

2006

A framework for specifying sourcing collaborations

Alex Nortá

Eindhoven University of Technology, a.norta@tm.tue.nl

Paul Grefen

Eindhoven University of Technology

Follow this and additional works at: <http://aisel.aisnet.org/ecis2006>

Recommended Citation

Norta, Alex and Grefen, Paul, "A framework for specifying sourcing collaborations" (2006). *ECIS 2006 Proceedings*. 208.
<http://aisel.aisnet.org/ecis2006/208>

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in ECIS 2006 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact elibrary@aisnet.org.

A Framework for Specifying Sourcing Collaborations

Alex Nort a and Paul Grefen

Eindhoven University of Technology, Faculty of Technology and Management, Department of Information Systems, P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands. a.nort a@tm.tue.nl

Abstract. Dynamic inter-organizational business process management (DIBPM) combines service-oriented business integration (SOBI) and workflow management as a promising approach for supporting commercial business-to-business (B2B) activities over web-based infrastructures. SOBI applies concepts from the field of service-oriented computing in the domain of dynamic business collaboration. Currently, SOBI technologies insufficiently support B2B collaboration where dynamic matching of structures of service consuming and service providing processes is performed. Collaborating parties want to control how much process detail they expose and which parts of them are monitorable. SOBI technology should offer rigor that permits verification of desirable features before enactment, e.g., correct termination. Furthermore, current SOBI technologies lack concepts which are useful for specifying and implementing B2B collaborations. Hence, several related critical issues are explored in this paper. Firstly, how to manage the inherent conceptual, business, and technological complexity of such business collaboration. Secondly, the issue is addressed of laying a foundation for rigor that is instrumental for verifying control-flow adherence and correct termination of coupled business processes. These requirements need to be guiding for the development of specification languages of inter-organizational business processes and related middleware that enact them in a web-based way. Exploring these critical issues leads to the proposal of *Sourcing* that employs a three-level framework for tackling the complexity of dynamically matching a service consuming and a service providing process. Furthermore, Sourcing offers rigor by utilizing well explored process theory that results in improved control over inter-organizational business process structure. Finally, the issue of suitability is tackled by discovering inherent Sourcing features that permit the positioning of Sourcing configurations in differing perspectives. Those values are instrumental for subsequently discovering patterns that are translatable into a language for full-fledged DIBPM support.

1 Introduction

The way companies conduct their business activities is experiencing significant changes. Employing information infrastructures in novel ways increases the pace of B2B collaboration, e.g., of outsourcing which is needed because companies focus on their core competencies or miss the know-how to perform certain business activities. A promising approach for B2B is the coupling of workflow concepts with service-oriented business integration. This emerging framework of DIBPM [17] offers a new model for addressing the need of organizations for dynamically bringing together a service consumer and a service provider over web-based infrastructures where the service is a business process. In this paper the term *dynamic* refers to automatically integrated business relations that are forged between business parties by matching structures of respective processes.

The setup of such B2B commerce is a client-server relationship where one party offers a service that is integrated into the process of a consumer. Thus, parties that wish to engage dynamically disclose process details to each other while they keep many details hidden to safeguard their competitive advantages. Though a service provider has to adhere to the contractual requirements agreed with the service consumer, the provider still needs flexibility for extending and adjusting the service provisioning to internal needs that remain opaque to the consumer, e.g., to perform back-office tasks. Still, processes of respective parties may be correct internally but fail to terminate inter-organizationally. For example, one party expects payment before shipping while the other party expects shipping before payment.

A language that permits process specification for dynamic structural matching in DIBPM is required. In the domain of SOBI, web service composition languages (WSCL) have emerged for supporting process specifications, e.g., BPEL, BPML and so on [14, 15]. Such languages compose services in a workflow, offering a complex service that carries out activities. However, WSCLs merely choreograph the execution of composite services that remain a black box to each other. Thus, the existing WSCLs are not suited for a client-server type of process-oriented B2B collaboration. Generally, these languages are semantically ambiguous and therefore lack expressive rigor for

verifying correct termination in advance of enactment. Current WSCLs are to a limited degree [11, 30] suitable for formulating business processes as they need to pay attention to an application environment of business, conceptual, and technological complexity.

