# Developing a Reference Architecture for Inter-Organizational Business Collaboration Setup Systems

A. Norta and P. 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.norta@tm.tue.nl

**Abstract.** The question of how a service consumer and a service provider should collaborate with each other in a business-to-business (B2B) setting is an ongoing research issue. The concept of electronic Sourcing (eSourcing) has been proposed as an integral concept with the EU research project CrossWork. The properties of eSourcing are explored, however, the question arises how a service consumer and a provider need to interact with each other during setup time for establishing an eSourcing configuration? The concept of eSourcing offers more flexibility than other approaches in that respect. Thus, this paper investigates the characteristics of interaction between a service consumer and provider during setup time for establishing an enactable B2B collaboration. These characteristics are employed to discover interaction patterns in a dop-down way that are exemplified for the concept of eSourcing. As such, the discovered patterns form the basis for the design of a reference architecture that serves as a foundation for the design of e-collaboration setup systems.

## 1 Introduction

Collaboration between industrial companies in the B2B domain is complex from a business,, conceptual, and technological point of view. Observing such collaboration, particular features can be pointed out. In the automobile industry an original equipment manufacturer (OEM) organizes the creation of value in an in-house process that is decomposable into different perspectives,e.g., control flow of tasks, information flow, personnel management, allocation of production resources, and so on. A complex product of an OEM typically comprises many components of which several need to be acquired from suppliers. The reasons for acquiring parts externally are manifold, e.g., the OEM can't produce with the same quality, or an equally low price per piece, the production capacity is not available, required special know-how is lacking, and so on.

Investigations about how OEMs in the automobile industry choose suppliers for creating their B2B supply chains show that OEMs impose strict requirements. The latter parties are requested to perform mirroring of the particular parts of the in-house process the OEM doesn't want to perform itself. In such a constellation the OEM is considered a service consumer and suppliers are service providers. In an extreme case, process mirroring means, that a service provider has to display the same behavior with respect to control-flow, data-flow, resources, and so on, during service enactment, compared to the situation where a consumer performs the service as a part of a bigger in-house process. A potential service provider is not considered by a consumer if the first party can't assure the capability of process mirroring. However, in other industry domains the opposite extreme is thinkable where a service consumer does not impose any restrictions on steps that create the desired service provision.

For a service consumer, finding a suitable partner is hampered by several factors. Firstly, experience shows that respective parties model their services using different methodologies. That way it is difficult to communicate and decide to which extent process mirroring is possible. Practitioners lack a concept with which inter-organizational business process collaboration can be expressed. Additionally, the parties are reluctant to disclose internal business details for fear of revealing their competitive advantages. That makes it even harder to asses the capability of process mirroring.

With the application of service-oriented information technology, companies hope to reduce the costs for personnel that deals with establishing and maintaining a B2B supply chain. It is a worthwhile objective that enterprizes should find each other dynamically with the support of middleware in an effective and efficient way for providing and consuming a service. Most research projects about inter-organizational collaboration [1, 11, 25, 30] focus predominantly on the business-process aspect.

Given the described difficulties, service consumers and providers are confronted with a considerable communication effort during the setup and enactment of B2B collaboration. Observations show various forces, e.g., the degree of bargaining power determine which party initiates service consumption, which suggests the setup phase of inter-organizational business process collaboration may follow different paths. As a means of investigating those different collaboration paths, this paper pursues a pattern-based investigation for achieving an improved insight concerning the setup phase of B2B collaboration. The result of that investigation is a set of interaction patterns between a service consumer and one or several service providers. These interaction patterns are input for exploring the features of middleware that supports dynamic inter-organizational business process collaboration in a suitable way.

The interaction patterns and the deduced reference architecture are embedded in the emerging framework of dynamic inter-organizational business process management (DIBPM) [18]. This framework of DIBPM 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. To do so, DIBPM merges service-oriented business integration (SOBI) and workflow management concepts. 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.

The structure of this paper is as follows. First, Section 2 presents examples of reference architecture and research projects about inter-organizational business process collaboration. Furthermore, since business-collaboration applications increasingly apply service-oriented technology, several XML-based industry standards are explained. Section 3 introduces the reader to the concept of electronic *Sourcing* that is instrumental in an ongoing research project where a proof-of-concept prototype for dynamic inter-organizational business process management is under construction. this prototype improves the effectiveness and efficiency of coordinating service providers across several tiers of a supply chain. In this paper the specifications of interaction papers use the concept of eSourcing as a vehicle for giving examples. In Section 4, characteristics and their further refining values are discovered for the domain of interaction patterns. These characteristics create a two-dimensional-logical space from which interaction patterns are deducted in a top-down way that occur between a service consumer and provider during the setup time of inter-organizational business process collaboration. In Section 5 and Section 6 interaction pattern specifications are given. Next, in Section 7 a reference architecture is presented that is a foundation for the design and development of an eSourcing setup systems. Section 8 illustrates in a case study how a chosen combination of interaction pattern is supported by the reference architecture. This case study is taken from industry partners of the automobile industry that are involved in an ongoing EU research project and finally Section 9 concludes the paper.

## 2 Related Work

The following subsections of related work are about pattern catalogues and how they are organized, examples of reference architectures, and completed and ongoing research projects about business collaboration. The patterns comprise of the intra-organizational domains called control-flow, data-flow, and resource. Furthermore, service-interaction patterns are outlined and finally e-business related patterns are mentioned. The selected pattern catalogues are useful for conceptually understanding how to set up intra-organizational business processes and show how inter-organizational e-business collaboration is possible with the help of service-oriented technology. The way these patterns are organized is input for organizing the interaction patterns of this paper.

The reference architectures describe how to design communications and computer network protocols for inter-organizational collaboration, how to enable intra-organizational business-process, and finally, how to establish electronic contracts between collaborating business parties. High-level components of the latter to reference architectures are incorporated in the reference architecture of this paper that is deduced from the specified interaction patterns.

In the next subsection four research project are listed that deal with the issue of inter-organizational business process management. These projects are dealing to different extents with the issue of interactions that are required during setup time. Dedicated matchmaking components with process catalogues, and more recently ontological concepts, are proposed for bringing business parties together.

### 2.1 Related Pattern Catalogues

Referencing patterns, Gamma et al. [17] 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 [7–9] 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 for the evaluation [6, 38] of WSCLs.

Following a similar approach as in the control-flow perspective, data-flow patterns [32] are grouped into various characteristics categories. One category is focuses on different visibility levels of data elements by various components of a workflow system. The category called data interaction focusses 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 [31] are aligned to a 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.

So-called service interaction patterns [14] 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 multi 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.

Many other e-business related patterns are textually available online [3] for the perspectives business-, integration-, composite-, custom design-, application- and runtime patterns. The business perspective highlights the most commonly observed interactions between users, businesses, and data. Integration patterns connect business patterns for creating composite patterns. Patterns of the composite perspective that combine business and integration patterns are only documented when they often occur in reality. The perspective custom design is similar to composite patterns. Finally, patterns from the application perspective focus on the partitioning of the application logic and data while patterns from the runtime perspective use nodes to group functional requirements that are interconnected to solve a business problem.

### 2.2 Reference Architectures

Many other efforts exist for the definition of reference architectures in the various domains of application of information systems. A few examples are the well-known OSI reference model [33], the Workflow Reference Model [37], the Reference Architecture for Workflow Management Systems [20], and more recently the E-contracting Reference Architecture [12].

The OSI reference model [33] is a layered abstract description for communications and computer network protocol design, developed as part of the Open Systems Interconnection initiative. The OSI model divides the functions of a protocol into a series of layers. Each layer has the property that it only uses the functions of the layer below, and only exports functionality to the layer above.

The high-level view of the Reference Architecture for Workflow Management Systems [20], and the Workflow Reference Model [37] are used in the reference architecture for supporting interaction patterns. Since the interactions between collaborating parties results in the creation of inter-organizational business process configurations, the Reference Architecture for Workflow Management Systems and the Workflow Reference Model are incorporated to enable local business-process setup and subsequently with their enactment.

Finally, the E-contracting Reference Architecture [12] caters for matchmaking-, partner selection-, negotiation and contract establishment-, contract enactment-, and management functionalities. In the reference architecture that supports the interaction patterns of this paper, high-level components of the E-contracting Reference Architecture are incorporated.

### 2.3 Research Projects

As the introduction of this paper mentions, several research projects have investigated inter-organizational process collaboration. In the CrossFlow project [30], the formation of virtual enterprizes is realized by dynamic outsourcing. A service matchmaker matches a service offerings and service requests. Based on this electronic contract a service enactment infrastructure [21] is established dynamically, employing on 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 [2] forms the internal process level.

In the WISE project [11, 25], a software platform is provided for process-based B2B electronic commerce that focusses on support for a network of small and medium enterprizes. While CrossFlow relies on cooperating pairs of autonomous workflow systems with a peer-to-peer relation, WISE relies on a central workflow engine to control cross-organizational process 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[11]. A service is offered by an involved organization in a WWW catalogue and can be a business process that is controlled by a local workflow management system.

In the CrossWork [1] project, the objective is pursued to develop automated mechanisms for allowing dynamic workflow formation and enactment, enabling collaboration and strong synergies between different organizations. CrossWork uses *eSourcing* [28] as an integral concept which focusses on matching on an external level conceptually formulated service consuming and providing processes. Further details are explained in Section 3.

The web-Pilarcos B2B middleware [24] represents one solution for managing the life-cycle of dynamic business networks in an inter-enterprize environment. It forms a loosely-coupled collaboration layer on top of service-oriented middleware by using meta-level information about business processes and service descriptions of participants. A renegotiation of this meta-level information results in an automatic reconfiguration of the underlying system. In web-Pilarcos inter-organizational collaborations are modelled as *eCommunities* that comprise of independently developed business services. Such eCommunities are established by a willing party that joins service that providers earlier submit to a public repository. However, the architecture requires further development of business process modelling techniques. According to [24] the collaboration of business processes or workflows should be modelled without revealing local processing steps. Instead, only the collaborative part's external view should be agreed on and monitored.

After presenting related work, the following section briefly presents the concept of eSourcing that is an integral part of the CrossWork research project. For this paper the eSourcing concept is a vehicle with which examples for interaction patterns are specified.

## 3 The Concept of eSourcing

Within DIBPM the concept of eSourcing is concerned with the structural matching of processes belonging to the domain of a service consumer and a service provider. Thus, in eSourcing those processes are matched by analyzing the respective control-flow. Furthermore, eSourcing is a foundation for the automatic formulation and enactment of electronic contracts [28]. To manage the inherent complexity of eSourcing, the employment of a three-level framework [19] is considered a suitable model.

In Figure 1 the three-level model is depicted as part of an eSourcing example. 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 center of Figure 1 the external level is stretching 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. Consequently eSourcing is defined as follows:

> *eSourcing is a framework for harmonizing on an external level the intra-organizational business processes of a service consuming and one or many service pro-*

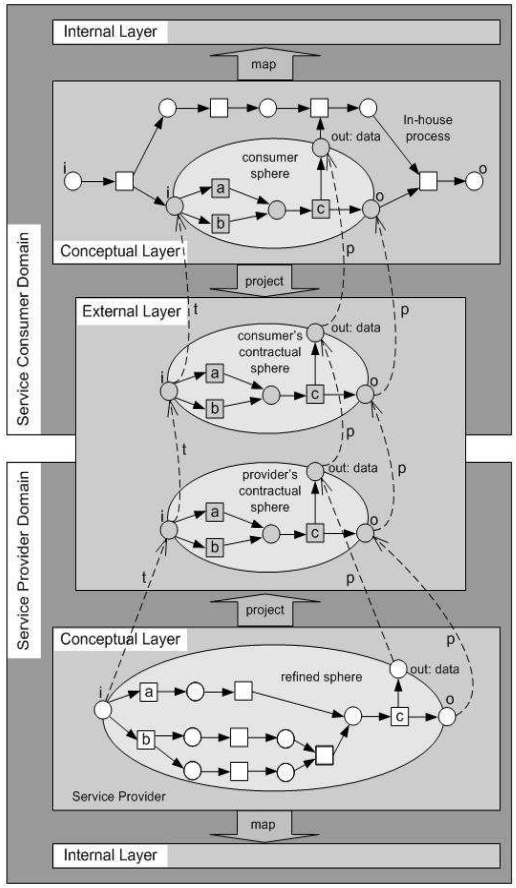


**Fig. 1.** A three-level business process framework

*viding organizations into a B2B supply-chain collaboration. Important elements of Sourcing are the support of different visibility levels of corporate process details for the collaborating counterpart and flexible mechanisms for service monitoring and information exchange.*

To understand the definition of eSourcing better, the example of Figure 1 is used to explain relevant parts of an eSourcing configuration. First, it needs to be mentioned that the example is modelled using labelled Petri nets. Squares are denoted as active nodes as they propel the net and circles are denoted as passive nodes representing states. Furthermore, active nodes are always equipped with labels. In a Petri net, passive nodes may contain tokens and active nodes consume these tokens from states. Active nodes are enabled when all their passive input nodes contain at least one token. Such an enabled, active node consumes one token from each of its passive input nodes during firing. After firing, an active node produces tokens that are placed into all its passive output nodes. Arcs in a Petri net do not link two active nodes or two passive nodes with each other. The special type of Petri nets used in eSourcing, namely workflow nets (WF-nets) [16], has one unique passive input node and one unique passive output node. Furthermore, all other active and passive nodes in a WF-net contribute to its processing. WF-nets carry the property of *soundness* [4, 22], which informally states that after the completion of a net, only one token must remain in the unique passive output node and all other passive nodes must be empty.

