

2 Cross-Organisational Deployment of Business Processes

In this chapter, we take a deeper look at the problem area and provide a comprehensive example from the field of supply chain management that will be used throughout the book, particularly in Chapters 4, 5, and 7, to explicate various aspects of the approaches introduced herein. Readers interested in an overview of the results presented may browse this chapter to only get an impression of the problems motivating the methods and procedures described in this book and to get an idea of the running example also used in other chapters.

Collaborative business processes (CBPs) as defined, for instance, by Coetzee and Eloff²² denote business processes that span organisational boundaries in order to support business interactions involving cross-organisational workflows. Modelling of such CBPs has been discussed, for instance, by Lippe *et. al.*²³. The definition of workflow views that provide as much information as required to allow for specification of CBPs assuring cross-organisational interoperability, but at the same time as less information as possible about the internal aspects of a workflow as implemented by a particular partner in a CBP is considered essential²⁴. Approaches to provide the required functionality for cross-organisational workflows based on SOA are considered particularly beneficial²⁵. SOC established by those approaches facilitates the definition of CBPs. Standardised BPDs play a central role in the definition of CBPs due to their ability to specify business processes on top of Web services and their platform-independency²⁶. Though currently the definition of executable business processes across organisational boundaries seems not yet to have found much interest in research, using standardised BPDs particularly for this purpose would exploit the capability offered by a standard more than is done currently. This consideration will be explained in more detail below.

In an SOC environment, first the situation is considered where the task of defining CBPs and related enhanced Web services using a BPD is distributed between several nodes in different organisations. This state-of-the-art employment of a BPD, where each organisation engaged in a particular CBP defines on its own the respective business processes or enhanced Web services executed within their system, is

²² Coetzee and Eloff, 2003

²³ Lippe *et. al.*, 2006

²⁴ Dickson *et al.*, 2004

²⁵ Papazoglou and van den Heuvel, 2007

²⁶ Sayaha and Zhang, 2005

depicted in Figure 2. Of course, agreement on the overall task of the CBP has to be achieved between the organisations involved.

Figure 2 illustrates an exemplary environment for the distributed development and execution of a BPD- defined collaborative business process, with two systems residing in two different domains A and B. Each node depicted in Figure 2 is supposed to belong to a different organisation, but still is capable of running processes defined in a particular BPD.

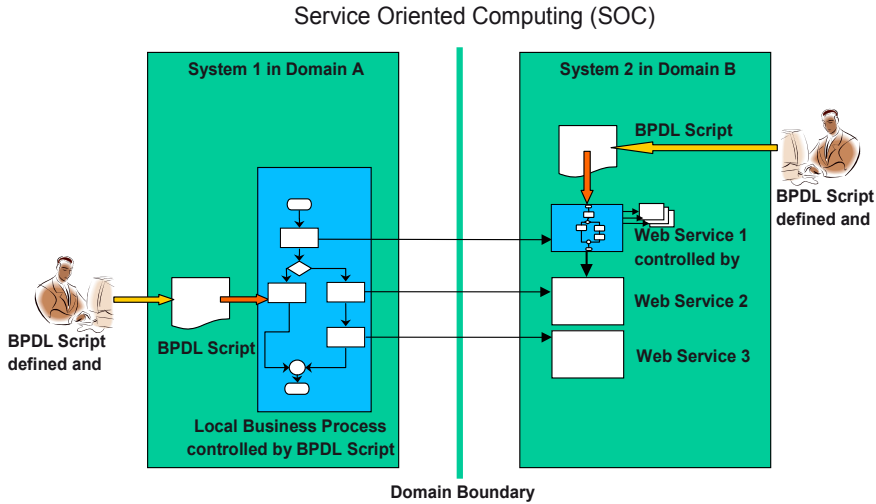


Figure 2: Collaborative Business Process Using Locally Defined Subprocesses

Consider the case where in domain A there is a need for a CBP, for instance, in a supply chain application, requiring information I_A offered by a Web service W_2 at system 2 in domain B. Because of restrictions imposed by security policies in domain B, Web service W_2 would not be allowed to be accessed directly from outside domain B, because, for instance, it provides further information besides I_A that must not be leaked from domain B. For solving this conflict with security policy restrictions, a conventional approach would be the provision of an enhanced Web service in domain B, say W_1 at system 2. W_1 would access the information required from Web service W_2 and offer the non-restricted part of the results (*i.e.*, I_A) to system 1 in domain A across the domain boundary. Since a business process defined by a BPD script offers services to its environment, it can itself be considered a Web service. Therefore, in this example W_1 is assumed to be defined by a BPD script S_1 .