Attempts are under way to create a WSCL and related enactment applications [1] that tackle the issues of complexity, suitability, and formal rigor. However, an integrating concept is missing that exploits existing research results. This paper tries to fill the gap of a suitable concept for DIBPM. *Sourcing* is proposed building on the idea of having a part of the overall business process of a service consumer performed by a service provider. Sourcing is a general paradigm allowing for flexible ways of commercial collaboration. To handle the inherent complexity, a three-level framework [20] is adopted. Furthermore, the concept of Sourcing is rigorous as it builds on Petri-net theory [24, 25]. Thus, in Sourcing the resulting configuration can be collapsed into a *workflow nets* (WF-net) [16] and checked before enactment for the notion of soundness, i.e., if the control flow of a Sourcing instance with a matched service-consumer and provider process terminates successfully.

With respect to the issue of suitability and expressiveness, a top-down exploration approach is chosen for discovering and exploring relevant characteristics of Sourcing. Various degrees of monitorability are resulting from differing ways of linking nodes of the consumer process and provider process. Conjoinment constructs in Sourcing lay the foundation for inter-organizational data exchange. Such constructs are paid attention to for checking the violation of soundness. Finally, different degrees of contractual visibility are discovered that permit collaborating parties to keep certain business activities hidden from each other. These characteristics are requirements and are subsequently instrumental for patterns discovery and specification. Those patterns are translatable into suitable WSCL-languages and enactment applications. Note that such pattern discovery is not the focus for this paper.

The structure of this paper is as follows. First, the context of Sourcing is given in Section 2 where the use of a three-level model is proposed, followed by a definition of the nature of Sourcing in Section 3. After presenting an exemplary Sourcing configuration in Subsection 3.1, the control-flow properties of Sourcing are defined in Subsection 3.2 and quality criteria of Sourcing are discussed in Subsection 3.3. Next, the Sourcing perspective is positioned and investigated in correlation to DIBPM in Section 4. After a multi-dimensional, logical space for Sourcing patterns is presented in Section 4.1, the values on the space axis are described in detail. Finally, subsequent research areas related to Sourcing are given in Section 5 followed by a conclusion in Section 6.

2 Tackling Complexity

Since Sourcing is embedded in DIBPM, a model is required to manage the complexity resulting from matching and subsequently enacting inter-organizational processes. A definition of DIBPM [17] 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 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 [20] is a suitable model.

The bottom level of Figure 1 shows the internal level where processes are directly enactable by process management applications, e.g., by workflow management systems. Using an internal level in the domains of a service consumer and provider caters towards a heterogenous system environment. In the conceptual level the business processes are designed independent from infrastructure and collaboration specifics. Conceptual processes are mapped to their respective internal level for enactment. A language used on the conceptual level should have clearly defined semantics. The external level stretches across the domains of process initiator and responder. Parts of the conceptual processes are projected to the external level to support automated and dynamically forged, process-based collaboration.

After tackling the issue of complexity in DIBPM, the following section defines Sourcing, gives an example, and presents important control-flow properties, followed by a subsection about quality features of Sourcing.

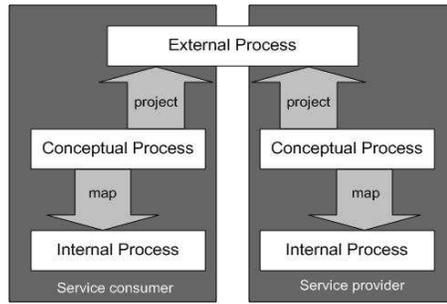


Fig. 1. A three-level business process framework

3 Sourcing and Control-Flow

Sourcing embedded in DIBPM is essential for the automatic structural matching of parties that wish to collaborate. The following definition of Sourcing is used:

Sourcing is matching on an external level conceptually formulated service consuming and providing processes belonging to the domains of autonomous organizations for the formation of an inter-organizationally linked business process.

The next section presents Sourcing in a configuration example that focuses on the control-flow perspective and is modelled using labelled Petri nets [24, 25].

3.1 An Exemplary Sourcing Configuration

Sourcing is found in many market configurations. For example, a travel agency is approached by a customer for booking a flight ticket. Thus, the travel agency serves as a front office that offers complex travelling products consisting of parts that it sources in from other, specialized tourism-industry companies.

Figure 2 depicts the in-house process of the travel agency on the conceptual level. Passive nodes labelled with i and o are the unique input and output places of a sphere and also of the in-house process of which the sphere is a subnet.

