A Reference Architecture for Dynamic Inter-Organizational Business Process Collaboration

A. Norta¹ and P. Grefen²

¹ Department of Computer Science, P.O. Box 68 (Gustaf Hällströmin katu 2b), FI-00014 University of Helsinki, Finland
alexander.norta@cs.helsinki.fi

² Eindhoven University of Technology, Faculty of Technology and Management, Department of Information Systems, P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
p.w.p.j.grefen@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 eSourcing (eSourcing) has been proposed as an integral concept for the EU research project CrossWork. The properties of eSourcing have been explored, however, the question arises how a service consumer and a service 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. Hence, this paper investigates the characteristics of interaction between a service consumer and service provider during setup time for establishing an enactable B2B collaboration. These characteristics are employed to discover interaction patterns in a top-down way that are specified together with interaction scenarios for the concept of eSourcing. These scenarios constitute the basis for evaluating the design of an eSourcing Reference Architecture (eSRA) that serves as a foundation for the design of e-collaboration setup systems.

1 Introduction

The collaboration between industrial companies in the B2B domain is complex from a business, conceptual, and technological point of view. Observing such collaboration, particular features are characteristic. For example, 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 not 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 suppliers are requested to perform a mirroring of the particular parts of the in-house process that the OEM does not 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 not 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 is expressible. Additionally, the parties are reluctant to disclose internal business details for fear of revealing their competitive advantages. That makes it even harder to assess the capability of process mirroring.

With the application of service-oriented information technology, companies hope to reduce the costs for personnel that deal with establishing and maintaining a B2B supply chain. It is a worthwhile objective that enterprizes find each other in an automated way with the support of middleware in an effective and efficient way for providing and consuming a service. Still, most research projects about inter-organizational B2B collaboration [1, 14, 38, 52] 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 determines which party initiates service consumption. Such forces suggest the setup phase of inter-organizational business process collaboration follows different paths.

As a means of investigating those different collaboration paths, this paper pursues a pattern-based investigation for achieving an improved insight with respect to 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 comprise scenarios that are the basis for exploring the features of middleware that supports dynamic inter-organizational business process collaboration in a suitable way. Hence, in this paper scenario-based software architecture evaluation methods [19, 21, 32, 33, 37] inspire the development of the reference architecture.

The interaction patterns and the deduced reference architecture are embedded in the emerging framework of dynamic inter-organizational business process management (DIBPM) [27]. 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 architectures and research projects about inter-organizational business process collaboration. Section 3 introduces the reader to the concept of eSourcing that is instru-
mental in an ongoing research project where a proof-of-concept prototype for dynamic inter-organizational business process management is under construction. In Section 4, an adapted version of the earlier mentioned scenario-based evaluation is used for exploring eSourcing in a pattern-based way. The resulting interaction scenarios serve as input for the formulation of functional requirements that the reference architecture of Section 5 must adhere to. Section 6 evaluates the reference architecture by showing how the eSourcing-setup scenarios from the specified interaction patterns are supported. Finally, Section 7 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 scenario-based evaluation methods, and completed 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 they 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 two reference architectures are incorporated in the reference architecture in the sequel of this paper. Furthermore, an overview of scenario-based methods is presented for evaluating reference architectures. An adapted version of these evaluation methods is used for designing and evaluation eSRA in this paper.

In the final subsection, four research project are listed that deal with the issue of inter-organizational business process management. These projects focus on 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. [26] 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 [10–12] by investigating several intra-organizational workflow systems for commonalities. The resulting patterns are grouped into different categories. Basic patterns contain a sequence, basic
splits and joins, and an exclusive split of parallel branches and their simple merge. Further patterns are grouped into the categories advanced branching and synchronization, structural patterns, patterns involving multiple instances, state-based patterns, and cancellation patterns. The resulting pattern catalog is instrumental for the evaluation \cite{9,58} of web service composition languages (WSCLs).

Following a similar approach as in the control-flow perspective, data-flow patterns \cite{53} are grouped into various characteristics categories. One category focuses on different visibility levels of data elements by various components of a workflow system. The category called data interaction focuses on the way in which data is communicated between active elements within a workflow. Next, data-transfer patterns focus on the way data elements are transferred between workflow components and additionally describe mechanisms for passing data elements across the interfaces of workflow components. Patterns for data-based routing deal with the way data elements can influence the control-flow perspective.

Patterns for the resource perspective \cite{54} are aligned to a lifecycle for work items. A work item is created and either offered to a single or multiple resources. Alternatively, a work item is allocated to a single resource before it is started. Once a work item is started it is optionally 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 are instrumental for grouping pattern specifications.

So-called service interaction patterns \cite{17} 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 available online \cite{5} 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 information-system applications. A few examples are the well-known OSI reference model \cite{55}, the WorkFlow Reference Model \cite{57}, the Reference Architecture
for Workflow Management Systems [29], and more recently the E-contracting Reference Architecture [15].

The OSI reference model [55] 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 that it only exports functionality to the layer above.

The high-level view of the Reference Architecture for Workflow Management Systems [29], and the Workflow Reference Model [57] are used in this paper for specifying the eSourcing Reference Architecture eSRA that supports the exemplary scenarios of the 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 in eSRA to enable a local business-process setup that is followed by their enactment.

Finally, the E-contracting Reference Architecture [15] 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 Scenario-Based Software Architecture Evaluation Methods

While the evaluation methods described below are guiding for the development and evaluation of eSRA, a more detailed discussion of the method used in this paper follows in the sequel. The Software Architecture Analysis Method (SAAM) [21, 32] is the initial scenario-based evaluation method developed for assessing the quality attributes of a reference architecture. SAAM supports architects to reason about software-system quality without offering assessment metrics or tool support.