Starting with the domain of the service consumer in Figure 1, an in-house process is depicted on the conceptual level that is a WF-net. The in-house process contains a subnet that is termed a consumer sphere that is visualized with a grey ellipse. On the border of the consumer sphere, labelled passive nodes 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 an active node in the sphere. If an interface place is *o* or *out*-labelled, it has one input arc from an active node 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 focusses on control-flow and abstracts from other relevant concepts [13] of electronic contracting.

The provider's contractual sphere is complemented by a refined sphere on the conceptual level of the service provider. Compared to the provider's contractual sphere additional nodes are contained on the refined sphere. In Figure 1 such refinement is depicted by unlabelled active nodes in the refined sphere that do not exist in the provider's contractual sphere. Furthermore, the refinement must be in accordance with *projection inheritance* [5] that is informally defined 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.

To establish connectability between the consumer sphere, the respective contractual spheres, and the refined sphere, the obligatory requirement of *well-directedness* of an eSourcing configuration must be mentioned. This requirement focusses on the interface places of the spheres, which are part of what can be considered exchange channels between spheres and the remaining in-house process. A eSourcing configuration is well-directed when the interface places of the consumer sphere, the respective contractual spheres of the service consumer and provider, and the refined sphere are equal in number and labelling.

Finally, in Figure 1 there exist referencing arcs connecting several passive nodes of the respective spheres. By using these referencing arcs, a transitive relationship is established between the consumer sphere and the refined sphere that ensures a correct start and end of service provision during the enactment of the eSourcing configuration. The notation of these referencing arcs of Figure 1 are part of so called monitorability patterns [29] that also cover the referencing of active nodes contained in the consumer sphere and the refined sphere.

After a brief overview of the eSourcing concept in this section, the following section is concerned with a feature analysis of interactions between collaborating parties in a B2B setting.

## 4 Dimensions and Values of Interaction Patterns

Inter-organizational business process collaboration comprises of several phases. Broadly, there is a setup phase, an enactment phase, and a post-enactment phase for, e.g., quality evaluations of the enactment phase. Business processes are driven by models of different perspectives, e.g., data, resource, control-flow, that are modelled during the setup phase of eSourcing configurations. When intra-organizational business processes are linked in a B2B supply chain, the setup phase requires an additional perspective that focusses on the way how business processes need to be linked inter-organizationally. Thus, the interaction perspective that is the focus of this paper, deals with the the setup phase of establishing an inter-organizational business process between a service consuming and a service providing organization. For the interaction perspective a particular definition is proposed:

> *The interaction perspective focusses on the way a service consuming and a service providing party interact with each other during the setup phase of a B2B supply chain with the objective of aligning their respective intra-organizational business processes on an inter-organizational level.*

The question arises how characteristics of an interaction perspective can be explored in a structured way. Revisiting Subsection 2.1, it is demonstrated that by choosing a pattern-oriented exploration, the problem of inherent technological and business complexity is tackled. Patterns offer valuable problem-solving insights and are relevant for assessing the requirements that applications and formal languages, e.g., XML-based industry standards [6, 38], should adhere to. For the patterns catalogues of Subsection 2.1, a bottom-up approach was chosen where several intra-organizational workflow management systems and web service composition languages (WSCLs) were investigated for commonalities. For a pattern-based analysis of the interaction perspective a bottom-up approach is not feasible. The reason for proceeding top-down is a lack of available systems for B2B collaboration that have gone through a lengthy period of adoption to real world business activities. Thus, a particular definition for interaction patterns is proposed:

*A pattern for the interaction perspective represents a technology independent structure that occurs 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 discover interaction patterns for eSourcing in a structured way, the following method is chosen. As depicted in Figure 2, two feature dimensions exist in the form of axis that create a two-dimensional, logical space that are the result of carefully studying CrossWork [1] case studies and related publications [10, 11, 21, 24, 25, 30]. On every axis further refining dimension values are located. Consequently, the axis and their contained values serve as a taxonomy for discovering, ordering, and relating a set of interaction patterns for eSourcing to each other. Furthermore, by evaluating the types of patterns used, it is possible to position an eSourcing configuration in the multi-dimensional, logical space.

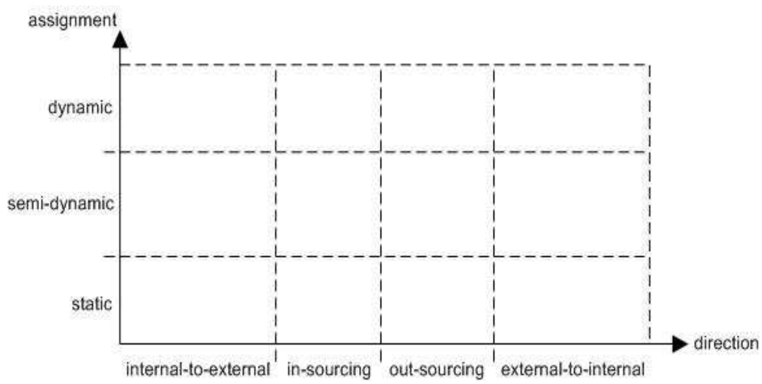


**Fig. 2.** Dimensions and values of interaction patterns

Figure 2 shows the dimensions for interaction patterns. One dimension is called *assignment*, which is concerned with the way a service provider is determined for an eSourcing configuration. The values on the *assignment* dimension state to which degree the collaborating parties know at the beginning of an interaction, that they collaborate with each other during enactment time of an inter-organizational business process configuration.

- *Static* assignment means the collaborating parties already know before setup time they surely collaborate with each other.
- On the other hand, *dynamic* assignment means the collaborating parties bid in an anonymous market for service provision and/or consumption and only towards the end it is clear who collaborates during enactment.
- In the *semi-dynamic* case, the number of collaborating parties that engage in a setup phase is limited at the beginning and therefor known. However, only at the end of the setup phase it is clear who collaborates.

The other dimension depicted in Figure 2 is named *direction* and focusses on the degree of external-level harmonization of inter-organizational business process collaboration. Thus, the interaction between collaborating parties may either start with a harmonization at the beginning of a setup phase or perform harmonization at the end.

- *Internal-to-external* assignment direction means the collaborating parties have internal processes that are only harmonized externally at the end of their setup interaction.
- Likewise, the assignment direction *in-sourcing* means a service provider has a service that is subsequently integrated into the process of a service consumer. Thus, external harmonization is only performed at a later stage.
- *Out-sourcing* is similar to in-sourcing with respect to harmonization. However, now the the consumer starts the interaction with externalizing a service demand first.
- Finally, *the external-to-internal* dimension means that external harmonization is the starting point of interaction and the collaborating parties set up internal processes at a later stage just before enactment starts.

For every setup phase of a B2B collaboration only one assignment and one direction value are chosen. Thus, according to the figure, there are 12 pattern combinations possible. The examples given in this paper represent a subset of these pattern combinations.

### 4.1 A Pattern Specification Template

For specifying discovered interaction 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.

For the examples of patterns, sequence diagrams are used that propose a possible interaction during the setup phase of an eSourcing configuration. However, for sake of brevity completeness of the interactions is not claimed. Furthermore, chunks of the overall interaction sequences are grouped into phases that are reused as interaction blocks in following sequence diagrams. In order to keep the figures brief, those reused interaction blocks abstract from the message exchanges and merely contain activation bars to depict which parties and applications are active in that particular interaction block. The labels of the interaction blocks are unique throughout this section. E.g., the interaction example of Figure 3 depicts the interaction block labelled *contracting* as a white box. Thus, in the first occurrence all message exchanges between the parties involved in contracting are depicted. In contrast, the interaction example of Figure 4 depicts the contracting interaction block as a black box, hiding for sake of brevity the identical message exchanges shown in Figure 3.

The following section explains first the assignment dimension of Figure 2 with the corresponding dimension values. Next, for each dimension value one pattern is deduced that is specified using the description template mentioned above. The same approach is used in Section 6 where the direction dimension is explained and corresponding patterns are specified for each dimension value. Each specification of Section 5 and Section 6 give one abstract example that is geared towards the concept of eSourcing introduced in Section 3, and a real-life example.

## 5 Assignment-Dimension Patterns

**Pattern 1 (Static Assignment)**
***Problem:*** Due to market pressures a company is involved in short production cycles of complex industrial goods with very small order numbers. To reach the objective, it is critical that a significant part of the overall production needs to origin from an external source that can guarantee high quality and time precision.
***Description:*** Before the setup phase of a B2B collaboration begins, a service consumer and a service provider limit their collaboration choice to one candidate. The provision candidate must be able to guarantee the consumer the capability of process mirroring. The sourced service is formulated as a template and externalized to initialize the setup interaction between collaborating parties with the objective to receive a service of predictable time precision and agreed upon quality.
***Forces:*** Achieving tight integration between a service consumer and a provider might fail for different reasons. For example, when a provider is not capable of performing a mirroring of consumer processes, it fails to be a credible candidate. Furthermore, failing technological integration attempts between provider and consumer are a potential obstacle. The provider might not be able to offer requested quality standards of services. Another obstacle may be whether the demanded service is available on time. Availability can be hampered by a lack of human and production resources. Furthermore, the qualification level of human resources may be inadequate.
***Examples:***

- A truck manufacturer has a competitive advantage by delivering according to customer specifications within 17 working days. Such prompt delivery is only assured when the truck manufacturer consumes clearly defined, external services that are reliably available. Therefore, there exists one specially prepared supplier who is capable of performing a mirroring of externalized consumer-processes. Since the chosen supplier happens to possess crucial production know how without which the truck manufacturer isn't able to reach his deadlines, the consumer is interested in a very tight supply-chain integration.

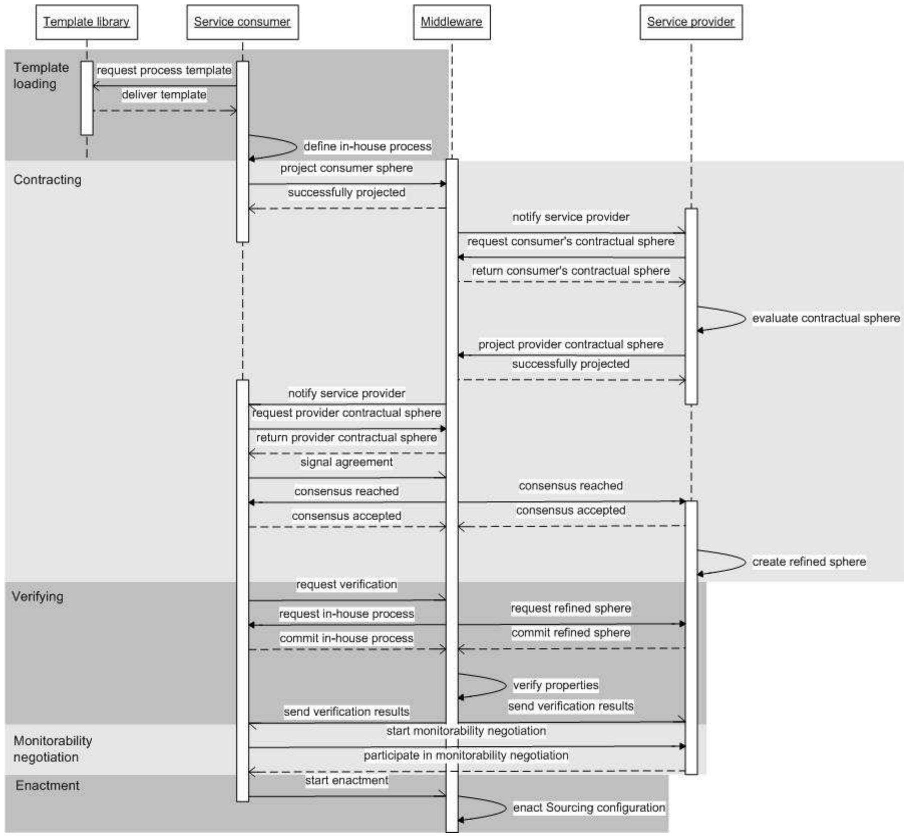


**Fig. 3.** An interaction-sequence example for static service assignment

– An abstract example of static assignment that focusses on eSourcing is given in Figure 3. The a priori tight integration efforts between service consumer and provider are abstracted from. Instead the sequence of interaction starts with a service consumer loading the predefined template resulting from earlier integration efforts. The template represents a consumer sphere that is integrated into the consumer's in-house process. Next, the consumer projects its sphere to the middleware situated between the consumer and the service provider. Consequently, the service provider is informed by the middleware about the committed consumer's contractual sphere. The provider responds with a sphere projection to the middleware. If the respective contractual spheres don't match, the projection procedures need to be repeated until the respective contractual spheres match and a consensus is created. The contracting phase of interacting is ended after the middleware informs the collaborating parties about the shared consensus and when the provider has created its refined sphere. Next, the service consumer requests the middleware to perform a verification of control-flow and data-flow properties of the eSourcing configuration. In order to keep each other's processes secret from each other, the collaborating parties supply the in-house process and the refined sphere to the middleware that performs all checks. The results are sent out to the collaborating parties. If the verification succeeds, the collaborating parties may engage in negotiating the monitorability aspect of the eSourcing configuration. Finally, the service provider sends a message to the middleware that enactment should commence.