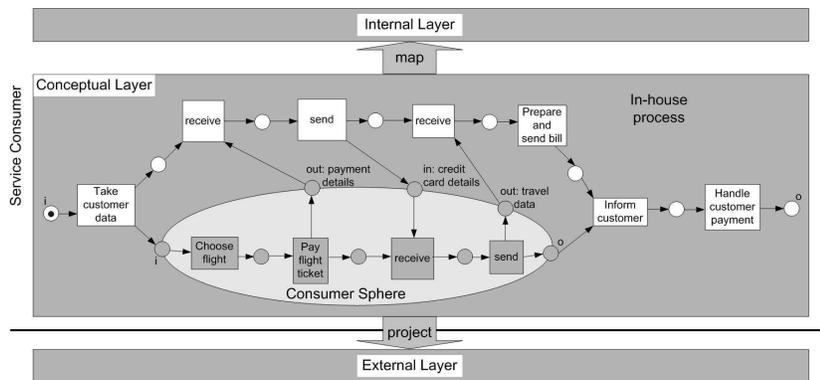


Fig. 2. A conceptual-level process of a service consumer

Next, the content of Figure 2 is explained. Following the three-level business process framework of Figure 1, the consumer's in-house process is mapped to an internal-level process that is employable for SOBI, e.g., BPEL or other

web service composition languages. Regarding the conceptual level process, after starting the process by taking the customer's details, the particular travelling wishes are filled into an online form. Next, a parallel split is modelled where one branch contains a consumer sphere that is depicted with a grey shaded ellipse.

The other parallel branch handles exchanges with the consumer sphere. In that sphere the travel agency allocates a flight ticket for the customer while billing matters are prepared and exchanged with the in-house process. Eventually, the parallel branches are joined by a node that results in informing the customer about all flight details followed by the handling of customer payment. The exchange direction between the consumer sphere and the rest of the in-house process is recognizable by the *in* and *out* labels of interface places.

Booking a flight ticket is not core business of the travel agency. Thus, the travel agency assigns the consumer sphere to a separate organization that functions as a service provider. Relating this situation to Figure 3, the consumer sphere is projected entirely to the external level turning it into the consumer's contractual sphere. The Sourcing counterpart responds with projecting a complementary provider's contractual level to the external level. Consensus is achieved when the respective contractual spheres are equal as depicted in Figure 3. Referencing different but equally labelled nodes contained in the respective contractual spheres defines for a service consumer the degree of monitorability of a Sourcing configuration during enactment.

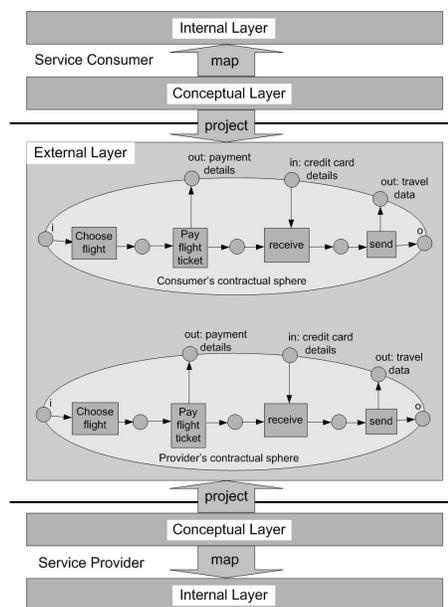


Fig. 3. An external level of a Sourcing configuration

In compliance with the three-level model of Figure 1, the service provider also has a conceptual level. In Figure 4 the conceptual-level process is depicted, which is a refined sphere in correlation to the contractual sphere. Active nodes with labels equally contained in the provider's contractual sphere are visualized using broader lining. Again, Figure 4 shows an internal level on which the refined sphere of the conceptual level is mapped.

Next, the refined sphere of the service provider is explained. An active node for choosing a flight ticket is followed by an added parallel branch in which the flight ticket is booked and the billing organized. These additional active nodes carry labels that do not exist in the corresponding contractual spheres. Thus, the service consumer is not aware of this refinement. After the provider sends the payment data, further refining nodes for posting the flight ticket and checking payment are contained in the refined sphere. The sequence continues with two consecutive nodes, one for receiving credit card details and a following send node for exchanging travel data with the service consumer's in-house process. Note that data flow in Sourcing is not the focus of this paper. However, explicit send