The evaluation steps of SAAM start with the development of scenarios, followed by a description of the architecture. Next, a classification into direct and indirect scenarios takes place together with a prioritization. While all events of a direct scenario are fully supported by the architecture, changes are required to achieve a total support of all events that comprise an indirect scenario. The latter scenario type is individually investigated to determine the extent of architecture change. Two or more scenarios interact when their evaluations indicate changes for the same component of an architecture. To avoid such a situation, the respective component must be modified or divided into sub-components. The final step of the SAAM method is a weighing of the scenarios relative to their importance for the architecture’s success. The weighing is instrumental for determining the overall ranking of multiple architectures.

The Architecture Trade-Off Analysis Method (ATAM) [21, 33] is based on SAAM and also explores quality attributes. However, differently to SAAM, a bigger emphasis is put on the interdependencies between these quality attributes. Briefly, the presentation phase focuses on presenting the ATAM method, the business goals that motivate the development, and the architecture itself. The investigation and analysis phase assesses the main quality attribute requirements versus different architectural approaches.
To support this phase, a quality-attribute-utility tree [33] is created that is specified down to the level of scenarios, annotated with stimuli and response, and prioritized. The testing phase evaluates earlier generated prioritized scenarios against architecture alternatives with the help of the quality-attribute-utility tree. Finally, the ATAM results are summarized in a report.

The Architecture-Level Modifiability Analysis (ALMA) [37, 19] is developed for assessing the software-architecture modifiability by employing a set of quality indicators. ALMA consists of five steps and commences with setting the analysis goal, followed by a presentation of the architecture. Next, change scenarios are elicited that play a role in an architecture’s modifyability. After evaluating the change scenarios against the earlier set goals, an interpretation of the results is presented.

For sake of completeness, two more evaluation methods are mentioned that have less influence on developing the evaluation method that is used in this paper. Differently to the already mentioned scenario-based evaluation methods, the Cost-Benefit Analysis Method (CBAM) [21, 31] includes as a criteria the costs that need to be considered among the tradeoffs based on the ratio “benefit divided by cost”. Finally, the Family-Architecture Assessment Method called (FAAM) [22] is a method for the architecture assessment of information-system families. FAAM focuses on two related quality aspects, namely interoperability and extensibility.

2.4 Research Projects

As the introduction of this paper mentions, several research projects have investigated inter-organizational process collaboration. In the CrossFlow project [52], the formation of virtual enterprises is realized by dynamic outsourcing. A service matchmaker matches service offerings and service requests. Based on the resulting electronic contract, a service enactment infrastructure [30] is established dynamically that employs workflow technology. CrossFlow has an external level that spans across organizational domains in which the process is part of a contract specification. The workflow-specification language of the workflow-management system IBM MQSeries Workflow [4], forms the internal process level.

In the WISE project [14, 38], a software platform is provided for process-based B2B electronic commerce that focuses on supporting a network of small and medium-sized enterprises. While CrossFlow relies on cooperating pairs of autonomous workflow systems with a peer-to-peer relation, WISE employs a central workflow engine to control cross-organizational processes that are termed virtual business processes. In WISE, a virtual business process consists of a number of black-box services that are linked in a workflow process [14]. A service is offered by an organisation that is involved in a WWW catalogue of business processes, which are controlled by local workflow-management systems.

In the CrossWork [1] project, the objective is pursued to develop automated mechanisms for allowing a dynamic workflow formation and enactment, enabling a collaboration and synergies between different organizations. CrossWork uses eSourcing [42–45] as an integral concept that focuses on matching on an external level conceptually formulated service-consuming and service-providing processes. In Section 3, further details are explained.
The web-Pilarcos B2B middleware [36] represents one solution for managing the lifecycle of dynamic business networks in an inter-enterprise environment. It forms a loosely-coupled collaboration layer on top of service-oriented middleware by using the 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 a service, which providers submit earlier to a public repository. However, the architecture requires further development of business-process-modelling techniques. According to [36], the collaboration of business processes or workflows must be modelled without revealing local processing steps. Instead, only the collaborative part’s external view must 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 based on which examples for interaction patterns are specified.

3 The Concept of eSourcing

Within DIBPM the concept of eSourcing focuses on the structural matching of processes that belong 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 [45]. To manage the inherent business, conceptual, and technological complexity of eSourcing, the employment of a three-levels framework [28] 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, ERP systems, and so on. 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 a matching that realizes an automated and dynamically forged collaboration between partners.

Without public processes on the external level and private processes on the conceptual levels of an eSourcing configuration, B2B collaboration is hampered [20]. Instead, if the collaborating parties share one common business-process definition or instance state, this constitutes a violation of their competitive knowledge protection. Furthermore, if process definitions are shared, correct message-exchange and message-transformation implementation, and the enactment of shared business rules becomes
Fig. 1. An abstract eSourcing-configuration example using a three-levels framework
problematic, since adding a collaborating party quickly results in an explosion of the business-process definition. 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 providing organizations into a B2B supply-chain collaboration. Important elements of eSourcing are the support of different visibility levels of corporate process details for the collaborating counterpart and flexible mechanisms for service monitoring and information exchange.*

To understand the definition of eSourcing better, Figure 1 is used to explain relevant parts of an eSourcing configuration, which are modelled using labelled Petri nets [50, 51]. Squares are denoted as transitions as they propel the net and circles are denoted as places representing states. Transitions are always equipped with labels. In a Petri net, places may contain tokens and transitions consume these tokens from places. Transitions are enabled when all their input places contain at least one token. Such an enabled transition consumes one token from each of its input places during firing. After firing, a transition produces tokens that are placed into all its output places. Arcs in a Petri net do not link two transitions or two places with each other.