**Pattern 2 (Dynamic Assignment)**
***Problem:*** For a part of the overall in-house process, an organization intends to externalize this process part to find a Sourcing counterpart. However, it is not clear who the best counterpart is. Thus, the service-externalizing organization wants to see the prospective candidates engage in a competition for service collaboration.

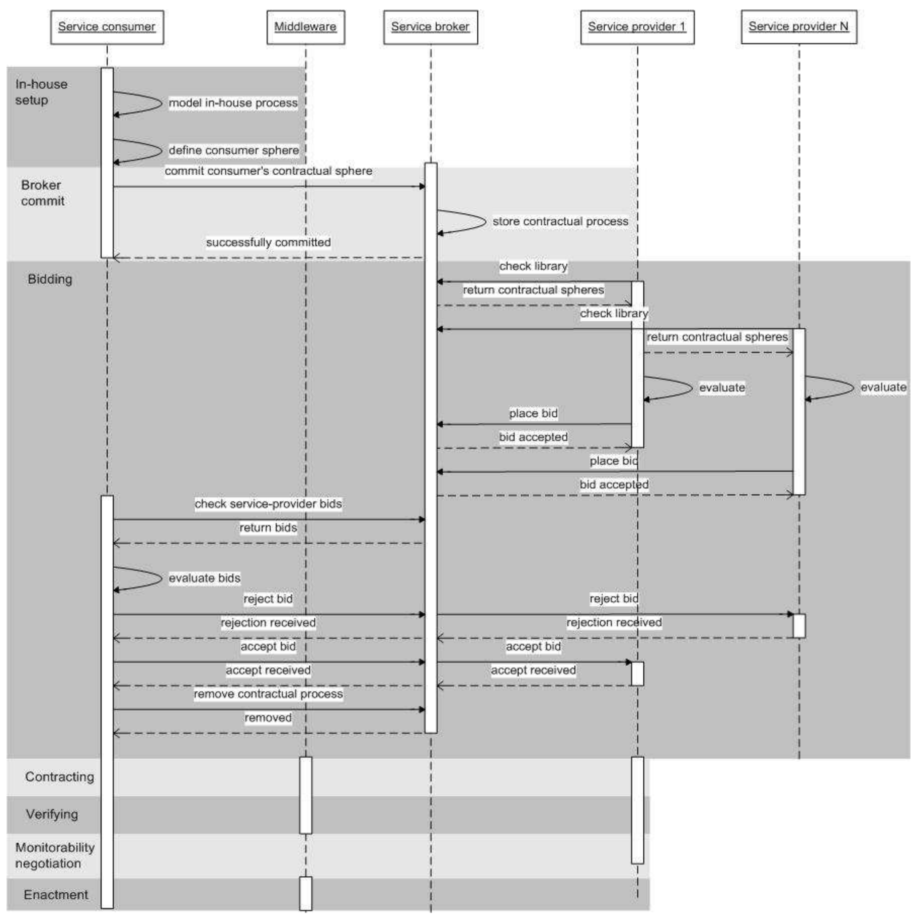


**Fig. 4.** An interaction-sequence example for dynamic service assignment

***Description:*** An organization externalizes its service to a public broker that functions as an anonymous process-market place for the purpose of engaging in a business collaboration. Other organizations may evaluate the published process and decide to engage in a bidding for service. The service-externalizing organization chooses the offer and rejects all other.
***Forces:*** The publicly available process broker needs to be equipped with a directory that allows service consumers to correctly file their process according to attributes like, e.g., the type of industry, geographic location, language, and so on. With the help of a search engine, potential service providers must be able to find those filed processes. Furthermore, it must be ensured that bids and eventually bid acceptance and bid rejections can be communicated between the potential service providers and the consumer. If the consumer happens to be confronted with a big number of provision bids, tool support must permit the evaluation of those bids in an efficient and effective way.
***Examples:***

- A travel agency carries out the booking of a trip for a customer who needs to travel to a capital city. The overall booking comprises a flight ticket, a hotel, and renting a car. However, the hotel booking is carried out by a different company that is actually located in the capital city and therefore has the appropriate expertise. Thus, the travel agency submits the part of the in-house process to a public broker as a request for service provision. Many potential service providers start bidding and the travel agency chooses the best deal and rejects all other offers.
- In Figure 4 a possible setup phase is depicted that uses a dynamic-assignment pattern for creating an eSourcing configuration. First, the service consumer creates an in-house process and defines a consumer sphere in it. Next, the consumer sphere is projected to a publicly available service broker that can be searched by potential service providers. Thus, interested parties place bids for service provisions that are evaluated by the service consumer. The latter party chooses a service provider that represents the best deal and rejects all other offers. After that interaction blocks termed contracting, verifying, monitorability negotiation, and enactment follow that contain interaction sequences similar to Figure 3.

**Pattern 3 (Semi-Dynamic Assignment)**
***Problem:*** An organization is involved in tight supply-chain integration where parts of an overall service are Sourced. While it is important that a service is Sourced, it is not clear which organization can offer guaranteed collaboration.
***Description:*** Before the setup time of a service exchange begins, the number of organizations that are willing to collaborate is predetermined. When the service to be Sourced is externalized to a broker, only the collaboration candidates may evaluate and engage in a bidding for the service. Any other organization that is not a predetermined candidate, is excluded. The service-externalizing organization chooses the offer and rejects all other.
***Forces:*** Concerning the tight integration of service providers, the forces of Pattern 1 apply. With respect to using a publicly visible process broker and managing a bidding procedure, the forces of Pattern 2 are applicable. The requirement of a notification system is specific for this semi-dynamic assignment pattern. Such a notification system is activated when the template process is committed to the public process broker

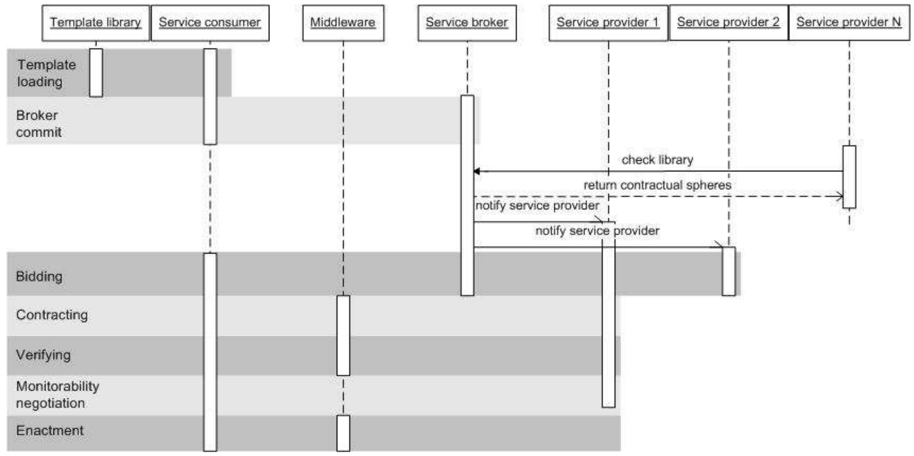


**Fig. 5.** An interaction-sequence example for semi-dynamic service assignment

by the service consumer. Since the template contains the set of predetermined service providers, the broker should notify those providers so that they may engage in the bidding.

***Examples:***

- An insurance company evaluates claims resulting from car crashes. A part of the in-house process is concerned with evaluating the damage. However, the insurance company doesn't consider such an assessment its core business and therefore has external companies providing such assessment services. Those companies form a pool of competing service providers that are shared between a number of different insurance companies. Thus, it is never clear which company is available for performing a car-damage assessment.
- An abstract example geared towards the creation of an eSourcing example is given in Figure 5. First, a service consumer loads a template that contains definitions referring to a set of tightly integrated service providers. After that the consumer commits its sphere to a publicly available service broker. A *service provider N* is browsing the service broker and receives several processes in return. However, processes where the predetermined service-provider candidates are defined are not returned. Instead the service broker sends out notifications to those providers defined in the process. The notified providers engage in a bidding procedure after which a contracting phase follows with the providers that were chosen by the service consumer. Afterwards the service consumer requests a verification of the preliminarily defined eSourcing configuration. If the verification concludes with an approval, the monitoring of process nodes is negotiated. Eventually the service consumer requests from the middleware to enact the ready eSourcing configuration.

After specifying assignment patterns, the following section comprises patterns deduced from the direction dimension of Figure 2. Just in the case of the assignment dimension, each direction value is translated into one pattern specification.

## 6 Direction-Dimension Patterns

**Pattern 4 (Out-Sourcing)**
***Problem:*** A company has a part of its in-house business process that is detrimental to its overall competitive advantage. Thus, the company knows that a subcontractor could carry out the process in a better way.
***Description:*** A part of an organization's in-house process that should be carried out by a third party, is demarcated into a subprocess. Next, the subprocess is taken over by an organization that agrees with offering the service. In the domain of the service provider further refinement of the service may take place that remains opaque to the service consumer. The subprocess in the domain of the service consumer and the refined process in the domain of the provider are linked with each other and the service consumer starts with enactment of the created inter-organizational configuration.
***Forces:*** Since the assignment patterns of Subsection 5 are variants of the out-sourcing pattern, all forces mentioned in the assignment patterns also apply for this direction pattern.
***Example:*** All examples mentioned for the assignment patterns of Subsection 5 are variants of the out-sourcing direction pattern.

**Pattern 5 (In-Sourcing)**
***Problem:*** A company has serendipitously discovered a process innovation that poses a competitive advantage. However, it is initially not clear which companies the potential purchasers of the process innovation are.
***Description:*** The service provider sets up a process in its own domain and subsequently exposes a subset of the process details publicly. Compared to the exposed version, the internal process contains additional refinement steps that remain opaque. Next, an interested service consumer adopts the exposed process and integrates it in the in-house process.
***Forces:*** The service consumer may find it impossible to integrate the exposed process of the provider. In that case extra negotiation about the exposed process content needs to unfold between the collaborating parties. The situation can occur that potential consumers are not aware of the offered service and as a result no collaboration comes into existence.
***Examples:***

– A chemical-engineering company conducts research into special lubricants for mechanical-engineering companies. However, it turns out this lubricant also offers the quality of being a very effective stain remover in textiles. Therefore, the company decides to change the packaging and labelling of the lubricant so that it can be sold to firms that are operating as cleaners of textiles. Thus, some special retailers are expected to be interested in purchasing the lubricant.