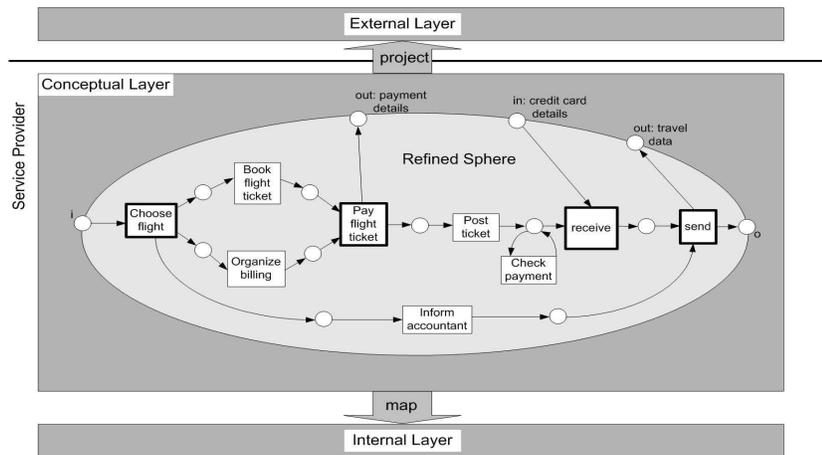


Fig. 4. A conceptual-level process of a service provider

and receive nodes in combination with *in* and *out*-labelled passive interface nodes lay the foundation for developing sophisticated inter-organizational data flow for Sourcing instances.

Once enactment of the refined sphere is completed, an active node the in-house process is enabled for informing the customer. Since this active node is outside of the consumer sphere, the service provider is not aware of such a step. The in-house process concludes with handling customer payment. After that the enactment of the Sourcing configuration is completed.

The figures of this subsection depict several Petri-net related properties that are required for a Sourcing instance. Thus, the next section defines those properties followed by a subsection about a desirable requirement of Sourcing.

3.2 Control-Flow Properties of Sourcing

With respect to control flow in Sourcing, a subclass of Petri nets is used, namely so called *workflow nets* (WF-net)[3] that have been further explored [4, 16]. Informally, a WF-net is a special subclass of Petri nets that has one unique input state (*i*) and one unique output state (*o*). There may be no ‘dangling active and/or passive nodes’, i.e., active and passive nodes 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 state *o* and all other states are empty. Moreover, it should be possible to execute an arbitrary active node by following the appropriate route through a workflow. The latter requirements are properties of *soundness* and a strong argument for using WF-nets for Sourcing. 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 [29].

Looking at the conceptual-level process of Figure 2, the in-house process with the contained consumer sphere is a sound WF-net. The in-house process has a unique input place with one token, no nodes are dangling, and there is a unique output place where only one token is left once enactment has completed. The consumer sphere of the service consumer in Figure 2 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 focusses on control-flow in isolation and does not take other context of a Sourcing configuration into account.

Figure 2 shows *in* and *out*-labelled passive nodes that are so-called interface places. They connect active nodes that are located in a consumer sphere and the rest of the in-house process. The labels specify the nature of exchange between the in-house process and a consumer sphere. 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 active input node, an exchange is attempted between the domains of a service consumer and provider.

Introducing these *in* or *out*-labelled interface places permit to set up the control-flow construct with which soundness can be verified before an actual exchange between Sourcing 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 Sourcing is not the focus of this paper.

A refined sphere of a service provider as depicted on the conceptual level of Figure 4 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 the figures of Subsection 3.1 show, the active nodes depict labels. Comparing the consumer sphere and the refined sphere, one notes the labels of active nodes 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 active-node labels from the domain of a consumer that are not projected to the external level. Likewise, a service consumer is not aware of labels contained in a refined sphere that are not projected to the external level, as depicted in Figure 3. Finally, instantiating and enacting the Sourcing configuration of Subsection 3.1 results in a special feature with respect to monitorability.

Since the refined sphere of the service provider contains additionally inserted active nodes, a so-called "early token" may occur in the consumer sphere that does not enable a subsequently following active node. For example, in Figure 4 the enactment of the active node labelled *choose flight* in the refined sphere is messaged to the corresponding node of the consumer sphere. As a result the subsequent output node receives an early token while inserted nodes still need to be carried out in the refined sphere before the active node labelled *pay flight ticket* is enabled.

3.3 Quality Criteria of Sourcing

Respecting control flow, the question arises how a service consumer ensures that a provider adheres to the contractual sphere in a Sourcing configuration while not imposing fixed routing in the provider's domain. Therefore a notion of *inheritance* is introduced that focusses on the dynamics rather than data and/or signatures of methods. Four notions of such dynamic inheritance have been identified in [6, 13] 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 Sourcing 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 behaviors 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 behavior 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 Sourcing configuration of the figures in Subsection 3.2, hidden steps are active nodes of a refined sphere where their labels are not contained in the corresponding contractual sphere. Let's assume that all the labels of active nodes of a consumer's sphere are also contained in a refined sphere. When a consumer's 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 a Sourcing configuration. Looking at the Sourcing configuration of the figures in Subsection 3.2, 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 active node from a contractual sphere to an interface place, then there is one receiving active node located in the in-house process outside of the consumer sphere. Likewise if there exists one sending active node from an in-house process outside of a sphere to an interface place, then there is one receiving active node located in a contractual sphere. Furthermore, if the sending active node is located in a consumer sphere, the refined sphere must equally have a sending node to an equally labelled interface place. When a receiving active node is located in a consumer sphere, the refined sphere must also have a receiving active node to an equally labelled interface place.