### 3.1 A Motivating eSourcing Example

Starting with the domain of the service consumer in Figure 1, an in-house process is depicted on the conceptual level. The special type of Petri nets used in eSourcing, namely workflow nets (WF-nets) [23], has one unique input place and one unique output place. All other transitions and places in a WF-net contribute to its processing. WF-nets carry the property of *soundness* [7, 34], which informally states that after the completion of a net, only one token must remain in the unique output place and all other places must be empty. WF-nets are particularly suitable for modelling and verifying business processes. Hence, the in-house process is a WF-net.

The in-house process contains a subnet termed a consumer sphere that is visualized as a grey ellipse. On the border of the consumer sphere, interface places are located. The interface places are either in or out-labelled to represent the exchange direction between the in-house process and its contained consumer sphere. Only one place of the sphere is *i*-labelled and only one is *o*-labelled.

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 contractual sphere. From the opposite eSourcing domain a complementary provider 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 [16] of electronic contracting.

The provider contractual sphere is complemented by a provider sphere on the conceptual level. Compared to the provider contractual sphere additional nodes are contained in the provider sphere. In Figure 1, such refinement is depicted by unlabelled
transitions that do not exist in the provider contractual sphere. The refinements in the provider sphere must be in accordance with projection inheritance [8] that is informally defined as follows:

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

Hence, process x inherits the projection of the process definition y while process x conforms to the dynamic behavior of its superclass by hiding transitions new in x. 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 provider sphere, the obligatory requirement of well-directedness of an eSourcing configuration must be mentioned. This requirement focuses on the interface places of the spheres, which are part of exchange channels for business-critical information between spheres and the remaining in-house process. An 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 provider sphere are equal in number and labelling.

Finally, in Figure 1 referencing arcs are depicted that connect several places of the respective spheres. By using these referencing arcs, a transitive relationship is established between the consumer sphere and the provider 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 is part of so called monitorability patterns [42, 43, 48] that also cover the referencing of transitions contained in the consumer sphere and the provider sphere.

After a brief overview of the eSourcing concept, the following section presents a method that is an adaption of the earlier discussed scenario-based methods of Section 2.3. In the sequel, the method used in this paper results in the development of an eSourcing reference architecture.

4 A Method for Developing and Evaluating eSRA

Inter-organizational B2B collaboration comprises 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. Intra-organizational 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 such business processes are linked in a B2B supply chain, the setup phase requires an additional perspective that focuses on the way how business processes need to be linked inter-organizationally. Thus, the interaction perspective deals with 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 focuses 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.

For eSourcing, the objective is pursued to realize expressive and suitable applications that support collaborating organizations in setting up and enacting eSourcing configurations in an automated way. Here, the setup phase focuses on specifying an inter-organizational business-process template and linking the information infrastructure of the respective collaborating parties according to the three-levels framework [28], i.e., the legacy systems on the respective internal levels are not directly linked with each other.

A well-structured approach is needed to explore the features that are inherent to setting up a collaboration scenario of Figure 1, in order to define a reference architecture that supports the setup phase of an eSourcing configuration. The problem must be tackled of having to deal with a high degree of business, conceptual, and technological complexity that results from a heterogeneous system environment of collaborating business parties. This complexity is tackled by adapting the approaches of Section 2.3 that use scenarios as input for the development and evaluation of reference architectures.

To explore the interaction perspective for eSourcing, the following method is chosen. Based on the careful exploration of research literature [13, 14, 30, 36, 38, 52], past research projects (see Section 2.4), and case studies with industry partners in the CrossWork project [1], interaction features are detected and organized. The most distinctive features are dimensions that establish a logical space for a two-dimensional grid. Along the dimensions, further refining interaction features are located that create a taxonomy of interaction scenarios within the logical space.

Every refining feature value of the interaction dimensions needs to be specified in a conceptual and technology independent way. Hence, a collection of interaction-pattern specifications is deducted from the logical space that comprise interaction examples for every pattern. These examples are scenarios modelled as sequence diagrams [25] from which requirements are extracted for the specification of a reference architecture. The exemplary scenarios from the interaction patterns are used to perform a final evaluation of eSRA. For brevity two specific scenarios are chosen and applied in an evaluation that is inspired by case studies of the CrossWork project.

While the method chosen for eSRA development and evaluation also uses scenarios, it differs compared to the earlier mentioned scenario-based evaluation methods of Section 2.3. Firstly, it must be mentioned that the latter methods do not detect scenarios in an equally thorough way as the method of this paper. Secondly, for the latter methods the specified scenarios do not influence to the same degree the definition of a reference architecture. Instead, the method of this paper uses the allocated scenarios up front for determining the architecture’s requirements. Hence, the methods of Section 2.3 need to be more elaborate towards the end when it is determined whether specific requirements are fulfilled. The method of this paper establishes a better fit between the set of scenarios and the reference architecture at the beginning and results in an evaluation at the end that demonstrates the architecture’s feasibility.

The remainder of this section focuses on the creation of an interaction-patterns catalogue with exemplary scenarios. All the features for eSourcing interactions during the setup phase are presented in Section 4.1 followed by the definition of a pattern-
specification template in Section 4.2. While in [42, 43, 47] the complete catalogue of interaction patterns is contained, two deducted pattern specifications are presented in Section 4.3 and Section 4.4. Finally, Section 4.5 lists the requirements for eSRA that result from the pattern-specification examples.