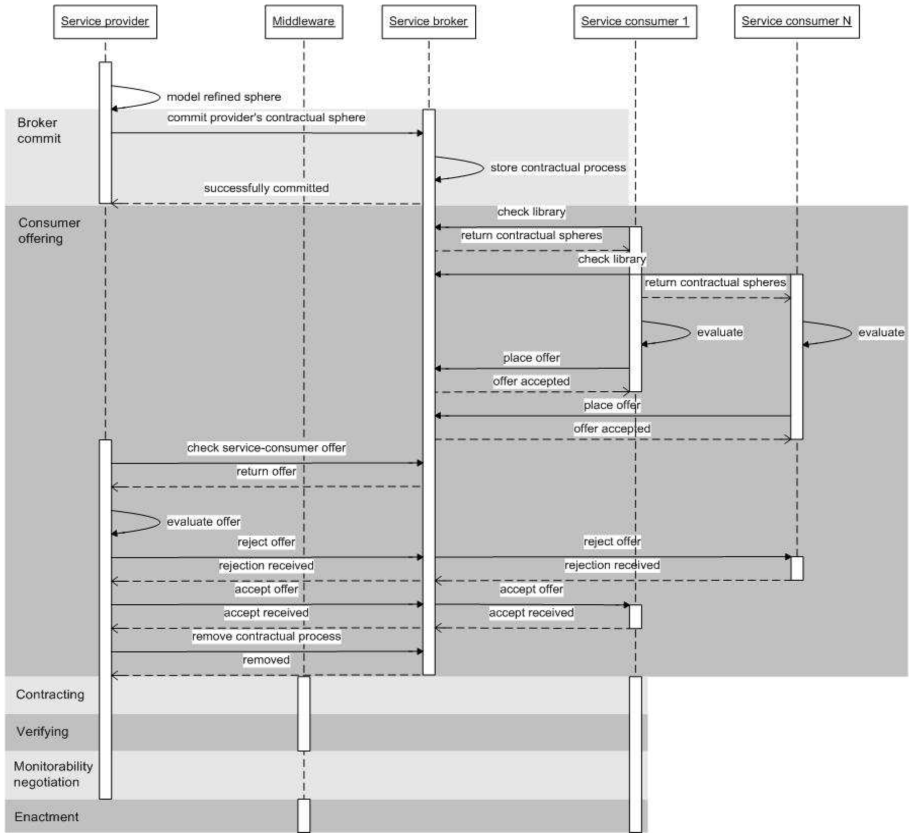


**Fig. 6.** An interaction-sequence example for in-sourcing

– In Figure 6 an in-sourcing example is depicted that focusses on the eSourcing concept of Section 3. First, a service provider models a refined sphere internally. A subset of this refined sphere is turned into a provider's contractual sphere and committed to a publicly available service broker. Next, interested service consumers check the committed contractual process and respond with placing offers. The service provider evaluates the offers, accepts the best, and rejects all other offers from the remaining consumers. Once two collaborating parties are allocated, the remainder of Figure 6 is described by interaction blocks that are detailed in the assignment patterns of Section 5. Thus, the collaborating parties engage in concrete contracting, followed by a verification phase. After negotiating which process steps need to be monitored, the resulting eSourcing configuration is enacted.

**Pattern 6 (External-to-Internal)**
***Problem:*** Two parties want to establish a B2B supply chain without existing constraints resulting from historically grown business processes in their own domains.

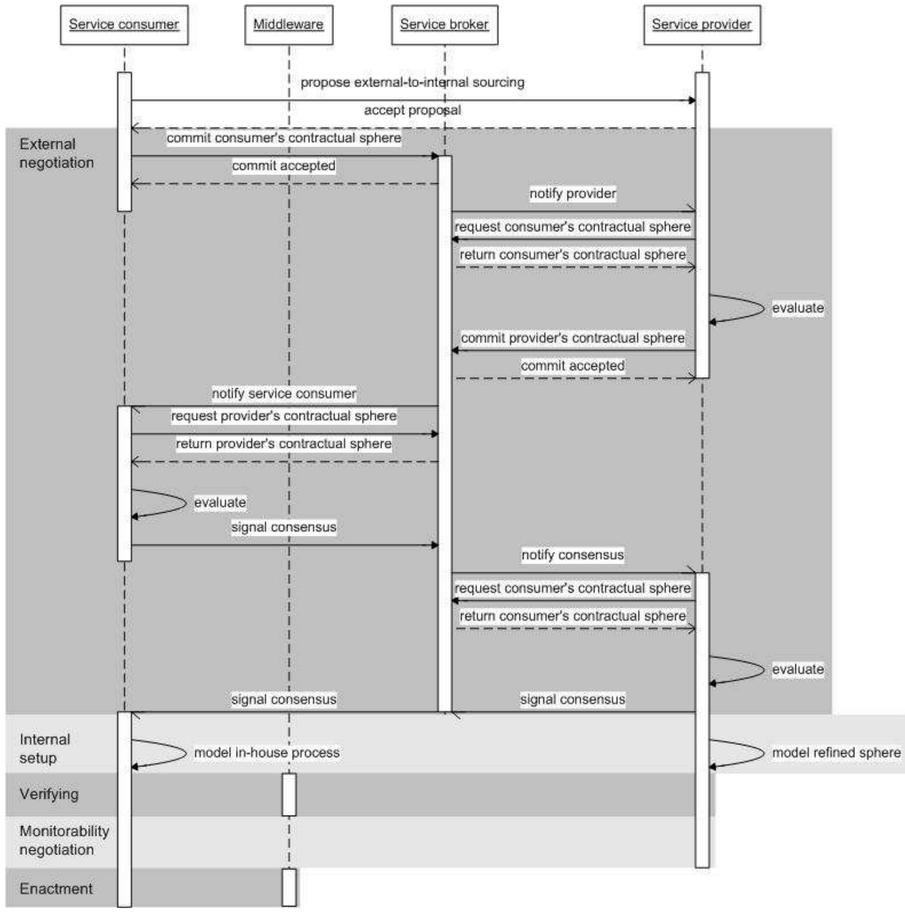


**Fig. 7.** An interaction-sequence example for external-to-internal sourcing

***Description:*** The service consuming and providing organizations start with negotiating process properties on an external level. When they have reached consensus, both parties take over the publicly agreed upon process for their internal domains. In the domain of the service consumer the adopted process becomes a subnet of a bigger in-house process, while in the service provider's domain further refining process steps are inserted.

***Forces:*** A service broker can not be employed if there is no published service offer available in the beginning. Thus, it might be a problem to find a collaborating party. Furthermore, integrating the externally agreed process might fail because of an inadequacy of internal resources, e.g., the organizational structure turns out to mismatch, no employees have the appropriate skills for carrying out certain tasks, required production resources might be lacking, and so on.

***Examples:***

– An IT-company intends to develop an innovative software package for a leading telecom enterprize. For this purpose personnel with state-of-the-art skills is hired. Still, a part of this new software package for the telecom enterprize can't be developed internally. Instead, a subcontractor that has been founded very recently is used for this externalized part of the overall software.
– In Figure 7 an example of external-to-internal sourcing is depicted. Since there is no initial contractual process available to publish in a public service broker, collaborating parties need to find each other in a different way. The example in Figure 7 suggests that a service consumer directly contacts a suitable provider to engage in contractual negotiation on an external level. Thus, the collaborating parties propose to each other their respective contractual spheres until a consensus is reached. Next, the contractual spheres are backed by processes in the domains of the service consumer and provide. The first party integrates the sphere in the in-house process, and the latter party extends the contractual sphere to a refined process. The following phases are depicted by interaction blocks that have been detailed in previous pattern examples. Thus, the service consumer initiates a verification phase that is followed by a negotiation phase for determining which process steps need to be monitored. Finally, the resulting eSourcing configuration is enacted.

**Pattern 7 (Internal-to-External)**
***Problem:*** Collaborating parties have historically grown, stabile business processes in their respective domains. These processes need to be harmonized on an external level across organizational boundaries.

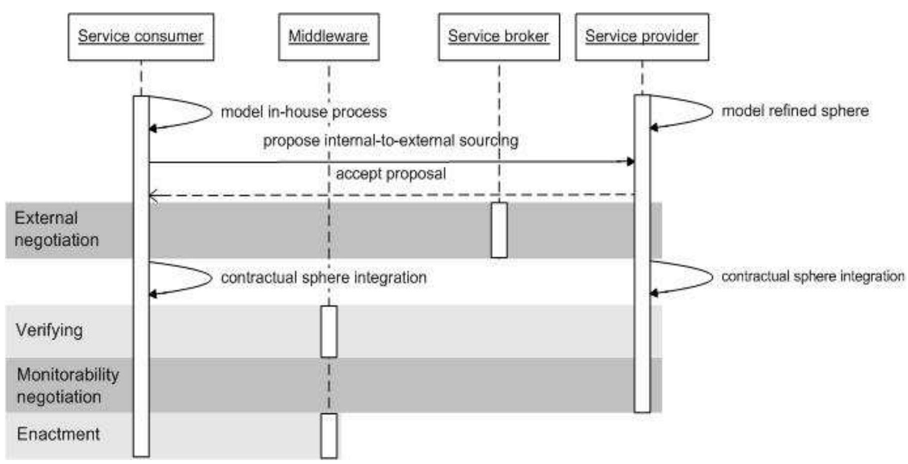


**Fig. 8.** An interaction-sequence example for internal-to-external sourcing

***Description:*** Both the service consumer and provider have established business processes in their domains. The consuming organization considers a part of its in-house

process to not be core business. On an external level, the consumer and provider engage in negotiating a consensus process that accommodates their already existent respective internal processes.
***Forces:*** The internal processes of the service consumer and provider may proof to differ so extensively from each other that it is not possible to find a consensus on an external level.
***Examples:***

- A market-share dominating OEM is a producer of high-precision track maintenance machines. The OEM feels no necessity to change the production and business processes. The suppliers of the OEM are equally not prepared to change their historically grown production and business processes. After investing in information technology and techniques the OEM and its suppliers begin to establish business collaborations electronically.
- An abstract example is depicted in Figure 8 as a sequence diagram. First the collaborating parties model their internal processes before they engage with each other proposing to commence with contractual negotiations on an external level. The external negotiations are depicted as an interaction block that is detailed in Figure 7. Once a consensus is reached, the contractual processes are integrated in the processes of the respective domains. The remaining interaction shows several interaction blocks for verification, monitorability negotiations, and the eventual enactment of the ready eSourcing configuration.

After specifying patterns for the direction- and assignment-interaction dimensions, the following section uses these patterns for deducing a reference architecture that is guiding for the design and development of applications that support the setup of inter-organizational business processes.

## 7 An Architecture for Supporting Interaction Patterns

Software development consists of three main phases, i.e., the analysis, design, and implementation phases [26]. In the previous section the pattern-based exploration represents an analysis that provides the necessary input for the design phase in the development of an e-collaboration-setup system. This section presents a conceptual reference architecture for collaboration setup that is elaborated upon based on the pattern specifications of the previous section.

A reference architecture for the domain of electronic interaction between collaborating business parties, provides the major design principles and specifies the functionalities that must be delivered by such an e-collaboration-setup system. Thus, a reference architecture serves as a starting point for software developers who are occupied with designing and implementing an information system for supporting the automated setup of business collaboration. In Subsection 2.2 several examples of reference architectures are mentioned.

Conceptual architectures (also known as logical architectures) facilitate the understanding of the interactions between components and the functionalities provided by the system, and are consequently a good technique for the definition of reference architectures. The proposed reference architecture of this section is aimed to serve as a foundation in the design and development of e-collaboration setup systems.

The following subsections first explain the design approach followed by the three detailing level of the reference architecture. The last subsection discusses how the reference architecture supports the eSourcing examples of Section 5 and Section 6 that are provided in the interaction pattern specifications.

### 7.1 Design Approach

The eSourcing-setup reference architecture (eSRA) is designed in accordance with the principle of functional decomposition of a system. This decomposition is also known as ”separation operation” and based on the part-whole principle [15]. Thus, at each level of the eSRA, the identified components provide functionalities that don't overlap with the remaining components that are located at the same level. To achieve completeness, the eSRA is designed in a top-down way. As a result the eSRA's functionalities and interactions are addressed in a step-by-step manner from a high level of abstraction on the top level, the level of detail gradually increasing on lower levels.

In Section 3, it is demonstrated how the concept of eSourcing is using three levels [19] for realizing the interoperability of business processes and their supporting heterogenous technologies that belong to the respective domains of collaborating parties. Thus, the architecture presented below also uses three levels. On the external level, components are located that enable interaction and exchanges with external parties while the conceptual level is focussing on modelling business rules and processes. These rules and processes are on the one hand projected to the external level but also mapped to the internal level where legacy systems are coordinated as web services. Thus, the positioning of the eSRA in the three-level framework provides the basis for achieving interoperability between collaborating parties. The next two subsections describe the eSRA's architecture in further detail.

### 7.2 First Detail-Layer of an eSourcing-Setup Reference Architecture

The depiction of Figure 9 shows the highest abstraction layer of the eSRA. In the figure two collaborating parties show the same set of components distributed across an external, conceptual, and internal level. The grey shaded boxes represent components and arcs between the components depict exchanges between the components.

On the external level the *Sourcing middleware* is replicated on the respective external levels of collaborating parties. This component is the main enabler of interoperability and direct information exchange exists between the eSourcing middleware of each collaborating party to synchronize the respective components. Between the collaborating parties a component is located termed *trusted third party* that exchanges information with the eSourcing middleware. A trusted third party is necessary for several reasons. Firstly, collaborating parties expose service demands or service offerings to the trusted third party for public evaluation. Secondly, the trusted third party performs verification services and checks quality features of eSourcing configurations before enactment. If collaborating parties perform verifications and checks of eSourcing configurations

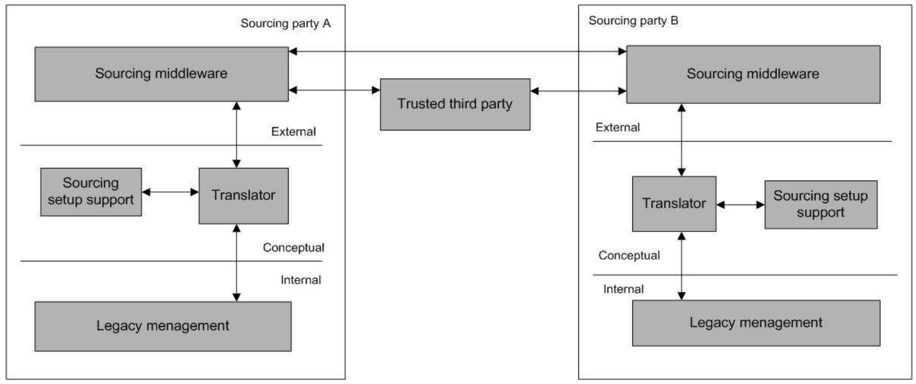


**Fig. 9.** Overall eSourcing enactment architecture

themselves, they would need to reveal competitive secrets to each other, which is undesirable.
The conceptual level of Figure 9, depicts two components, namely the translators and the eSourcing setup support. The *translator* component exchanges information between the components of the external and internal level. The *Sourcing setup support* contains among other functionality tools for modelling business rules and processes. Finally, the internal level depicts a *legacy management* component that interfaces on the one hand with the translator component of the conceptual level and the legacy system of a collaborating party. The legacy systems are wrapped in web services that are coordinated by local workflow engines and business rules engines.