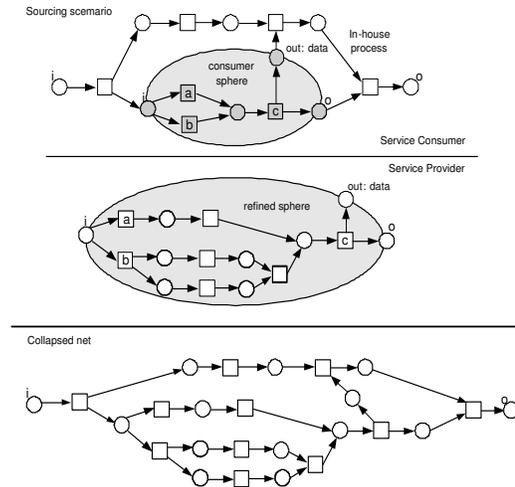


Fig. 5. Collapsing a Sourcing configuration

Finally, after presenting the notions of projection inheritance and well-directedness, the requirement of soundness of a Sourcing configuration is covered. An evaluation is achieved by introducing the procedure of *collapsing*. Figure 5 shows at the top the service consumer's and provider's conceptual-level processes. 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's sphere. Likewise, the corresponding refined sphere of the provider has an interface place where a sending active node 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 5.

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 Sourcing configuration must always be a sound WF-net. In the case of *Sourcing 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 a Sourcing 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 a Sourcing configuration is a required minimum that must always be adhered to with respect to control flow. Otherwise the enactment of a corresponding Sourcing 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 doesn't 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 Sourcing 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 Subsection 3.1 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 Sourcing, the third relevant issue of investigation in this paper is a suitability analysis. That way relevant features for Sourcing are detected that supporting WSCLs and their corresponding enactment applications should to incorporate.

4 Suitability Features of Sourcing

As the previous section demonstrates, control-flow is one fundamental perspective of Sourcing. In this section other related perspectives are discussed. Informally, a perspective is a particular angle from which a certain domain is regarded. Figure 6 relates Sourcing 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. Sourcing rests on other relevant perspectives depicted as pillars in the center of Figure 6 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 a Sourcing 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. 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 how the involvement of such resources is defined. Finally, the enactment of a Sourcing configuration must offer a degree of certainty. By including a transaction pillar, enacted Sourcing steps are secured and exceptional situations are handled and compensated if required.

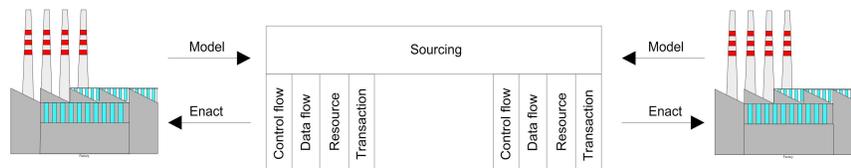


Fig. 6. Relating perspectives for DIBPM

For Sourcing the objective is pursued to realize expressive and suitable WSCL and web-based enactment middleware for supporting a scenario as depicted in Figure 6 in an automated way. Therefore, a well-structured approach is needed to explore the features of the perspectives contained in Figure 6 in order to create a foundation for developing a suitable language for Sourcing. The problem of having to deal with a high degree of complexity must be tackled that results from a heterogenous system environment of collaborating business parties.

To explore the Sourcing perspective, the following method is chosen. DIBPM contains several feature dimensions in the form of axis 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 axis, a logical space is created that represents a particular DIBPM perspective. Subsequently, the axis 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. The discovered patterns are instrumental for creating suitable WSCLs and web-based enactment middleware for Sourcing. Note that the specification of such patterns is not the focus of this paper.

By evaluating the features used in a particular Sourcing scenario, it is possible to perform a positioning in the logical space. The following subsection creates such a logical space for the Sourcing perspective and gives a detailed description of dimension values that are located on the perspective axis. Different logical spaces can be constructed for other perspectives.