4.1 Dimensions and Values of Interaction Patterns

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 like XML-based industry standards [9, 58], should adhere to. For the pattern catalogues of Subsection 2.1, a bottom-up approach was chosen where several intra-organizational workflow-management systems and 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. 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.

Figure 2 shows the dimensions for interaction patterns. One dimension is called assignment, which focuses on 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. The assignment dimensions are described as follows.

- **Static** assignment means, the collaborating parties already know before setup time they surely collaborate with each other.
- **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 therefore 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 focuses on the nature 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 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 combined. Thus, according to the figure, there are twelve pattern combinations possible. For sake of brevity, the examples covered in this paper represent a subset of these pattern combinations.

4.2 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.

While in this paper only one interaction-scenario example is presented below, in [42, 43, 47] additional scenario examples are contained in the complete list of specified interaction patterns. For the scenario examples of patterns, sequence diagrams are used that propose particular interactions during the setup phase of an eSourcing configuration. For sake of brevity, completeness of the interactions is not claimed.

Chunks of the overall interaction sequences in [42, 43, 47] 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, 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 additional interaction example in [42, 43, 47] depict the contracting interaction block as a black box, abstracting from the identical message exchanges that Figure 3 reveals.

Next, for two dimension values of Figure 2 one pattern is deduced that is specified using the description template mentioned above. The patterns are specified in a way so that each tries to factor out specifics of one chosen dimension value.

4.3 Chosen Assignment-Dimension Pattern

**Pattern 1 (Static Assignment)**

**Problem:** Due to market pressures a company is involved in optimized 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 offering an agreed upon service. 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, if a provider is not capable of performing the agreed upon service, 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.
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.

An abstract example of static assignment that focuses 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 do not 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 provider 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 provider 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.

4.4 Chosen Direction-Dimension Pattern

**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 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 are variants of the out-sourcing direction pattern.

4.5 Requirements for eSRA

In the complete list of interaction patterns in [42, 43, 47], six exemplary scenarios are contained for the deduction of requirements for an e-collaboration setup system. The specified interaction blocks of these scenarios indicate functional requirements for eSRA.

Functional requirements cover specific behaviors of a system while non-functional requirements specify criteria that can be used to judge the operation of a system, rather than specific behaviors. Other terms for non-functional requirements are “quality attributes” and "quality of service requirements". For the functional requirements listed below, it is assumed that business processes also comprise business rules that influence
the enactment direction of control-flow. The first requirement is not deducted from interaction blocks, but results from the three-levels framework [28] the eSourcing concept uses.

**a.) Indirect linking of business processes and legacy systems** This requirement must be realized in an e-collaboration setup system by distributing components on three levels that serve distinct purposes in accordance with the three-levels framework [28]. Local enactment components need to be situated on the internal level. The conceptual level needs to have components for conceptually expressing business processes and the external level must comprise components that facilitate an inter-organizational harmonization of business processes. Translation components on the conceptual level must enable a communication between the external level and the respective internal levels of the collaborating counterparts. Such communication implies the availability of functionality for projecting business processes and their accompanying rules to the external level, and functionality for mapping processes to internal-level processes and accompanying rules.

**b.) Support for the conceptual formulation of business processes and their accompanying rules** The respective in-house process and the provider sphere must be expressed in a conceptual way. Hence, components must be available for conceptual modelling. In order to prevent a constant reinvention of business-process constructs, it should be possible to store in and retrieve from a database such conceptually formulated constructs. This requirement corresponds to the interaction block termed **in-house setup**, **template loading**, and **internal setup**.

**c.) Brokering capability of projected business processes** Both, the service consumer and the service provider must be able to place their externalized business processes to a broker environment of a trusted-third-party service. Such functionality is important for collaborations in an anonymous collaboration environment where the potential business parties do not know each other or where it is of interest to engage in a bidding procedure for either service provision or service consumption. This requirement corresponds to the interaction block termed **broker commit**.

**d.) Bidding capability of projected business processes** Such a bidding environment must be offered as a trusted-third-party service. Service offers and service requests must be searchable for potential business partners and these must be enabled to place bids. The collaborating counterpart should be able to evaluate the bids and choose the best deal while rejecting all others. Once the bidding is over, the request for service consumption or service provision is removed. This requirement corresponds to the interaction block termed **bidding** and **consumer offering**.

**e.) Setup support of an eSourcing collaboration for known collaborating parties** When collaborating parties have found each other, they need a platform for starting
the contracting negotiations on the external level of an eSourcing configuration. This negotiation involves the projection of business processes to the external level until a matching is achieved that establishes a consensus about the nature of service provision and service consumption. A trusted-third-party service must verify if such a matching of business processes is achieved. Additionally, the collaborating parties must negotiate to which extend it is possible for a service consumer to monitor during enactment time the progress of service provision. This requirement corresponds to the interaction block termed contracting, monitorability negotiation, and external negotiation.

f.) Verification and evaluation of eSourcing collaborations From a control-flow point of view it is important to verify an overall electronic collaboration for correct termination. A verification must ensure that a service provision internally adheres to what is externally promised to a service consumer. Verification must also cover other perspectives than control-flow, e.g., data-flow, resources, transactional properties, and so forth. This requirement corresponds to the interaction block termed verifying.

g.) Enactment of a ready collaboration configuration When the setup phase is completed, the enactment of an electronic collaboration needs to commence. Although a verification of different collaboration perspectives must be performed, errors may still occur when all perspectives are enacted. Hence, an evaluation component for business processes must be available for a-priori test enactment. The actual enactment components must be present on an internal level for coordination legacy systems. On the other hand, additional enactment components on the external level need to coordinate the internal components of the respective collaborating parties. This requirement corresponds to the interaction block termed enactment.