### 7.3 Second Detail-Layer of the Reference Architecture

In this subsection, each grey-shaded component of the reference architecture depicted in Figure 9, is further refined. The first refinement of Figure 10 covers all components that are located on the external level, namely the Sourcing middleware, and the trusted third party. This focus is visualized by grey shading. In contrast, the translator component is not grey shaded as it is refined in a later figure. In all figures of this subsection the refined components of focus are depicted with their exchanges to bordering components.

In Figure 10 the Sourcing middleware of one collaborating party is depicted. The Sourcing counterpart contains the same second layer components, however, for sake of brevity only the relationship between the coordination-interface components is depicted. Furthermore, Figure 10 shows the trusted-third-party component as a white box and its relationship with the Sourcing-middleware component. A dashed arc between the trusted-third-party component and the half-opened Sourcing-middleware symbolizes that equal information exchanges exist compared with the fully opened Sourcing-middleware and the trusted-third-party.

**Trusted-Third-Party Refinement**

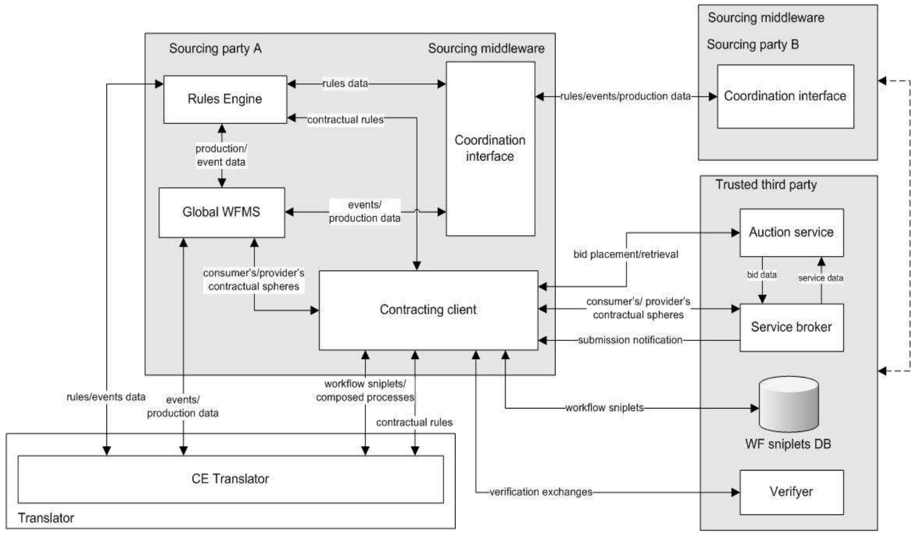


**Fig. 10.** External-level collaboration

In the latter component several refining components are contained. The contracting client component provides support for the management of an e-contracting process. Concretely, the contracting client semi-automatically assembles services by using workflow sniplets that are stored in a corresponding database of the trusted third party. Depending on whether a collaborating party slips into the role of a service consumer or service provider, the contracting client submits or retrieves contractual spheres from a service broker.

If a submitted service contains the definition of a concerned party, a submission notification is sent out from the service broker. If several parties are interested in the same service, a bid can be placed with the auction service. The latter component relates the bid data with services stored in the service broker. Finally, when an eSourcing configuration (see Section 3) is established, the collaborating parties send their in-house processes and refined spheres to a verifier component for testing the correct termination, i.e., the soundness [4, 22], of the overall eSourcing configuration. The verification results are returned to the collaborating parties.

**Sourcing-Middleware Refinement**

When an eSourcing configuration is established, the contracting client distributes the business rules and the processes contained in the contract to the global rules engine and the global workflow management system (WFMS) respectively. In order to synchronize the global WFMSs and rules engines in the Sourcing-middleware components of other collaborating parties, events-, production-, and rules data is communicated via a

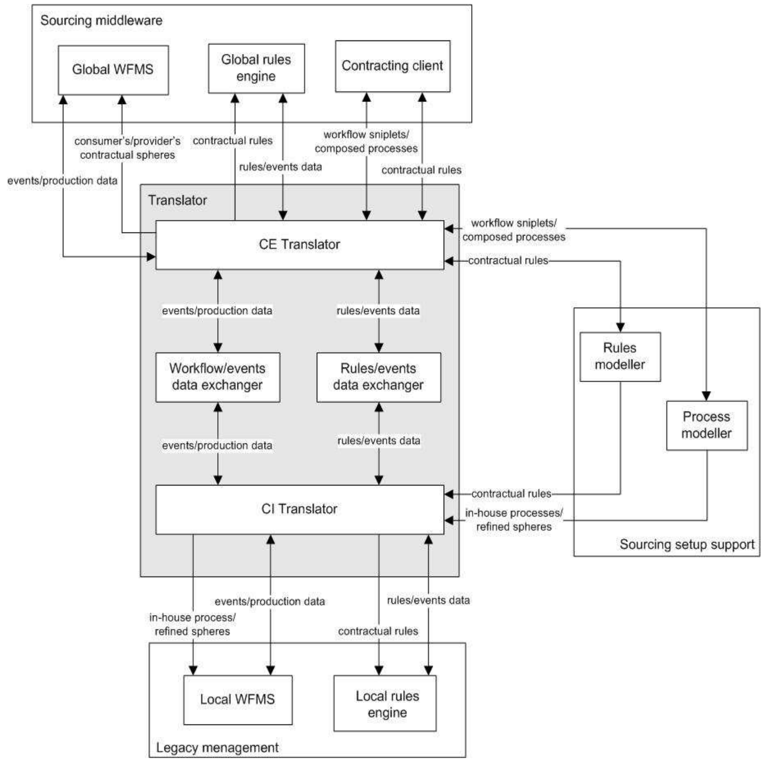


**Fig. 11.** Translating between external and internal level

coordination interface. Furthermore, the global WFMS and rules engine also exchange production-, and event data with each other.

Finally, the contracting client sends workflow sniplets and composed processes and contractual rules to the translator. Also the global WFMS and rules engine send date to the translator component that is depicted as a white box in Figure 8.1. The translator contains two main translator components for transferring data between the external, conceptual and internal level.

### Translator Refinement

The CE translator component translates data from the conceptual to the external level and vice versa. The component is connected with the rules and process modelers of the Sourcing-setup-support component. The relationships between the CE translator and components contained in the Sourcing middleware is explained above. Two components exchange data between the CE translator and and CI translator, namely the worklfow/events data exchanger and the rules/events data exchanger. Those data exchangers contain information about where data needs to be routed to. For example, several instances of WFMSs and rules engines on the external and internal level may enact several instances of different Sourcing configurations. Furthermore, on the internal level several web services wrap legacy systems to which exchanged data needs to be routed to.

The CI translator component translates data between components of the conceptual and internal levels. From the data-exchanger components, events-, rules-, and production data are translated bi-directionally to the local WFMS and rules engine on the internal level. Furthermore, the CI translator receives contractual rules from the rules modeler and in-house processes and refined spheres from the process modeler. They are translated to the local WFMS and rules engine on the internal level.

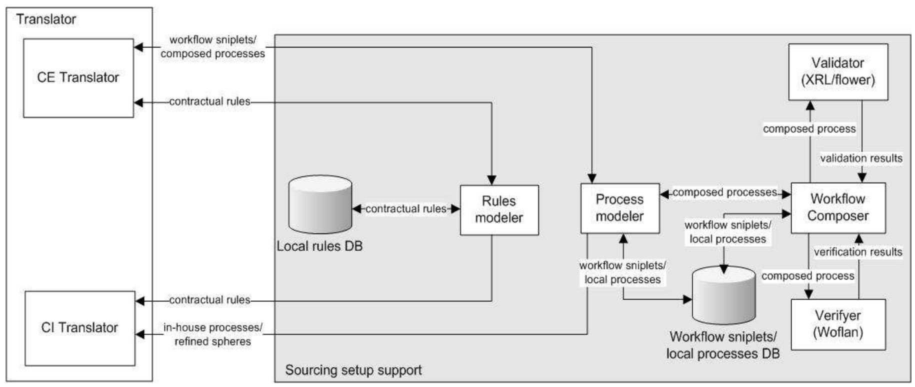


**Fig. 12.** Setup functionality

**Sourcing-Setup-Support Refinement**

In Figure 12, the Sourcing-setup-support component is located on the conceptual level. The component has two core functions, namely modelling contract-rule constructs and processes, and composing workflows that are on the one hand evaluated and on the other hand verified for correct termination. Contract rules are business rules relevant to the contracting relations between the contracting parties. In the existing literature four basic types of business rules are discussed, i.e., integrity rules, derivation rules, reaction rules, and deontic assignments [35, 36]. Thus, the rules modeler and the process modeler are responsible for the first function. The second function is related to the workflow composition component.

For composition, workflow sniplets or local processes are taken from a dedicated database, which are supplied by the process modeler. A composed workflow is either an in-house process or a refined sphere and it is checked in two ways. Firstly, with respect to control flow, correct termination by a verifyer tool for which the process first needs to be mapped to a place/transition net. In Figure 12 the tool Woflan [34] is suggested for verifying correct termination. If the net is a WF-net, Woflan checks for structural conflicts, i.e., deadlocks or lack of synchronization. Thus, if the WF-net is verified to terminate correctly, it conforms to the notion of soundness [4]. Secondly, the in-house process or refined sphere needs to be verified for other conflicts, e.g., data-flow or resource.

Although it is desirable to have verification tools for several workflow related perspectives, e.g., data-flow and resource, it is essential to validate the in-house process and refined spheres of an eSourcing configuration with an additional tool. Among other aspects, such a validation is meaningful for testing the technological infrastructure required for workflow enactment, e.g., the correct functioning of web services that are orchestrated by the processes. In the Sourcing-setup-support component of Figure 12 XRL/flower [27] is depicted as a validation tool. XRL/flower uses XML technology and is implemented in Java on top of the Petri-net Kernel PNK [23]. Standard XML tools can be deployed to parse, check, and handle XRL documents. The XRL enactment application is complemented with a web server, allowing actors to interact with the system through the internet.

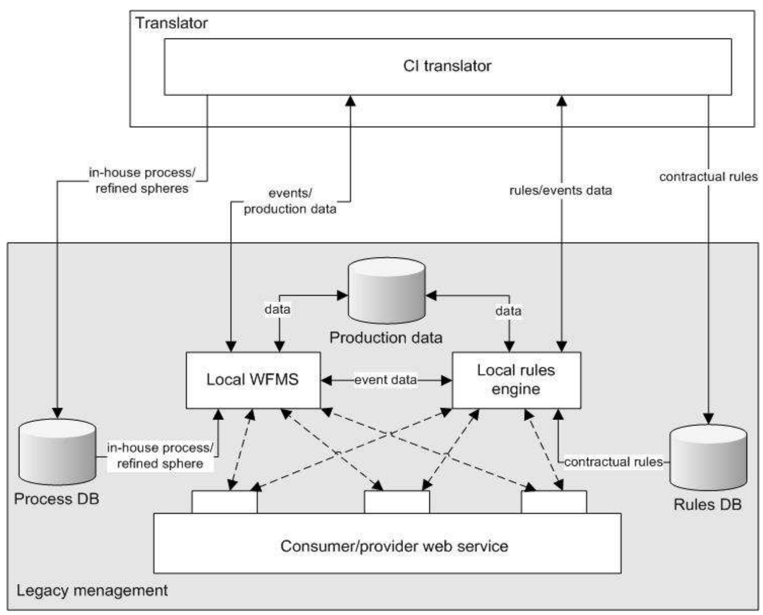


**Fig. 13.** Connecting to internal legacy systems

### Legacy-Management Refinement

Finally, Figure 13 visualizes a second-layer refinement of the legacy-management component. In it a local WMFS and rules engine constitute the core components. These components are exchanging data between each other and are instrumental for coordinating legacy systems that are wrapped in web services. The business rules and processes that are enacted by the WFMS and rules engine are translated down to the internal level by the CI translator. For enactment, the local WFMS and rules engine use a production database. Furthermore, to coordinate the enactment on an internal level and external level, the local WFMS and rules engine communicate events-, rules-, and production data bi-directionally.