4.1 A Multidimensional Space for Positioning Sourcing Configurations

The approach of investigating DIBPM by looking at features within different, related perspectives is represented in Figure 7. A cube is depicted, created by three axis representing different Sourcing dimensions on which further detailing values are positioned. Each of those values creates a taxonomy and is guiding for later pattern discovery. The created multi-dimensional space permits the positioning of Sourcing configurations. In Figure 7 the black circles on the walls of the cube stand for dimension-value instances, i.e., patterns, that are employed to create a Sourcing configurations.

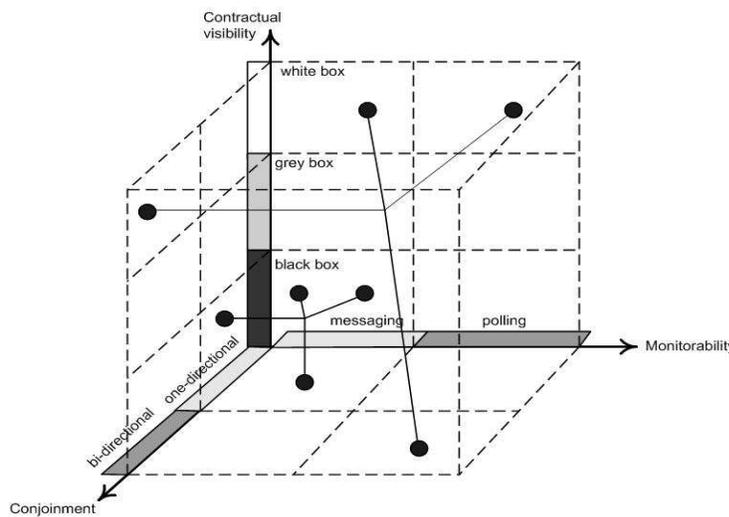


Fig. 7. Dimensions and values of the Sourcing perspective

In Figure 6 Sourcing is depicted as an integral perspective that rests on other perspective pillars in the domains of the service provider and consumer. Contractual visibility, monitorability, and conjoinment create a perspective cube for positioning Sourcing configurations. However, these dimensions are also a foundation for positioning DIBPM configurations in different multi-dimensional perspective spaces. For example, the conjoinment dimension in combination with other dimensions result in the creation of a perspective space for positioning DIBPM configurations from a data-flow point of view. Examples of such dimensions are data visibility that relates to the extent and manner in which data elements can be viewed by various components of a workflow process, or data transfer that considers the means by which the actual transfer of data elements occurs and the various mechanisms by which data elements can be passed across the interface of spheres. On the other hand, the monitorability dimension in combination with other dimensions creates a space comparable to Figure 7 for creating DIBPM configurations from an inter-organizational resource-perspective point of view, and so on.

While the specification of dimension-values deduced patterns is out of scope for this paper, the following conclusions hold. The axis of the multi-dimensional space of Figure 7 carry the dimensions called contractual visibility, monitorability, and conjoinment. Further refining values depicted on the first dimension have exactly one pattern assigned. The other values positioned on the latter two dimensions have multiple patterns assigned.

4.2 Detailing the Sourcing Dimensions

The cube dimensions and values of Figure 7 are described as follows:

