifications that allows to deal with the inherent complexity of eSourcing in a technology independent way.

The case study shows that eSourcing is a suitable concept for matching service consuming and providing business processes while allowing each party to keep essential business-process parts hidden from each other. Tool support is available for verifying before enactment the correct termination of processes that are part of an eSourcing configuration. Furthermore, without disclosing process details of internally refined service provision, tool support is instrumental for assuring a consumer that a provider adheres to the agreed-upon service without imposing fixed, standardized routing on the latter party.

Based on the concept of eSourcing, scope for follow-up research is given. Firstly, the mentioned patterns for eSourcing are formulated conceptually and in a technology independent way. However, formulating patterns leaves room for ambiguity that is undesirable for the development of proof-of-concept prototypes. Thus, it is relevant to equip the patterns mentioned in this paper with clear semantics. Secondly, translating eSourcing into a proof-of-concept prototype in connection with the CrossWork [17] project promises to yield contributions in the domain of service-oriented computing, such as distributed application architecture and XML-based language development for eSourcing.

References

1. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.
2. F. Leymann and D. Roller. *Production Workflow: Concepts and Techniques*. Prentice-Hall PTR, Upper Saddle River, New Jersey, USA, 1999.
3. K. Baina, K. Benali, and C. Godart. Dynamic interconnection of heterogeneous workflow processes through services. In R. Meersman, Z. Tari, and D.C. Schmidt, editors, *OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003*, number 2888 in Lecture Notes in Computer Science, pages 444–461, Catania, Sicily, Italy, 2003. Springer Verlag, Berlin.
4. J. Krogstie and H.D. Jorgensen. Interactive models for supporting networked organizations. In A. Persson and J. Stirna, editors, *Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004*, number 3084 in Lecture Notes in Computer Science, pages 550–563, Riga, Latvia, 2004. Springer Verlag, Berlin.
5. A. Kumar, J. Wainer, and Z. Zhang. Meta-workflows and ESP: A framework for coordination, exception handling and adaptability in workflow systems. In C. Bussler, D. Fensel, M.E. Orlowska, and J. Yang, editors, *Web Services, E-Business, and the Semantic Web, Second International Workshop, WES 2003*, number 3095 in Lecture Notes in Computer Science, pages 13–27, Klagenfurt, Austria, 2004. Springer Verlag, Berlin.
6. J. Zdravkovic and P. Johanesson. Cooperation of processes through message level agreement. In A. Persson and J. Stirna, editors, *Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004*, number 3084 in Lecture Notes in Computer Science, pages 564–579, Riga, Latvia, 2004. Springer Verlag, Berlin.
7. P. Grefen. Service-Oriented Support for Dynamic Interorganizational Business Process Management. to appear, 2006.
8. F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana. *Business Process Execution Language for Web-Services 1.0*. http://www-106.ibm.com/developerworks/library/ws-bpel/, 2003.
9. IBM. *Web Service Flow Language (WSFL) 1.0 Specification*. http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf, 2003.
10. S. Thatte. *XLANG: Web Service for Business Process Design*, 2003.
11. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns Home Page. http://www.workflowpatterns.com.
12. G. Alonso, SU. Fiedler, C. Hagen, A. Lazcano, H.Schuldt, and N. Weiler. WISE: business to business e-commerce. In *Proceedings of the 9th International Workshop on Research Issues on Data Engineering*, pages 132–139, Sydney, Australia, 1999.
13. A. Lazcano, H. Schuldt, G. Alonso, and H Schek. WISE: Process Based E-Commerce. *IEEE Data Engineering Bulletin*, 24(1), 2001.
14. IBM Research. Crossflow architecture description, Technical report, ESPRIT Crossflow EP 28653, 1999.
15. Y. Hoffner, H. Ludwig, C. Gülcü, and P. Grefen. Architecture for Cross-Organizational Business Processes. In *Procs. 2^{nd} Int. Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pages 2–11, Milpitas, CA, USA, 20050.
16. IBM MQSeries workflow. http://www-4.ibm.com/software/mqseries/workflow.
17. CrossWork: Cross-Organisational Workflow Formation and Enactment, IST no. 507590. http://www.crosswork.info/.
18. A. Norta. *eSourcing: electronic Sourcing for business to business*. http://www.cs.helsinki.fi/u/anorta/research/eSourcing/, 2006.
19. A. Norta and P. Grefen. Discovering Patterns for Inter-Organizational Business Col-