After proposing an architecture that enables the setup of inter-organizational collaboration, the following section illustrates how a combination of two chosen interaction patterns is supported by the components of the reference architecture.

## 7.4 Third Detail-Layer of the Reference Architecture

In this subsection the dark-grey shaded components of Subsection 7.3 are further detailed according to the principles of functional decomposition. First the CE translator and CI translator are refined in Figure 14 and Figure 16.

### CE-Translator Refinement

The refinement of the CE translator depicts a CE projector component that is performing projections between the external and conceptual levels. To perform that function, the CE projector uses a projection rules database.

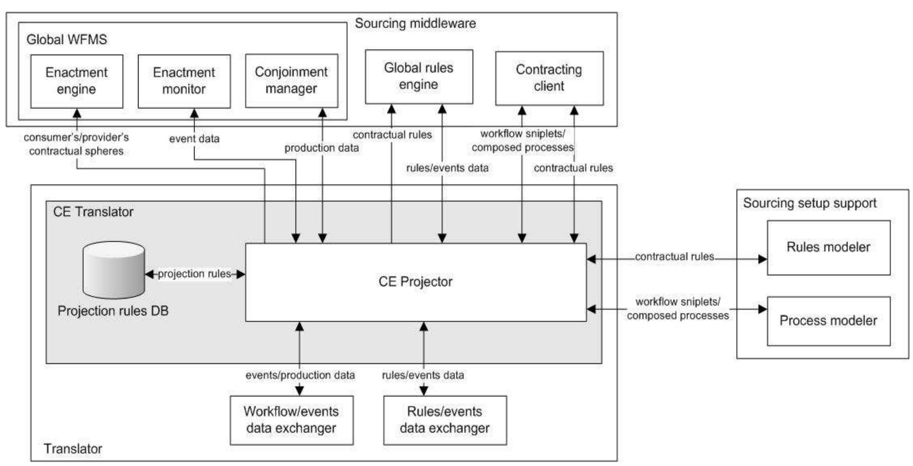


**Fig. 14.** The CE translator in detail

Figure 14 shows several bi-directional arcs to the projector of which subsets are related to each other. The rules- and process-modeler components exchange contractual rules, workflow sniplets, and composed processes via the CE projector with the contracting client on the external level. The global rules engine receives contractual rules from the

CE projector through which rules- and events data is exchanged via the rules- and events data exchanger down to the local rules engine on the internal level. Figure 14 depicts a detailed exchange between the CE projector and components of the global WFMS. Firstly, the enactment engine receives contractual spheres from the service consumer or provider respectively. During enactment, an exchange occurs with the enactment monitor and the conjoinment monitor, which is explained below and depicted in Figure 15. The latter two components exchange events- and production data via the CE projector and the workflow/events data exchanger down to the local enactment monitor and conjoinment monitor that are located on the internal level of the reference architecture.

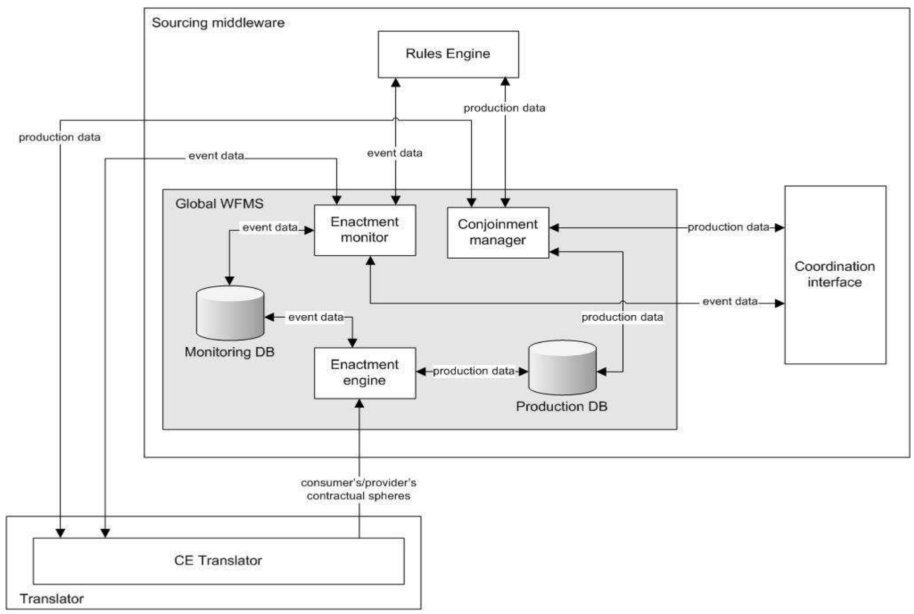


**Fig. 15.** The global WFMS in detail

## Global-WFMS Refinement

In Figure 15 the global WFMS component of the Sourcing middleware that is situated on the external layer, is depicted as a refinement. It shows an enactment engine that enacts the consumer's or provider's consumer spheres that are delivered from the CE translator. Event and production data is created during enactment and also needed for enactment and therefore stored and retrieved from dedicated databases. In order to support the concept of Sourcing, Figure 15 shows an enactment-monitor component and conjoinment-monitor component. These components are important to support Sourcing patterns that are formulated in [29]. Concretely, the enactment monitor is responsible for aligning the enactment progress of internal level processes of the collaborating parties. Likewise, the conjoinment-manager component supports the conjoinment patterns specified in [29]. Both, the enactment monitor and the conjoinment manager exchange production- and event data with components in the domain of the collaborating party via the coordination interface. Furthermore, production- and event data is communicated to the internal level via the CE translator to coordinate local components.

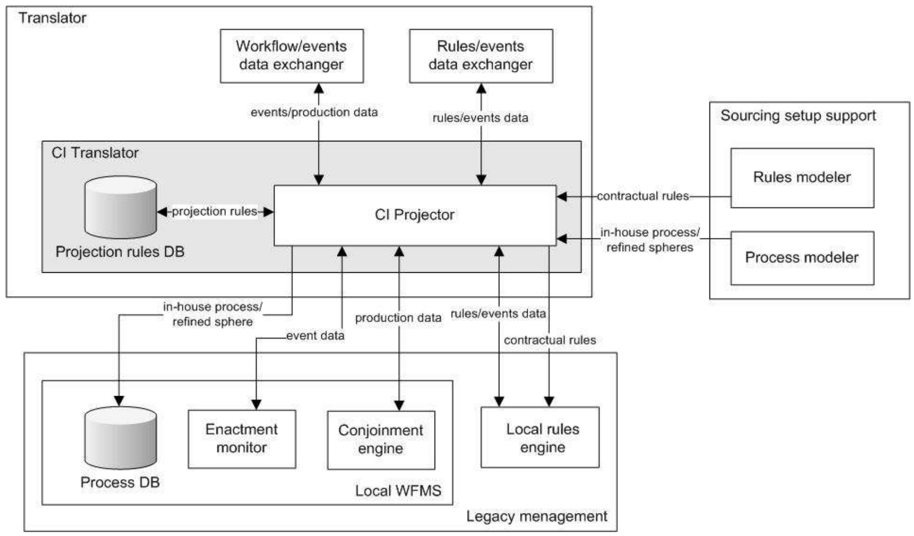


**Fig. 16.** The CI translator in detail

### CI-Translator Refinement

The refinement of the CI translator in Figure 16 depicts a similar setup as for the CE translator. However, the information exchange to neighboring components differs. The CI translator contains a CI-projector component that projects information between the conceptual and internal level. To do so, a projection-rules database is exchanging rules with the CI projector. With respect to information exchange between the CI translator and its environment, different subsets of arcs depicted in Figure 16 are related to each other.

The CI projector receives contractual rules from the rules modeler, and in-house processes and refined spheres from the process modeler. The contractual rules are projected to the local rules engine of the internal level. Furthermore, the in-house processes and refined spheres are also projected to the internal level where a process database stores them until the local WFMS loads the processes for enactment. To coordinate the local WFMS and rules engine with corresponding components on the external level, the CI-projector transfers production-, rules-, and events data between the internal and external levels of the reference architecture.

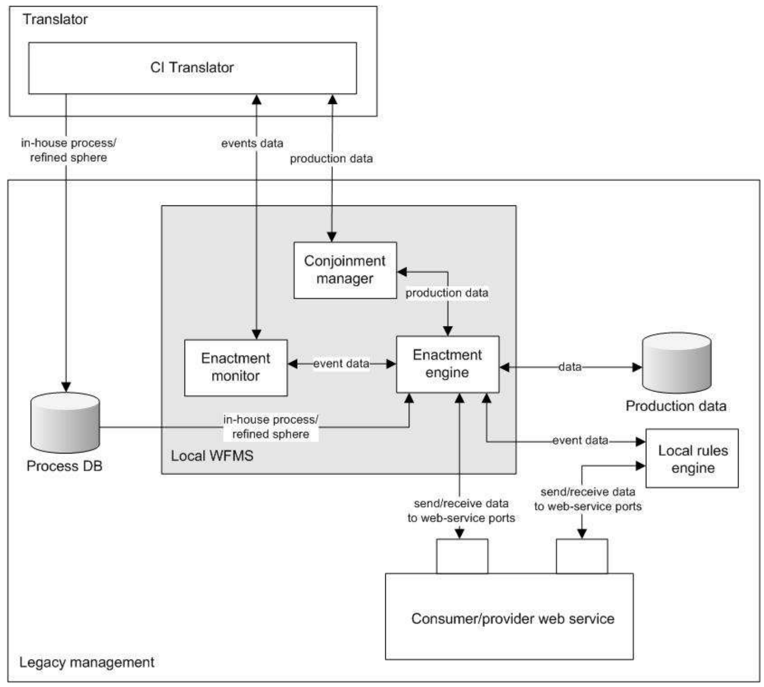


**Fig. 17.** The local WFMS in detail

### Local-WFMS Refinement

The internal level refinement of Figure 17 shows a setup that is comparable to the global WFMS of Figure 15. The local WFMS contains an enactment engine that receives in-house processes and refined spheres from the process database. Production data that is produced and consumed during process enactment, is exchanged with the production-data database. Event data is exchanged with the local rules engine that carries out contractual rules. Furthermore, the enactment engine exchanges data with ports of web services that wrap legacy systems. To coordinate the local enactment progress with the external level, production data and event data are exchanged with the conjoinment manager and the enactment monitor respectively. The latter two components exchange events- and production data via the CI translator with the equally named components located on the external level.

### Service-Broker Refinement

The service-broker refinement within the trusted third party of Figure 18 reveals a service-library database that receives and sends contractual spheres of service consumers and providers via the template search engine. The latter component exchanges

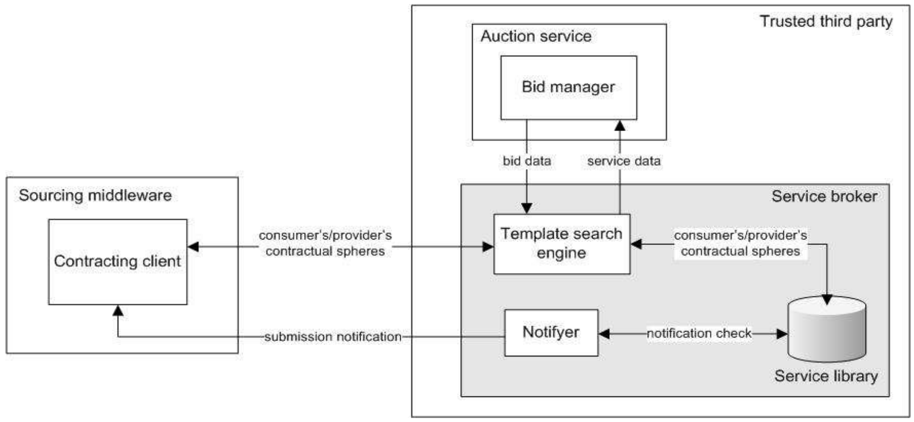


**Fig. 18.** The service broker in detail

contractual spheres with the contracting client of the Sourcing middleware that is located on the external level of the collaborating parties. Furthermore, the template search engine exchanges data with the bid-manager component of the auction service. The notifyer component checks contractual spheres that are stored in the service library for data about a collaborating party that needs to be informed. If such facts are defined, the notifyer informs the specified contracting client of the respective parties about the submission of the contractual sphere. Consequently, informed parties check the contractual sphere and either engage in a bidding procedure or commit to the contractual sphere by instantaneously responding with committing a contractual sphere of equal content to the trusted third party.