- **Contractual Visibility:** is focussing on how much process detail is disclosed at build time of a Sourcing configuration. In Figure 3 a consensus exists since both spheres are similar. Three values are located on the contractual-visibility dimension of Figure 7. The values are regarding the correlation of the amount of nodes contained in the contractual sphere of the external level and the amount of nodes in the conceptual-level's consumer sphere. First, a *white-box* value means that all nodes of a consumer sphere are contained in the contractual sphere of the external level. In case of a *black-box* classifier only the interfaces of a consumer sphere are projected to the contractual level. Finally, the *grey-box* classifier means the interface places and a subset of the nodes and arcs of the consumer sphere are projected to the external level's contractual sphere. Patterns covering different levels of visibility in Sourcing configuration are mentioned in relation with dynamic service outsourcing [20]. However, the listed patterns are all values of a control-flow dimension and are not separated into the axis contractual visibility and monitorability as in this paper. For example, the black box pattern roughly corresponds to black-box contractual visibility in the sense that a consumer has no information about how the service is executed by the provider. The glass-box pattern roughly corresponds to the white-box contractual visibility as the consumer is aware of internal states of the outsourced process. Since it is not clear to which extent internal states are disclosed, the glass-box pattern can also be interpreted as similar to the grey box. The half-open box and the open box patterns are for service synchronization where the first pattern only synchronizes from service consumer to provider and the latter pattern allows for synchronization in both directions. While nothing equivalent is contained in the dimension contractual visibility, these patterns are assignable to the dimension described below.
- **Monitorability:** covers the way how nodes in a consumer's and provider's contractual spheres of the external level are linked with each other. Available linking values are *messaging* and *polling*. The nodes of the contractual spheres are connected to nodes in the corresponding spheres in the respective conceptual levels. Various degrees of monitorability of service provision are achieved for a service consumer during the enactment of a Sourcing configuration. At a minimum all interface places of both contractual spheres need to be linked with each other to transitively relate the consumer sphere and refined sphere with each other. Additional passive and active nodes of the contractual spheres may be linked. *Messaging* can be applied to linked passive and active nodes contained in both contractual spheres. If two passive nodes are linked in a messaging way, the node experiencing a change in number of tokens signals that information to the linked destination node. When active nodes are linked in a messaging way, the executing node messages this event to the linked active node. The classifier termed *polling* links nodes of the service consumer's and provider's contractual sphere in a way where one node periodically checks whether the linked node has changed. When passive nodes are linked in a polling way, one node checks if the number of contained tokens has altered. Subsequently, the change is mirrored by the polling node. If two active nodes are linked in a polling way, one node periodically checks if the linked node has experienced an execution event. Subsequently, the polling node reflects the execution event.
- **Conjoinment:** between the in-house process and the consumer sphere is projected to the external level. Consequently the provider's contractual sphere contains equal conjoinment constructs that result in transitive exchanges during enactment between the in-house process and the refined sphere of the service provider. This way the basis for exchange is established across the domains from the service provider to the consumer and vice versa. *One-directional* conjoining implies that there is one *out* or *in*-labelled interface place present that is either complemented by an active send node or an active receive node respectively. Those active send and receive nodes are contained in the spheres of a Sourcing configuration and handle the exchange between the domains of a service consumer and provider. *Bi-directional* conjoining is initiated by a sending active node to the domain of the Sourcing counterpart that returns the communication exchange immediately to the initiating Sourcing party.

When the values are used as a taxonomy for the specification of patterns, the following statements hold: For creating an instance of a Sourcing configuration, exactly one pattern belonging to a value of contractual visibility must be chosen. Regarding the monitorability dimension, at least two patterns have to be used, either from the value messaging or polling. That way the service provider's refined sphere is transitively connected to the consumer sphere via the external level. The minimum amount of connected nodes are the respective *i* and *o*-labelled interface places of

the spheres located on the external and conceptual levels. If increased monitorability is desired by the consumer and granted by the provider, multiple patterns from both values are optionally used to link more nodes contained in the spheres. Finally, the conjunction dimension is optional and not required for the creation of a Sourcing configuration.

5 Related Research

Earlier research dealing with inter-organizational process management has been carried out. The WISE project (Workflow based Internet Services) [12] resulted in a software platform that enacts a virtual business process consisting of several, linked black-box services. The CrossFlow project resulted in the exploration of outsourcing [18, 19]. However, in CrossFlow, parts of a service consumer's process are assigned to a willing service provider without looking at more formal matters of inter-organizational collaboration. Examples for such matters are the evaluation of overall correct process termination, the matching of a service consumer and provider, and a notion of inheritance that goes further than node refinement. In the CrossWork project [1] Sourcing is an integral concept. CrossWork is developed under the auspices of the European Commission within the IST-VI Framework program. In CrossWork the objective is to develop automated mechanisms for allowing dynamic workflow formation and enactment, enabling hard collaboration and strong synergies between different organizations.

The issue of private and public workflow views [28] has been addressed for a client-server model of B2B collaboration and translated into an inter-organizational workflow management system called Nehemiah. However, the approach is based on workflow graphs where it is problematic to verify before enactment the correct control-flow termination [7] of an inter-organizational process. Furthermore, it is not clear how the workflow-view approach ensures internal adherence of a service provider to the agreed upon public process.

Regarding the required formal rigor for client/server DIBPM, numerous research results are available. For evaluating the control-flow adherence of a service-providing to a consuming process, a well explored notion of projection inheritance [5] is instrumental. That notion of projection inheritance is defined employing Petri-net theory [24, 25]. Concretely, a subclass of Petri-nets is used, namely *workflow nets* (WF-net) [16] that is well explored [3, 4] and carries the appealing notion of soundness [2], i.e., whether a WF-net can successfully terminate or not. In more recent research, the issue of correct termination of composed web services is regarded [23] by analyzing them respectively as white boxes using a weak notion of soundness. It is evaluated whether a modular web service adheres to the so-called notion of usability [22] by other workflow modules in a composed system. Correlated to usability, the notion of semantic compatibility [21] between web services is investigated. However, this research formally investigates distributed business processes in the shape of composed web services in a peer-to-peer model.