The listed requirements are guiding for the development of an e-collaboration setup system for which a specification is presented in the following section. In this paper, the e-collaboration setup system is presented for two refinement levels, although in [42, 43, 47] a third refinement level is contained.

5 An Architecture for Supporting Interaction Patterns

Software development consists of three main phases, i.e., the analysis, design, and implementation phases [39]. 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.

A reference architecture for the domain of electronic interaction between collaborating business parties embodies essential 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 paper serves as a foundation in the design and development of e-collaboration setup systems.

The following subsections first explain the design approach followed by two detailing level of the reference architecture of which in [42, 43, 47] a third refinement level is available. The last subsection discusses how the reference architecture supports the eSourcing examples of Section 4.3 and Section 4.4 that are provided in the interaction pattern specifications.

5.1 Design Approach

The eSourcing 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 [18]. Thus, at each level of eSRA, the identified components provide functionalities that do not overlap with the remaining components that are located at the same level. To achieve completeness, eSRA is designed in a top-down way. As a result eSRA's functionalities and interactions are addressed in a step-by-step manner from a high level of abstraction on the top level and the level of detail gradually increasing on lower levels.

In Section 3, it is demonstrated how the concept of eSourcing is using three levels [28] for realizing the interoperability of business processes and their supporting heterogeneous 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 focusing on modeling 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 eSRA in the three-levels framework provides the basis for achieving interoperability between collaborating parties. The next two subsections describe eSRA in further detail.

5.2 First Detail-Layer of an eSourcing-Setup Reference Architecture

The depiction of Figure 4 shows the highest abstraction layer of 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 of services and checks quality features of eSourcing configurations before enactment. If collaborating parties perform verifications and checks of eSourcing configurations themselves, they would need to reveal competitive secrets to each other, which is undesirable.

The conceptual level of Figure 4, depicts two components, namely the translators and the eSourcing setup support. The translator component exchanges information between the components located on 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 on the other hand with the legacy system of a collaborating party.

5.3 Second Detail-Layer of the Reference Architecture

In this subsection each component of the reference architecture depicted in Figure 4, is further refined. The first refinement in Figure 5 covers all components that are located on the external level, namely the eSourcing 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 5, the eSourcing middleware of one collaborating party is depicted. The eSourcing counterpart contains the same second-level components. For sake of brevity, only the relationship between the coordination-interface components is depicted. Figure 5 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 Sourcing-middleware is an abstraction of an information exchange that is described in [42, 43, 47] in further detail.
In the trusted-third-party 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 snippets that are stored in a corresponding database of the trusted third party. That way the workflow snippets are communicated between collaborating organizations.

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, bids need placed with the auction service.

The latter component relates the bid data with services stored in the service broker. When an eSourcing configuration is established, the collaborating parties send their in-house processes and provider spheres to a verifier component for testing the correct termination, i.e., the soundness [7, 34], of the overall eSourcing configuration. The verification results are returned to the collaborating parties. By having a trusted third party perform the verification, the collaborating parties do not have to disclose their internal business details to each other.

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 workflow management system (WFMS) respectively. In order to synchronize the global WFMSs and global rules engines in the Sourcing-middleware components of other collaborating parties, events-, production-, and rules data are communicated via a coordination interface. For example, production data are the specification of a
product or data needed for condition statements. The global WFMS and rules engine also exchange production-, and event data with each other.

The contracting client sends workflow snippets and composed processes and contractual rules to the translator. That way the workflow snippets and composed processes and contractual rules are prepared for the heterogenous system environment that exists on lower internal levels of a collaborating party. Also the global WFMS and rules engine send data to the translator component that is depicted as a white box in Figure 6. The translator contains two main translator components for transferring data between the external, conceptual and internal level.

The CE translator component of Figure 6 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-set-up-support component. The relationships between the CE translator and components contained in the eSourcing middleware are explained above.

Two components exchange data between the CE translator and CI translator, namely the workflow/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 eSourcing configurations. On the internal level, several web services wrap legacy systems to which exchanged is routed.

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 provider spheres from the process modeler. These rules and processes are translated to the local WFMS and rules engine on the internal level.

In Figure 7, the Sourcing-setup-support component is located on the conceptual level. The component has two core functions, namely modelling business rules and processes, and composing workflows that are on the one hand evaluated and on the other hand verified for correct termination. Thus, the rules modeler and the process modeler are responsible for the first function for which they are supported by a pattern knowledge base. In [42, 46, 49], the pattern knowledge base is presented in further detail. The second function is related to the workflow composition component. For composition [24], workflow snippets 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 provider sphere and is checked internally in two ways. First, with respect to control flow, correct termination is verified by the tool Woflan [56] for which the process needs to be mapped to a place/transition net. 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 [7]. Secondly, the in-house process or provider 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 provider spheres of an eSourcing configuration with an additional tool. Among
other aspects, such a validation is meaningful for testing how the different perspectives fit together 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 7 XRL/flower [41] is depicted as a validation tool, which uses XML technology and is implemented in Java on top of the Petri-net Kernel PNK [35]. 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.