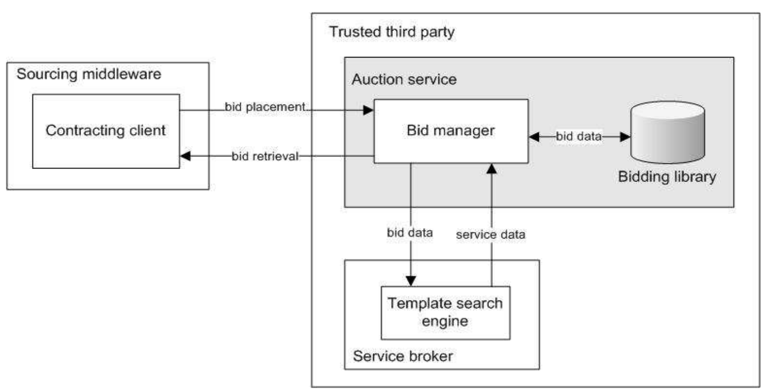


**Fig. 19.** The auction service in detail

### Auction-Service Refinement

The refined auction service of the trusted third party is depicted in 19. In the auction service component the contained bidding library stores bids that are committed and retrieved by a bid manager. This component is communicating with the contracting-client component that places and retrieves bids from the bidding library. As described earlier, the bid manager is exchanging bid- and service data with the template-search-engine component of the service broker.

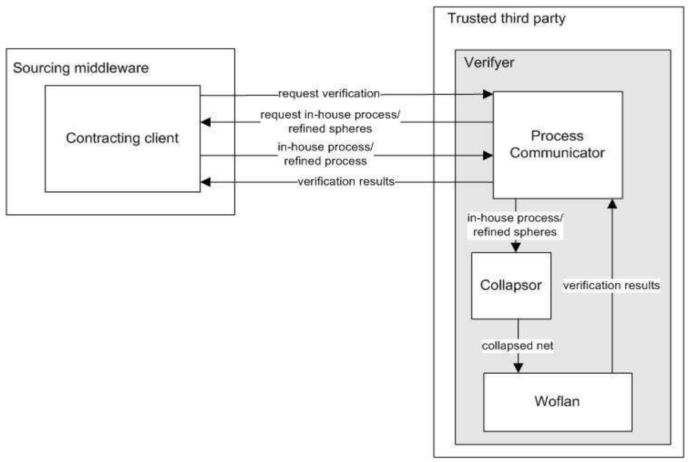


**Fig. 20.** The verifier in detail

### Verifyer Refinement

In Figure 20 the last component refinement of the trusted third party is visualizes, namely the verifyer. In the refinement, a process-communicator component receives a request from the contracting client to perform a verification of a created Sourcing configuration. The process communicator requests the in-house process and refined spheres of all collaborating parties. Next, the collected in-house process and the refined spheres are delivered to the collapsor component where the Sourcing configuration is transformed into a ”collapsed” net. The collapsor component replaces the consumer spheres of the in-house process with the corresponding refined spheres and delivers the collapsed net to the Woflan [34] component.

Comparable to the Sourcing-setup-support component depicted in Figure 12, Woflan is used to verify the correct termination of the collapsed net and therefor the correct termination of the Sourcing configuration from a control-flow perspective. Such verification is necessary because processes that terminate correctly on their own may contain structural problems, e.g, deadlocks, when they are connected to other processes.

However, besides verifying correct termination, Woflan can also be used to check other properties such as projection inheritance [5] in correlation to its consumer sphere.

## 8 An Interaction Case Study

The case study is input for the development of a proof-of-concept prototype in the CrossWork [1] project that focusses on the automobile industry. 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 setting up 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. The case study applies the *internal-to-external* pattern from Section 6 of the direction dimension that is explained in Section 4 on an application infrastructure that is based on the reference architecture of Section 7. The *internal-to-external* pattern is combined with the *static* assignment pattern of Section 5, since the OEM has one predetermined supplier to collaborate with. In the following subsections it is stressed which parts of eSRA from Section 7 are adopted in CrossWork's proof-of-concept prototype to support the case study of this section.

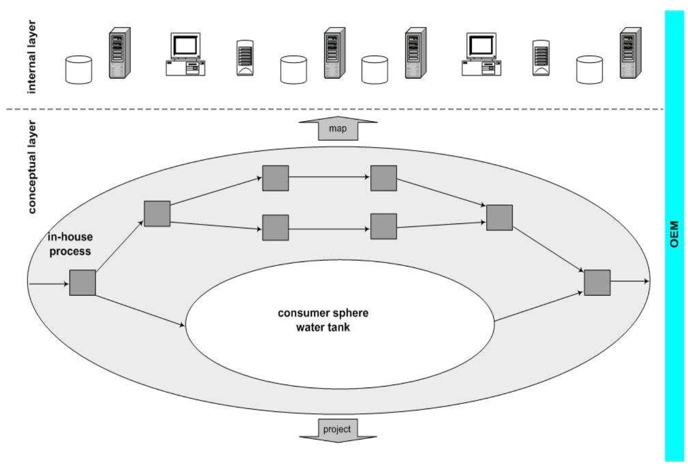


**Fig. 21.** OEM in-house process

Referring to the pattern example of Figure 8, the interaction assumes the existence of internal processes in the domains of the service consumer and service provider. The service consumer proposes the creation of an eSourcing configuration. The negotiations about creating a common external-level process are followed by an integration of the contractual spheres in the respective domains. Next, the eSourcing configuration is verified and the extent of monitoring is negotiated. Finally, the enactment of the completed eSourcing configuration goes ahead.

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.

## 8.1 Conceptual-Level Setup

The OEM possesses a historically grown process for producing a truck. The starting point for this process is the truck order from a customer and the ending point is the shipping of the constructed truck that adheres to the customer's requirements. Likewise, service providers of the automobile cluster posses internal processes that have grown historically and that are not easily changeable.

Figure 21 depicts the in-house process setup for the OEM. On top, the internal level shows legacy systems that support the OEM's business process. On the conceptual level of Figure 21, the in-house process is depicted for producing a truck. Contained is a consumer sphere that delimits a subnet for eSourcing a water tank from service providers. Different to the eSourcing example of Figure 1, the in-house process of Figure 22 doesn't use Petri-net formalisms. In CrossWork the proof-of-concept prototype equally doesn't use Petri-nets for the creation of eSourcing configurations. However, high-level eSourcing configurations are mapped to a Petri-net representation for the purpose of verifying control-flow properties, e.g., soundness.

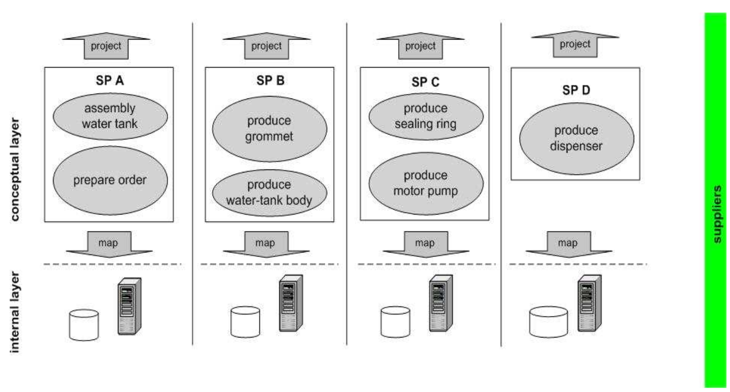


**Fig. 22.** Providers in-house processes

The service providers are depicted in Figure 22 with their internal setup. Four providers are contained in the figure that are part of the automobile cluster with each containing processes that create parts of a water tank for the OEM. According to Figure 22 a service provider may be capable of contributing several processes for producing parts of a water tank. Comparable to the OEM, all service providers have legacy systems on their respective internal levels that need to be integrated in the eSourcing configuration. To do so, the OEM and service providers map their conceptual-level processes on the internal level.

To create the processes depicted in Figure 21 and Figure 22, all parties use their respective process-modeler components of Figure 12 and the workflow-composer component that are part of the Sourcing-setup-support component. Although the case study focus on the business processes, it is assumed that business rules are created with the rules-modeler example that is equally depicted in Figure 12. Both processes and rules are stored in their respective databases. To ensure good quality of the conceptual-level processes, the collaborating parties need to utilize the validation- and the verification components. Furthermore, each party needs to ensure that conceptual-level processes are mapped to their internal-level legacy systems. Figure depicts that processes and rules are delivered to the internal level via the CI translator component of Figure 8.1. Furthermore, in Figure 13 the local WFMS and rules engine are the recipients that consequently control web-service wrapped legacy systems.

### 8.2 External-Level Setup

When all the conceptual-level processes are in place in the respective domains, the collaborating parties need to harmonize their respective processes on the external level. In this case study, the OEM proposes to service provider *A* from the automobile cluster the initiation of negotiations for external-level harmonization. The latter party accepts and the OEM and the automobile cluster engage in external-level negotiations.

The OEM starts by filling the gap of the consumer sphere depicted in Figure 21 with using the rules-modeler and the process-modeler components of the Sourcing-setup-support component of Figure 12. Next, the consumer sphere for eSourcing a water tank from the automobile cluster is transferred via the CE translator of Figure 8.1 to the contracting client of the external level. The latter component exposes the OEM's contractual sphere to the service broker of Figure 10 that is part of the trusted third party. Since service provider *A* is the unique contact point for the OEM, the first party has the task to find and organize other service providers for the fulfillment of the externally agreed upon service provision.

Service-provider *A* is notified by the service broker of Figure 10 and requests the OEM's contractual sphere for a check. In Figure 23 the OEM's consumer sphere is depicted that is entirely projected to the external level. The consumer sphere shows several nested consumer spheres for the production of the water tank that is designed to be a fitting subnet within the overall in-house process. First, the order needs to be prepared by service provider *A* who has to find the appropriate automobile-cluster members for carrying out the required nested spheres.

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*. Finally

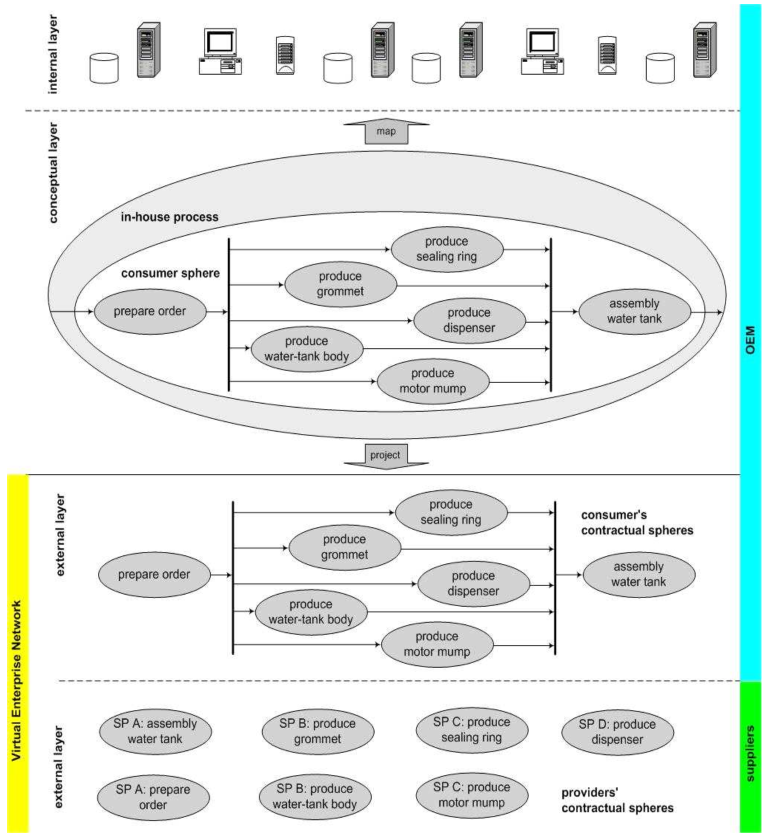


**Fig. 23.** External-level contractual processes

provider *A* takes over the services of preparing the order and assembling the produced parts to one water tank that can be shipped to the OEM. The external level of Figure 22 shows how the OEM wants to have the nested spheres controlled. After preparing the order, the spheres related to producing water-tank parts are carried out in parallel and finally the assembly is carried out.

On the service provider's side of the external level in Figure 22 a consensus is created as equal nested spheres are projected. Thus, an overall consensus is given at the end of negotiation phase of the interaction between the OEM and the automobile cluster. Optionally an integration of the contractual spheres of the external level is required on the conceptual level. Such a step is necessary if the negotiation phase has resulted in major external-level deviations compared to what is originally projected from the respective conceptual levels. Finally, it needs to be stressed that conceptual-level processes in the domains of the service providers represent refinements of the corresponding external level contractual spheres. In the case study this is illustrated by showing in Figure 22 bigger sized ellipses that represent refined spheres compared to the external level of Figure 23. For the external-level negotiations the collaborating parties employ from Figure 10 the contracting-client component of the Sourcing middleware and service-broker component of the trusted third party.