laboration in a Top-Down Way. BETA Working Paper Series, WP 163, Eindhoven University of Technology, Eindhoven, 2006.
20. ATHENA: Advanced Technologies for Interoperability of Heterogenous Enterprise Networks and their Application. http://www.athena-ip.org/index.php.
21. C. Bussler, editor. *B2B Integration: Concepts and Architecture*. Springer, Berlin, 2002.
22. C. Bussler. THe Application of Workflow Technology in Semantic B2B Integration. *Distributed and Parallel Databases*, 12:163–191, 2002.
23. J.L.G. Dietz. *Enterprise Ontology - Theory and Methodology*. Springer-Verlag Berlin Heidelberg, 2006.
24. A. Albani, A. Keiblinger, K. Turowski, and C. Winnewisser. Dynamic Modelling of Strategic Supply Chains. In *E-Commerce and Web Technologies, 4th International Conference, EC-Web 2003, Proceedings*, volume 2738 of *Lecture Notes in Computer Science*, Prague, Czech Republic, September 2003.
25. A. Albani and J. Dietz. Identifying Business Components on the basis of an Enterprise Ontology. In *International Conference on Interoperability of Enterprise Software and Applications (INTEROP-ESA 2005)*, Geneva, Switzerland, February 2005.
26. J.M. Gomez, A. Haller, and C. Bussler. A conversation oriented language for B2B integration based on Semantic Web Services. Web Service Semantics. In *Towards Dynamic Business Integration Workshop. In conjunction with The 14th International World Wide Web Conference (WWW 2005)*, Chiba, Japan, May 2005.
27. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.
28. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets II: Applications*, volume 1492 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.
29. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Professional Computing Series. Addison Wesley, Reading, MA, USA, 1995.
30. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Advanced Workflow Patterns. In O. Etzion and P. Scheuermann, editors, *7th International Conference on Cooperative Information Systems (CoopIS 2000)*, volume 1901 of *Lecture Notes in Computer Science*, pages 18–29. Springer-Verlag, Berlin, 2000.
31. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(3):5–51, 2003.
32. W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, and P. Wohed. Pattern-Based Analysis of BPML (and WSCI). *QUT Technical report*, (FIT-TR-2002-05):487–531, 2002.
33. P. Wohed, W.M.P. van der Aalst, M. Dumas, and A.H.M. ter Hofstede. Analysis of Web Services Composition Languages: The Case of BPEL4WS. In I.Y. Song, S.W. Liddle, T.W. Ling, and P. Scheuermann, editors, *22nd International Conference on Conceptual Modeling (ER 2003)*, number 2813 in Lecture Notes in Computer Science, pages 200–215, Chicago, Illinois, 2003. Springer Verlag, Berlin.
34. N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Data Patterns. *QUT Technical report*, (FIT-TR-2004-01), 2004.
35. N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Resource Patterns. *BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven*, 2004.
36. A. Barros, M. Dumas, and A.H.M. ter Hofstede. Service interaction patterns. In W.M. P. van der Aalst and F. Curbera B. Benatallah, F. Casati, editors, *Business Process Management: 3rd International Conference, BPM 2005*, number 3649 in Lecture Notes in Computer Science, pages 302–318, Nancy, France, 2005. Springer Verlag, Berlin.

37. A. Norta and P. Grefen. A Framework for Specifying Sourcing Collaborations. In Jan Ljungberg and Bo Dahlbom, editors, 14^{th} *European Conference on Information Systems: Grand Challenges*, pages CD–ROM, Gothenburg, Sweden, 2006. IT-University Gothenburg.
38. P. Grefen, H. Ludwig, and S. Angelov. A Three-Level Framework for Process and Data Management of Complex E-Services. *International Journal of Cooperative Information Systems*, 12(4):487–531, 2003.
39. H.M.W. Verbeek, T. Basten, and W.M.P. van der Aalst. Diagnosing Workflow Processes Using Woflan. *The Computer Journal, British Computer Society*, 44(4):246–279, 2001.
40. A. Norta and P. Grefen. Developing a Reference Architecture for Inter-Organizational Business Collaboration Setup Systems. BETA Working Paper Series, WP 170, Eindhoven University of Technology, Eindhoven, 2006.
41. S. Angelov and P. Grefen. The 4W framework for B2B e-contracting. *International Journal of Networking and Virtual Organizations*, 2(1):78–97, 2003.
42. W.M.P. van der Aalst. Verification of Workflow Nets. In P. Azéma and G. Balbo, editors, *Application and Theory of Petri Nets 1997*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426. Springer-Verlag, Berlin, 1997.
43. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
44. C.A. Ellis and G.J. Nutt. Modelling and Enactment of Workflow Systems. In M. Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993*, volume 691 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, Berlin, 1993.
45. W.M.P. van der Aalst. Structural Characterizations of Sound Workflow Nets. Computing Science Reports 96/23, Eindhoven University of Technology, Eindhoven, 1996.
46. H.M.W. Verbeek, T. Basten, and W.M.P. van der Aalst. Diagnosing Workflow Processes using Woflan. *The Computer Journal*, 44(4):246–279, 2001.
47. W.M.P. van der Aalst and T. Basten. Inheritance of Workflows: An Approach to Tackling Problems Related to Change. Computing Science Reports 99/06, Eindhoven University of Technology, Eindhoven, 1999.
48. T. Basten. *In Terms of Nets: System Design with Petri Nets and Process Algebra*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, December 1998.
49. W.M.P. van der Aalst and T. Basten. Inheritance of workflows: An approach to tackling problems related to change. *Theoretical Computer Science*, 270(1-2):125–203, 2002.
50. P. Grefen, K. Aberer, Y. Hoffner, and H. Ludwig. CrossFlow: Cross-organizational Workflow Management in Dynamic Virtual Enterprises. *International Journal of Computer Systems, Science, and Engineering*, 15(5):277–290, 2001.
51. M.P. Papazoglou. Web Services and Business Transactions. *World Wide Web: Internet and Web Information Systems*, 6:49–91, 2003.
52. B. Kratz, T. Wang, J. Vonk, and P. Grefen. Flexible Business Transaction Composition in Service-Oriented Environments. BETA Working Paper Series, WP 154, Eindhoven University of Technology, Eindhoven, 2005.
53. T. Wang. Towards a transaction framework for contract-driven service-oriented business processes. In A. Hanemann, editor, *Proceedings of the IBM PhD Student Symposium at 3rd International Conference on Service Oriented Computing (ICSOC2005)*, pages 43–48, 2005.