2.1 Extended Use of Business Process Definition Languages in CBP Scenarios

In order to allow for fast development and adaptation to changing requirements of business processes, it would be desirable to concentrate the definition of all business processes and enhanced Web services at one particular node and distribute the BPD script resulting from this location to other nodes for execution. This would reduce the coordination overhead implied by distributed definition of the parts of the CBP in different organisations and, therefore, could help to save time and to increase flexibility during the specification and implementation of CBPs.

Since the need for the particular business process in this example arose in domain A, it is very probable that also requests for changes to this business process will arise in this domain. In order to circumvent the requirement that requests for change arising in domain A must be presented to developers in domain B in order to have them change the Web service W_1 , it would be conceivable that W_1 running on behalf of a business process in domain A will be defined by developers in domain A. The defining BPD script S_1 will subsequently be brought to execution at system 2 in domain B as indicated by the arc from the developer workstation at domain A to domain B in Figure 3.

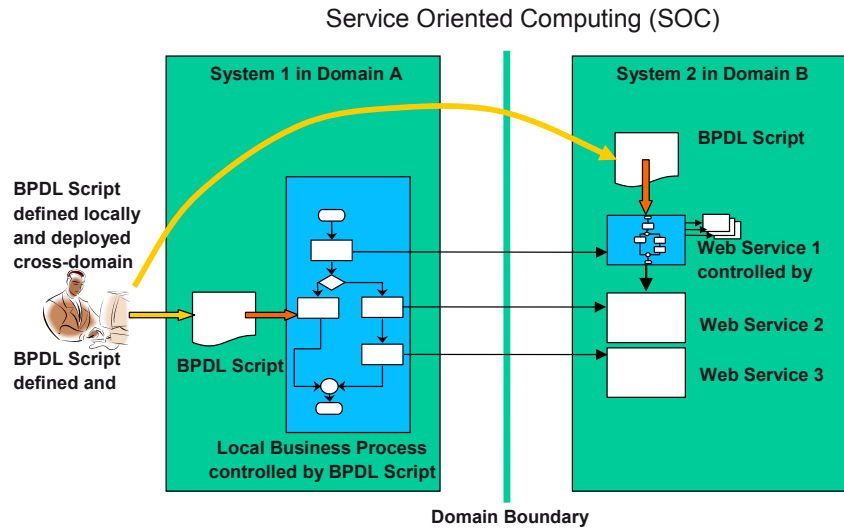


Figure 3: Cross-Organisational Deployment of Business Processes

This approach would greatly facilitate the adaptation of W_1 in domain B to changing requirements originating in domain A. Given both systems are based on a BPD-enabled platform using the same preferably standardised BPD, this scenario, as depicted in Figure 3, would be technically feasible. However, it would induce severe security weaknesses in domain B, if S_1 would be executed in domain B without particular precautions. Prior to running S_1 , it has to be determined whe-

ther the semantics of W_1 defined by S_1 comply with security policies effective in domain B. Unless these security issues can be solved, it may prevent this extended use of BPDL scripts from being actually applied in a real-world cross-organisational environment.

2.2 Motivating Example of Cross-Organisational Business Process

In Figure 4, an example from the area of supply chain management (SCM) is shown that will be used to illustrate the security issues arising when remotely defined BPDL scripts are being deployed across enterprise domain boundaries.