### 8.3 Selected Spheres in Detail

After the negotiation achieved a preliminary external harmonization of inter-organizational business processes, the following phase in Figure 8 of the internal-to-external interaction pattern is focussing on the verification of the collaborating processes. Earlier internal verification ensures that the business processes of collaborating processes enact correctly on their own. However, when intra-organizational processes are linked inter-organizationally, structural problems may occur, e.g., deadlocks, that prevent the correct termination of the overall business process.

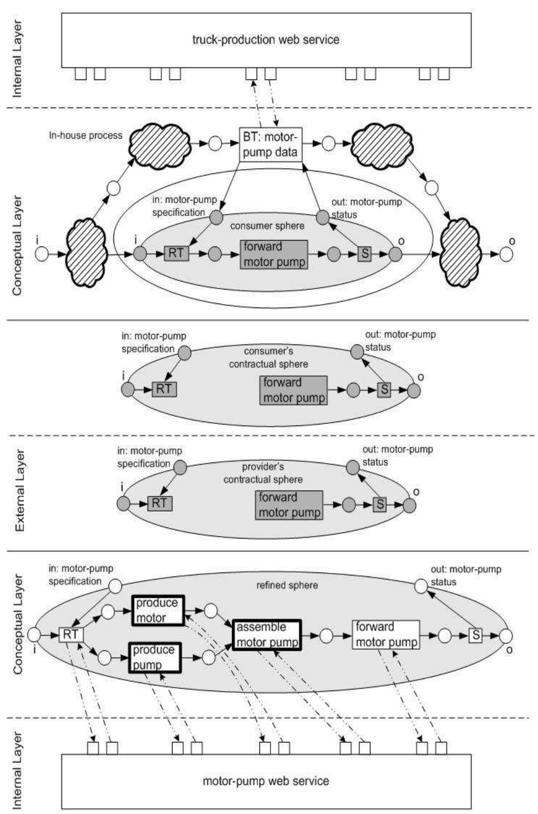


**Fig. 24.** Related eSourcing spheres in detail

In Figure 24, a subset of related eSourcing spheres is depicted in detailed. The conceptual level of the service consumer shows a consumer sphere that is a subnet of an in-house process. Depicted clouds mean the figure abstracts from details of the in-house process. However, it is assumed that an and-split is contained that results in the enactment of parallel branches that is complemented by an and-join at the end. There is only one node depicted that interacts with the ports of the consumer sphere. This interacting node carries a *BT* label that indicates there is a bi-directional exchange with the consumer sphere modelled. The *BT*-labelled node 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 node accepts the motor-pump specification. Next, a node is contained for forwarding 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 *BT*-labelled node via the *out*-labelled interface place.

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 projection [29] is used 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 contractual spheres of the conceptual-level refined sphere with the required additional motor-pump production steps. This scenario is depicted on the external level of Figure 24 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 24 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 from a process point of view that is the result of the earlier negotiation phase between the collaborating parties.

On the conceptual level of the service provider, the contractual sphere is refined by additional nodes. Since a grey-box projection is used, the provider has freedom to complete the content of the contractual-sphere process in the refined sphere. The grey-box projection to the external level does not require adherence to projection inheritance (see Subsection 3). In the motor-pump spheres the OEM does not dictate in which way the production steps should be defined. Figure 24 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. In [29] the constructs used in the detailed eSourcing configuration of Figure 24 are described in detail.

The created eSourcing configuration of Figure 24 must be verified for correct termination before enactment. Thus, the parties independently submit their respective processes to a trusted third party that is depicted in Figure 10. 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 24 this verification fails. The reason is a deadlock contained in the processes that are caused by the *BT*-labelled node. As this node needs to wait for the *S*-labelled node to fire, it can never be enabled.

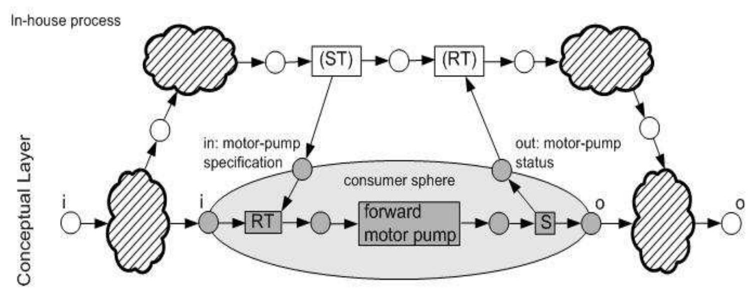


**Fig. 25.** Corrected in-house process

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 of Figure 25 has the *BT*-labelled node replaced with different nodes that establish an exchange between the in-house process and the consumer sphere. Consequently, a repeated termination verification by the trusted third party is successful, provided no deadlocks are caused by other parts of the eSourcing configuration.

### 8.4 Monitorability negotiations

After the inter-organizational processes are verified, the processes need to be linked for a synchronized enactment. In Section 3 such linking across the domains of collaborating parties is termed montorability. In [29] several patterns for monitorability are specified that are available for linking processes into an eSourcing configuration.

In Figure 26 two monitorability constructs are used, namely token messaging and token propagation. Token messaging is used for connecting the *i*-labelled interface places. For an enactment application of the eSourcing configuration, token messaging

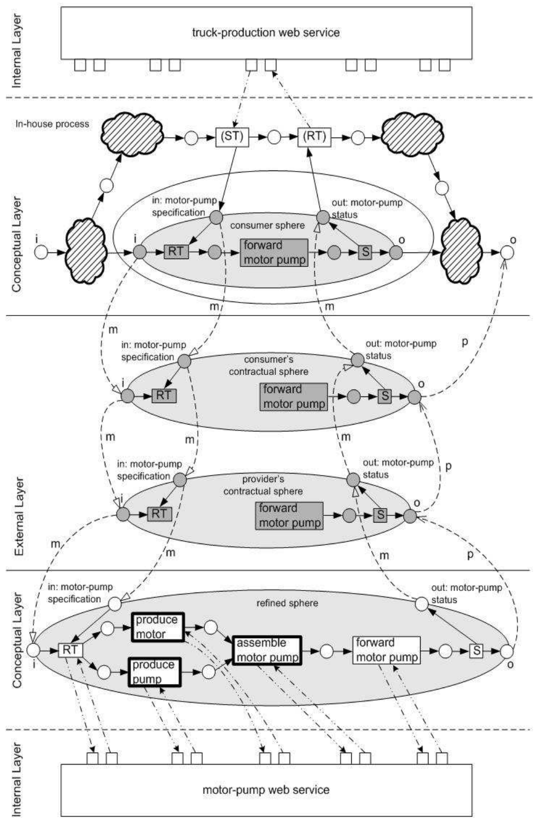


**Fig. 26.** Related eSourcing spheres in detail

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, 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 Figure 26 this means the enactment of the refined sphere is terminated and this event is communicated to the domain of the OEM where the enactment of the next consumer sphere is starting.

For realizing monitorability, the collaborating parties negotiate directly with each other without the help of the trusted third party. Thus, the eSourcing middlewares of the external levels are involved where the global WFMSs and the rules engines need to be linked via the respective coordination interfaces. The agreed upon montorability constructs need to be further realized by appropriately linking the local WFMS and rules engine on the internal level. When this setup is completed, the enactment of the eSourcing configuration commences.

After showing an example of how the internal-to-external interaction pattern is supported by the reference architecture of Section 7, the following section gives a conclusion for this paper.

## 9 Conclusion

This paper focusses on the setup phase of B2B collaboration between service consumers and service providers. The concept of eSourcing is proposed as a foundation for exploring the setup-phase interaction between collaborating business parties. eSourcing uses a three-level framework for the matching of service-consuming and service-providing processes across the domains of collaborating organizations and for the integration of legacy systems into inter-organizational collaboration. Furthermore, eSourcing is an integral concept of the EU research project called CrossWork that has the objective of developing mechanisms for automated workflow formation and enactment, enabling electronic networks of excellency in the automobile industry.

A top-down pattern-exploration approach is chosen for investigating interactions during the setup phase between different business parties for the establishment of inter-organi-zational business-process collaboration. The dimensions called assignment and direction create a bi-dimensional logical space that is organized by combinations of dimension values. Consequently, specifications of interaction patterns are deduced for which the bi-dimensional logical space creates a taxonomy. For each interaction-pattern an example is proposed that is geared towards the concept of eSourcing.

To support the interaction patterns in an automated way, a reference architecture is proposed that is designed in accordance with the principle of functional decomposition. The reference architecture comprises components on three refinement layers that communicate with each other across the three-level framework. By using the reference architecture, the design and development of e-collaboration setup systems is facilitated. In an illustrative case study taken from CrossWork, it is demonstrated how a combination of selected interaction patterns from the assignment- and the direction dimension are supported by the reference architecture.

Scope for further analysis exists for the dimensions and values mentioned in Figure 2. Firstly, additional patterns can be generated by exploring new ways of combining the assignment- and direction-dimensions values. Secondly, added interaction dimensions with corresponding values may extend Figure 2 for new interaction-pattern generation. For example, an added dimension called *initiation* comprises values that state which party initiates interaction for the creation of an eSourcing configuration. Consequently, dimension values may be *service consumer* or *service provider*. Further scope for interaction patterns is given when the dimension *quantity* with the values *one* or *many* is added to Figure 2 for the number of collaborating parties in the role of service consumer or service provider. Adding extra interaction dimensions results in an of the architecture with new components.

## References


1. CrossWork: Cross-Organisational Workflow Formation and Enactment. http://www.crosswork.info/.
2. IBM MQSeries workflow. http://www-4.ibm.com/software/mqseries/workflow.
3. IBM patterns for e-business. http://www-128.ibm.com/developerworks/patterns/.
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 Loosing Control? BETA Working Paper Series, WP 46, Eindhoven University of Technology, Eindhoven, 2000.
6. 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.
7. 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/.
8. 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.
9. 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.
10. W.M.P. van der Aalst and M. Weske. The P2P approach to Interorganizational Workflows. In K.R. Dittrich, A. Geppert, and M.C. Norrie, editors, *Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01)*, volume 2068 of *Lecture Notes in Computer Science*, pages 140–156. Springer-Verlag, Berlin, 2001.
11. 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.
12. S. Angelov. *Foundations of B2B Electronic Contracting*. Dissertation, Technology University Eindhoven, Faculty of Technology Management, Information Systems Department, 2006.
13. S. Angelov and P. Grefen. The 4W framework for B2B e-contracting. *International Journal of Networking and Virtual Organizations*, 2(1):78–97, 2003.
14. 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.
15. L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Series in Software Engineering. Addison Wesley, Reading, MA, USA, 1998.
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. 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.
18. P. Grefen. Service-Oriented Support for Dynamic Interorganizational Business Process and Data Management of Complex E-Services. to appear, 2006.
19. 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.

20. P. Grefen and R.R. de Vries. A reference architecture for workflow management systems. *Data & Knowledge Engineering*, 27(1):31–57, 1998.
21. 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.
22. E. Kindler and W.M.P. van der Aalst. Liveness, Fairness, and Recurrence. *Information Processing Letters*, 70(6):269–274, June 1999.
23. E. Kindler and M. Weber et al. *Petri Net Kernel (PNK) Home Page*. http://www.informatik.hu-berlin.de/top/pnk/, 2003.
24. Lea Kutvonen, Janne Metso, and Toni Ruokolainen. Inter-enterprise collaboration management in dynamic business networks. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE*, volume 3760 of *Lecture Notes in Computer Science*, Agia Napa, Cyprus, November 2005.
25. A. Lazcano, H. Schuldt, G. Alonso, and H Schek. WISE: Process Based E-Commerce. *IEEE Data Engineering Bulletin*, 24(1), 2001.
26. L.A. Maciaszek. *Requirements Analysis and System Design. Developing Information Systems with UML.* Addison Wesley, 2001.
27. A. Norta. XRL/flower Home Page. http://is.tm.tue.nl/research/xrl/flower.
28. 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.
29. A. Norta and P. Grefen. Discovering Patterns for Inter-Organizational Business Collaboration in a Top-Down Way. BETA Working Paper Series, WP 163, Eindhoven University of Technology, Eindhoven, 2006.
30. IBM Research. Crossflow architecture description, Technical report, ESPRIT Crossflow EP 28653, 1999.
31. N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Resource Patterns. *QUT Technical report*, forthcoming.
32. 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.
33. A.S. Tanenbaum. *Modern Operating Systems*. Prentice Hall, 1992.
34. 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.
35. G. Wagner. How to design a general Rule Markup Language?, Invited talk at the Workshop XML Technologien fur das Semantic Web (XSW 2002), Berlin, 24-25 Juni 2002.
36. G. Wagner, S. Tabet, and H. Boley. MOF-RuleML: The abstract syntax of RuleML as a MOF model. br/03-10-02, 2003.
37. WFMC. Workflow reference model. Technical report, Workflow Management Coalition, Brussels, 1994.
38. 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.