Figure 8 visualizes a second-level refinement of the legacy-management component. In it, a local WMFS and rules engine constitute the core components. These components exchange data between each other and are instrumental for coordinating legacy systems. The business rules and processes that are enacted by the WMFS and rules engine are translated down to the internal level by the CI translator. For enactment, the local WMFS and rules engine use a production database. Furthermore, to coordinate the enactment on an internal level and external level, the local WMFS and rules engine communicate events, rules, and production data bi-directionally.

With respect to the list of functional requirements in Section 4.5, the presented eSRA specification must be evaluated to check for adherence. To allow the reader an enhanced insight, the following evaluation employs two additional interaction patterns that differ from the two specifications earlier presented in this paper. By complementing the earlier listed functional requirements with additional non-functional requirements in the following section, the evaluation of eSRA is improved.
6 Scenario-Based Evaluating of eSRA

For evaluating the concept of eSourcing, business cases from the automobile industry are used. Since the case study for evaluating eSRA are inspired by CrossWork [2, 3] case studies, it is important to stress the differences. First, in the latter case the emphasis of the case studies is on solving specific practical problems that CrossWork industry partners are facing. Secondly, these case studies are large and extensively documented while in contrast, the case study in this paper needs to be brief. Still, the case in this paper is inspired by large CrossWork case studies. Finally, it must be stressed that eSRA is not fully implemented as a proof-of-concept prototype [40] in CrossWork. Instead the latter prototype represents a subset of eSRA components.

The remainder of this section is structured as follows. First, Section 6.1 characterizes supply-chain features of the automobile industry, followed by a presentation of additional non-functional eSRA requirements in Section 6.2. The case study in Section 6.3 focuses on the eSRA application in the case study. Finally, the requirements of Section 4.5 and Section 6.2 are revisited and it is assessed in Section 6.4 how eSRA adheres to them.

6.1 Characteristics of Automobile Supply Chains

In the automobile industry, OEMs have several tiers of suppliers that agree to deliver systems collaboratively. For example, the OEM assembles cars with systems like a cockpit, or an engine, etc. These systems are manufactured by Tier 1 that gets the components for those systems from a Tier 2 supplier.

![Supply-chain hierarchy in the automobile industry.](image)

As depicted in Figure 9, the supply-chain relationship between an OEM and suppliers resembles a pyramid where the OEM at the top spends considerable time and effort on aligning first and second tier suppliers for achieving the desired service provision.
Additionally, the overall number of produced cars and also the number of variants is going up while the lifetime of car-types is shortening, which means the number of cars per type is decreasing. To deal with the resulting complexity in manufacturing as well as in design and development, OEMs are shifting parts of their activities down the organizational hierarchy. By applying eSourcing with specifying the inter-organizational collaboration with eSML, this coordination effort between collaborating parties is facilitated.

The CrossWork-inspired case study is for the assembly and supply of water tanks for trucks. First, the order of the OEM 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 body, a grommet, a motor pump, a dispenser, and a sealing ring. Provider A receives the order for the entire automobile cluster from the OEM and organizes the distribution of the production of water-tank parts to partners of the automobile cluster.

It is decided the 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, 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. For evaluating the concept of eSourcing, business cases from the automobile industry were taken and translated into instances of the eSourcing Markup Language eSML [42, 43], an XML-based proof-of-concept language.

### 6.2 Evaluation Requirements

Since the requirements listed in Section 4.5 are of a functional nature, it is important to complement them with an additional list of requirements that are primarily non-functional and suitable for the conducted case study in the sequel. Table 1 presents a requirements classification, stating which are functional and non-functional.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasibility</td>
<td>functional</td>
</tr>
<tr>
<td>Scalability</td>
<td></td>
</tr>
<tr>
<td>Interoperability</td>
<td></td>
</tr>
<tr>
<td>Coherence</td>
<td></td>
</tr>
<tr>
<td>Applicability</td>
<td></td>
</tr>
<tr>
<td>Completeness</td>
<td></td>
</tr>
</tbody>
</table>

The requirement **feasibility** means that it is possible to set up a business case with eSRA components. **Scalability** refers to the ability of eSRA to combine many collaborating parties into one eSourcing configuration. **Interoperability** is the capability of eSRA to
realize a technical and conceptual harmonization of intra-organizational business processes for a linked enactment phase. In the following section, the requirements for eSRA are evaluated. Coherence must be established by eSRA between external tasks and their internal counterparts. Applicability states that eSRA is instrumental for guiding the development of application systems for inter-organizational collaboration that support an indirect connection of internal legacy systems. Completeness is the quality of comprising the components required for setting up and enacting an eSourcing configuration satisfactorily.

6.3 eSRA-Requirements Evaluation

A truck-producing OEM 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. Both, the OEM and the suppliers have historically grown business process that can not be rearranged easily, which is why an external-level process harmonization is required. Thus, the case study applies the internal-to-external interaction pattern of the direction dimension. The OEM always receives the same system for the truck from the same set of suppliers that are united in a cluster.

Within that cluster one supplier serves as the unique communication and coordination party between the OEM and all other suppliers of the cluster. Thus, the internal-to-external pattern is combined with the static assignment pattern (see [42, 43, 47]) since the OEM has one predetermined supplier to collaborate with. In this case study, it is pointed out in which way components of eSRA are used to support this case study.

Fig. 10. OEM in-house process.
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 commences.

**Conceptual-Level Setup:** Figure 10 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 10, the in-house process for producing a truck is depicted. Contained in the in-house process is a consumer sphere that delimits a subnet for eSourcing a water tank from service providers. Since for this eSourcing-concept evaluation the original processes are too large to fit into this chapter, the in-house process of Figure 11 doesn’t use Petri-net formalisms, but a high-level conceptual visualization.