Supply Chain Management

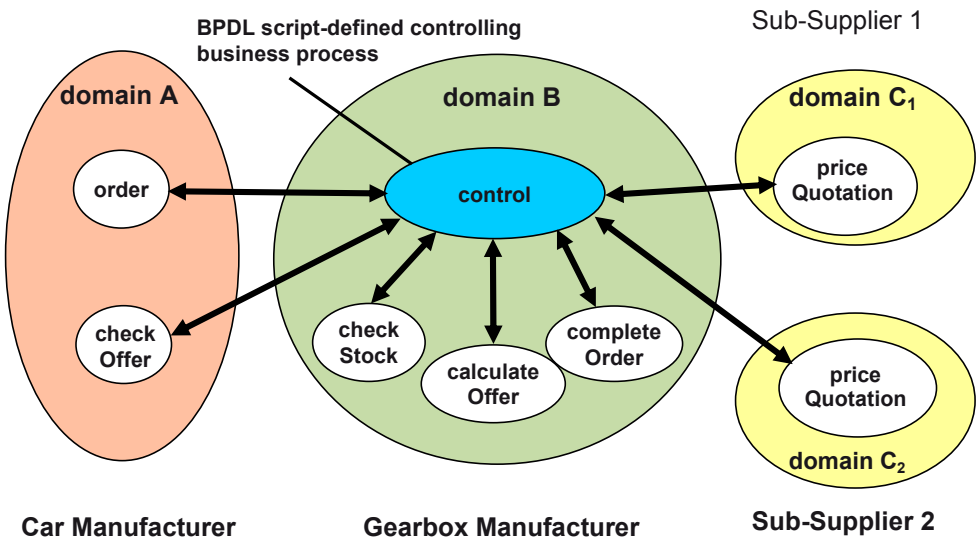


Figure 4: Collaborative Business Process Example

The application context of the distributed business process depicted in Figure 4 is the order processing of a car manufacturer ordering gearboxes or components thereof from a gearbox manufacturer who, in turn, orders components for gearboxes from different sub-suppliers (two in this example).

The business process is set up in a SOC environment where all functions used for the application are provided as Web services and the composition of Web services is accomplished using BPDL for the definition of the controlling workflow. In such a context, BPDL scripts are not required to perform any kind of data manipulation or data processing. Since standardised BPDs usually do not provide any language constructs for performing data manipulations, constructs of other languages such

as XQuery²⁷, XSLT²⁸, and XPath²⁹ would have to be imported for these purposes. Since in a service oriented application context, all data processing can be kept outside the controlling BPD scripts, it is assumed that the BPD scripts considered here only make use of elements imported from XPath, if any, in expressions specifying conditions for flow control purposes not implying any data manipulation.

2.2.1 Description of Business Process Example

In this example, a BPD script is executed in a system of the gearbox manufacturer defining a controlling business process denoted by `control` in Figure 4. An order process of the car manufacturer (that may itself be a Web service or a BPD-defined business process) invokes the Web service offered by the `control` process at the gearbox manufacturer providing a list of gearbox components to be ordered by the car manufacturer. Before placing an order, the car manufacturer expects a price offer accompanied by a commitment with respect to the delivery date.

The `control` process invokes a `checkStock` Web service for checking the availability of the ordered items in stock. For this purpose, the list of items to be ordered is passed to this Web service. After checking the availability in stock, the `checkStock` Web service returns two lists of items that are to be ordered from sub-supplier 1 and sub-supplier 2, respectively. Together with these lists of items, a transaction ID for the order in progress and the credentials required to invoke the respective Web services of the two sub-suppliers are returned by `checkStock`.

Upon receiving the response of `checkStock`, the `control` process invokes the `priceQuotation` Web services of the sub-suppliers and provides the respective list of items to each of them. In order to get access to these Web services, the credentials returned by `checkStock` are used by the `control` process. Of course, each Web service of the two sub-suppliers requires its own set of credentials. Therefore, the `control` process has to provide the proper instance of credentials to each of them.