With respect to workflow patterns, in the area of control flow a set of patterns was generated [8–10] by investigating several intra-organizational workflow management systems. Furthermore, patterns for the intra-organizational data-flow and resource management [26, 27] were discovered and specified.

6 Conclusion

This paper positions Sourcing in the framework of dynamic inter-organizational business process management, which combines workflow concepts with service-oriented business integration. Three crucial issues for the evolution of Sourcing are investigated, namely complexity management, control-flow rigor, and suitability.

Sourcing uses a three-level business process framework to manage the conceptual, business, and technological complexity involved in inter-organizational business process collaboration. Furthermore, this paper informally defines Sourcing and its control-flow properties to lay a foundation for termination verification. These requirements and supporting notions with respect to control-flow result in a method for checking during build time the adherence of Sourcing parties to process-behavior requirements and whether the enactment of a Sourcing configuration can successfully terminate.

To succeed in exploring crucial suitability features for Sourcing, the establishment of a multi-dimensional, logical space is pursued. This logical space permits the positioning of individual Sourcing configurations. The axis values of the multi-dimensional Sourcing space are elaborated upon. The logical space creates a taxonomy for subsequent, top-down patterns discovery. These patterns will be instrumental for developing suitable and expressive web service composition languages and their corresponding applications for formulating and enacting dynamic inter-organizational business processes.

References

1. CrossWork: Cross-Organisational Workflow Formation and Enactment. <http://www.crosswork.info/>.
2. W.M.P. van der Aalst. Structural Characterizations of Sound Workflow Nets. Computing Science Reports 96/23, Eindhoven University of Technology, Eindhoven, 1996.
3. 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.
4. 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.
5. W.M.P. van der Aalst. Inheritance of Interorganizational Workflows: How to Agree to Disagree Without Losing Control? BETA Working Paper Series, WP 46, Eindhoven University of Technology, Eindhoven, 2000.
6. 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.
7. W.M.P. van der Aalst, A. Hirnschall, and H.M.W. Verbeek. An Alternative Way to Analyze Workflow Graphs. In J. Mylopoulos and C. Woo, editors, *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02)*, volume 2348 of *Lecture Notes in Computer Science*, pages 535–552. Springer-Verlag, Berlin, 2002.
8. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns Home Page. <http://www.tm.tue.nl/it/research/patterns/>.
9. 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.
10. 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.
11. W.M.P. van der Aalst, M.Dumas, A.H.M. ter Hofstede, and Petia Wohed. Pattern based analysis of bpml (and wsci). *QUT Technical report, FIT-TR-2002-04, Queensland University of Technology, Brisbane, 2002.*, 2002.
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. 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.
14. BPMN.org. *Business Process Modeling Language (BPMN) version 1.0*. Accessed August 2003 from www.bpmi.org, 2003.
15. F. Curbera, Y. Golland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana. *Business Process Execution Language for Web-Services*. <http://www-106.ibm.com/developerworks/library/ws-bpel/>, 2003.
16. 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.
17. P. Grefen. Service-Oriented Support for Dynamic Interorganizational Business Process Management. to appear, 2005.
18. 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.
19. P. Grefen, A. Dan H. Ludwig, and S. Angelov. Web service support for dynamic business process outsourcing. *IBM Research Report*, (RC22728(W0302-089)), February 2004.
20. 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.
21. Axel Martens. On compatibility of web services. *Petri Net Newsletter*, 65:12–20, 2003.
22. Axel Martens. On usability of web services. In Coral Calero, Oscar Daz, and Mario Piattini, editors, *Proceedings of 1st Web Services Quality Workshop (WQW 2003)*, Rome, Italy 2003.
23. Axel Martens. *Verteilte Geschäftsprozesse - Modellierung und Verifikation mit Hilfe von Web Services*. Dissertation, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2003.
24. 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.
25. W. Reisig and G. Rozenberg, editors. *Lectures on Petri Nets II: Applications*, volume 1492 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.
26. N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Resource Patterns. *QUT Technical report*, forthcoming.
27. 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.
28. K.A. Schulz and M.E. Orłowska. Facilitating cross-organizational workflows with a workflow view approach. *Data & Knowledge Engineering*, 51:109–147, 2004.

29. 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.
30. P. Wohed, W.M.P. van der Aalst, M. Dumas, and A.H.M. ter Hofstede. Pattern based analysis of bpm4ws. *QUT Technical report, FIT-TR-2002-05, Queensland University of Technology, Brisbane, 2002.*, 2002.