![Fig. 11. Provider domains with their respective provider spheres.](image)

The service provider domains are depicted in Figure 11 with their internal setup. Four provider domains are contained in the figure that are part of an automobile cluster with each containing processes that create parts of a water tank for the OEM. According to Figure 11, a service provider may be capable of contributing several processes for producing parts of a water tank. Similarly 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.

For modelling the conceptual level processes depicted in Figure 10 and Figure 11, eSRA comprises process-modeler components. Furthermore, the workflow-composer component is available to support the suppliers of the automobile cluster in establishing a composed workflow. The process-modeler and workflow-composer component are
part of the Sourcing-setup-support component depicted in Figure 7. Although the case study focuses on the business processes, it is assumed that business rules are created with the rules-modeler component that is also depicted in Figure 7.

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 6 depicts that processes and rules are delivered to the internal level via the CI translator component of Figure 6. Furthermore, in Figure 8 the local WFMS and rules engine are the recipients that consequently control web-service wrapped legacy systems.

**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 10 by the rules-modeler and the process-modeler components of the Sourcing-setup-support component of Figure 7. Next, the consumer sphere is transferred via the CE translator of Figure 6 to the contracting client of the external level. The latter component exposes the OEM’s contractual sphere to the service broker of Figure 5 that is part of the trusted third party. Note that a trusted-third-party component is not part of the CrossWork prototype. As a result, service brokering is out of focus in CrossWork. 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 5 and requests the OEM’s contractual sphere for a check. In Figure 12 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. The higher-level consumer sphere is designed to be a fitting subnet within the overall in-house process while each contained consumer spheres is eSourced from a supplier of the automobile cluster.

On the service provider’s side of the external level in Figure 11, 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 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. The refinements are additional tasks that are integrated in the provider sphere that remain opaque for the service consumer. In the case study this is illustrated by showing in Figure 11 bigger sized ellipses that represent provider
spheres compared to the external level of Figure 12. For the external-level negotiations the collaborating parties employ from Figure 5 the contracting-client component of the Sourcing middleware and service-broker component of the trusted third party.

**Selected Spheres in Detail:** After the negotiation achieved a preliminary external harmonization of inter-organizational business processes, the following phase of the internal-to-external interaction pattern focuses on the verification of the collaborating processes. Earlier, internal verification ensures that the business processes of collaborating parties enact correctly on their own. However, when intra-organizational processes are linked inter-organizationally, structural problems may occur. For example, deadlocks that prevent the correct termination of the overall business process.

In Figure 13, a subset of related eSourcing spheres are depicted in detail. The conceptual level of the service consumer shows a consumer sphere that is a subnet of an in-house process. Clouds denote abstracted details from 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. Only one depicted node 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, which is part of a conjoinment pattern (see [42–44]). 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 assembling a motor pump. When this node contains further assembly information, the truck producer makes a particular way of assembly mandatory for the service provider if it is assumed the consumer sphere exists before the provider 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 in Figure 13 is fully projected to the external. Thus, the content of the external level consumer contractual sphere and the consumer sphere on the conceptual level are isomorph, which corresponds to a white-box contractual visibility
pattern (see [42–44]). Consequently, the service provider has the options of responding either with a grey-box or a white-box projection in order to achieve a contractual consensus.

On the external level of Figure 13, the provider contractual sphere is isomorphic compared to the consumer contractual sphere, which means the collaborating parties have reached a contractual consensus. However, the service provider needs to fit the contractual sphere into the internal organizational setup. Additional tasks need to be carried out for manufacturing water-tank components before an assembly takes place and the final water tank needs to be forwarded to the site of the service consumer.

On the conceptual level of the service provider, the contractual sphere is refined by additional nodes, which means a grey-box contractual visibility pattern (see [42–44]) is used by the service provider. Figure 13 shows bold lined tasks that represent inserted nodes in the provider 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 finished 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.

The created eSourcing configuration of Figure 13 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 5. It is not necessary to collapse the eSourcing configuration for checking the correct termination. The reason is that the collaborating parties use a white-box and grey-box projection, which means that local checks of the in-house process and the provider sphere suffice to guarantee the overall eSourcing configuration has sound control-flow (see [43]). For the eSourcing configuration of Figure 13 these local verifications fail because the in-house process can not terminate correctly. The reason is a deadlock contained in the processes that are caused by the arcs of the $BT$-labelled node. As this node needs to wait for the $S$-labelled node to fire, it can never be enabled.

After the local checking of correct termination, the service consumer has to remodel the in-house process. The changed in-house process of Figure 14 has the $BT$-labelled

Fig. 14. Corrected in-house process.
node replaced with different conjoinment nodes that establish an exchange between the in-house process and the consumer sphere. Consequently, a repeated local check of the in-house process’ control flow by the service consumer succeeds.

**Monitorability negotiations:** After the inter-organizational processes are verified, the processes need to be linked for a synchronized enactment. In [42, 43, 47] such linking across the domains of collaborating parties is termed monitorability. Several patterns for monitorability and conjoinment were specified for linking processes into an eSourcing configuration.

![Fig. 15. Related eSourcing spheres in detail.](image)

In Figure 15 two monitorability constructs are used, namely token messaging and token propagation. Token messaging is used for connecting the *in*-labelled interface places. For an enactment application of the eSourcing configuration, token messaging means once the enactment of the in-house process has reached the consumer sphere, such a
state is messaged across the organizational domains and the enactment of the provider’s provider 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 out-labelled interface places of the provider sphere and the consumer sphere via the external level. In the eSourcing configuration of Figure 15 this means the enactment of the provider 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 component (see Section 5.3). 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 monitorability constructs need to be further realized by appropriately linking the local WFMS and rules engine on the internal level to the conceptual level. When this setup is completed, the enactment of the eSourcing configuration commences.

6.4 Evaluation Results

As a final step, the functional requirements listed in Section 4.5 and the predominantly non-functional requirements of Table 1 are discussed. First, the requirements of Section 4.5 are revisited.

- For requirement a, the case study establishes that eSRA incorporates the three-levels framework [28] and distributes components so that the functionality supports the conceptual purpose of each level.
- Requirement b is fulfilled as the conceptual level of eSRA comprises components for modelling business processes and business rules that can be stored in and retrieved from databases that are located on the same level.
- For requirement c, the trusted-third-party component and its nested components permit a brokering of service requests and service offers if collaborating parties are anonymous and first need to go through a bidding process for establishing a collaboration.
- The case study does not show adequately how requirement d is fulfilled by eSRA as a static assignment is chosen where the parties know in advance that they collaborate with each other. Still, eSRA provides components that caters for functionality that enables a bidding process.
- For requirement e, eSRA provides contracting components that are replicated on the respective external levels of each collaborating party. These contracting components communicate with each other through coordination-interface components that bridge the gap between organizational domains. The projections of business processes are supported by components that enable a translation between the conceptional and external level. The trusted-third party comprises components for independently verifying whether an external-level process matching has been achieved. Furthermore, the case study points out that eSRA also enables a negotiation about the extent of enactment monitorability within an eSourcing configuration.
To fulfill requirement $f$, eSRA comprises on the conceptual level of each collaborating party components for the evaluation and verification of business processes that are part of an eSourcing configuration. Additionally, the trusted-third-party component also contains a component for verifying the correct termination of an eSourcing configuration and for verifying independently whether the service provider internally adheres to the service provision that is externally agreed upon.

Finally, for requirement $g$, eSRA comprises of components that enact in a coordinated way business processes and business rules on the internal level of each collaborating party. These components orchestrate services that wrap legacy systems. To coordinate the orchestration of legacy systems across the domains of collaborating organizations, eSRA contains additional components on the external level of each collaborating party that communicate with each other through the coordination interfaces.

An assessment of the predominantly non-functional requirement of Table 1 is presented below. The evaluation results are listed in an order that corresponds to the table.

- The case study shows it is feasible to use applications that are modelled after eSRA for inter-organizational business process collaboration. This statement holds as the case study itself is inspired by a bigger CrossWork project [1] related case study.
- The scalability of eSRA is demonstrated in the case study as it is possible to unite one service consumer with several service providers in one eSourcing configuration. To do so, an in-house process needs to contain several consumer spheres that are separately projected to the external level. Hence, for each consumer sphere, a separate external-level consensus between contractual spheres is backed by a respective provider sphere on several independent conceptual levels.
- The inter-operability requirement is fulfilled for eSourcing as the case study in Section 6.3 shows how an OEM and the service providers are able to reach a collaboration consensus from a business, conceptual, and technological point of view.
- The requirement of coherence is fulfilled because in an established eSourcing configuration that the external-level tasks are complemented by conceptual-level business processes where labels match with the external-level contractual spheres. In eSRA the verification component contained in the of the trusted third part is instrumental for ensuring such coherence.
- With respect to applicability, it needs to be mentioned that the case study of Section 6.3 is realized with a proof-of-concept prototype [40] that implements a subset of eSRA components. However, for a total evaluation of eSRA applicability, the full set of specified components would need to be implemented and tested.
- The completeness of eSRA is confirmed by the case study for the direction pattern internal-to-external in combination with the assignment pattern static [42, 43, 47]. For completeness, it is desirable to perform an eSRA evaluation with all combination of direction and assignment patterns. However, for this paper such a broad evaluation is not possible.

In summary, the evaluation results of all functional and non-functional requirements confirm in a satisfactory way the validity of eSRA as a reference architecture for guiding the development of e-collaboration setup systems. Given the numerous combination
options of interaction patterns and the amount of eSRA components, the limited scope of this case study needs to be complemented by evaluations with a proof-of-concept prototype that implements the full spectrum of eSRA components. In the CrossWork project, the exploitation phase may result in such a complete prototype that may be applied in real industry cases. Furthermore, the concept of eSourcing is also adopted in the SOAMeS project [6] where it is the objective to investigate interoperability management at a process and pragmatic level. In SOAMeS, non-functional aspects are addressed for supporting business strategy needs. Part of the work is performed in industry-partner case projects where eSRA is input for introducing a service-oriented architecture on top of the existing infrastructure of legacy systems. Hence, a second option for an enhanced eSRA evaluation is available.

7 Conclusion

This paper focuses 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-levels 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 has been 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.

As an adapted version of scenario-based evaluation methods for reference architectures, a top-down pattern-exploration approach is chosen for investigating interactions during the setup phase between different business parties for the establishment of inter-organizational 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. Each interaction-pattern comprises a scenario example 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-levels 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. The case study is used to show how a set of functional and non-functional requirements is supported by the eSourcing reference architecture.

Scope for further analysis exists for the dimensions and values mentioned in the logical interaction-pattern space of this paper. 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 the
logical interaction-pattern space 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 currently existing logic interaction-pattern space. The latter dimension represents the number of collaborating parties in the role of service consumer or service provider. Adding extra interaction dimensions leads to an extension of the interaction-patterns catalogue, which potentially results in an extension need of the architecture with further components.

References