After checking availability of the items on the respective list, each `priceQuotation` Web service returns a list augmented by prices and availability on stock or dates of delivery. The `control` process then invokes a `calculateOffer` Web service of the gearbox manufacturer to prepare an offer for the car manufacturer. For this purpose, the `control` process passes the augmented lists returned from both sub-suppliers to the `calculateOffer` Web service together with the transaction ID that was returned to it before by the `checkStock` Web service.

The `calculateOffer` Web service uses the transaction ID to identify the proper order request of the car manufacturer and to find the information relating to this

²⁷ Boag *et al.*, 2007

²⁸ Kay, 2007

²⁹ Berglund *et al.*, 2006

order in the data base of the gearbox manufacturer provided there by the `checkStock` Web service. For instance, information about items found to be available in stock and potentially reserved for this order by the `checkStock` Web service could be identified by the `calculateOffer` Web service in the course of its processing. Finally, the offer is returned to the `control` process and will be passed to a `checkOffer` Web service of the car manufacturer. This Web service will return an 'OK' or 'Reject' response to the `control` process after having checked whether the offer would be acceptable to the car manufacturer.

The response from the `checkOffer` Web service is passed to a `completeOrder` Web service by the `control` process. Depending on the type of response, this Web service either completes the order processing within the gearbox manufacturer if the response was 'OK' or discards all intermediate information such as items reserved for this transaction ID if the response was 'Reject'.

After the `completeOrder` Web service has terminated its task, it returns a corresponding result to the `control` process that, in turn, provides this result to the `order` Web service of the car manufacturer as a response to its own invocation, thereby completing the workflow of this business process.

For the purpose of this discussion it is supposed that the `control` process could be specified by the car manufacturer as a BPDF script and sent to the gearbox manufacturer for execution within his domain. Reasons for doing so could be the ability to better adapt the order processing with respect to the communication requirements between the car manufacturer and the gearbox manufacturer, and to react faster to changing requests concerning the workflow on the side of the car manufacturer, thereby providing more flexibility for definition of the collaborative business process to the car manufacturer. However, unless the security issues related with this approach (as discussed in the next section) could be solved satisfactorily, the gearbox manufacturer would not accept this remotely defined BPDF script for execution.

2.2.2 Security Policy-Induced Restrictions in Cross-Organisational Business Process Execution

When the controlling BPDF script is brought in from the car manufacturer for execution in the domain of the gearbox manufacturer, the processing performed by the controlling business process will be subject to several restrictions derived from security policies of the gearbox manufacturer.

The security policies of the gearbox manufacturer may mandate that the list of items that are not in stock and, therefore, have to be ordered from the sub-suppliers may not be disclosed to the car manufacturer and the respective competing sub-supplier for strategic reasons. The same will obviously hold for the credentials required for granting access to the `priceQuotation` Web services of the sub-suppliers. This information may only be passed to the respective sub-supplier,

but neither to the competing sub-supplier nor to the car manufacturer for obvious security reasons.

Other restrictions in the example of Figure 4 may require that the list containing prices and delivery dates for the items to be ordered returned by each of the sub-suppliers has to be passed unmodified to the `calculateOffer` Web service while the respective source of the lists may not be confused in order to allow for proper calculation of an offer to the car manufacturer. Finally, this information may also not be disclosed to the car manufacturer or the competing sub-supplier. Furthermore, it may be required that the offer containing prices and delivery dates returned from the `calculateOffer` Web service is passed to the `checkOffer` Web service of the car manufacturer without any modification in order to prevent manipulation of this offer, for instance, by changing the committed delivery date or the prices.

A further type of possible restrictions implied by security policies of the gearbox manufacturer may require that particular input parameters of a Web service may not be used, or only be used with a restricted range of allowed values if invoked in a BPDF script imported from the car manufacturer. An example of such restrictions could be the `calculateOffer` Web service that could have a further input parameter for controlling the type of rebate to be taken into account in calculating the offer. If invoked by the `control` process defined by the car manufacturer, this parameter may be forbidden to be used at all or may be restricted to one or a few values. Without such a restriction, the car manufacturer could define in the `control` process any amount of rebate the `calculateOffer` Web service is able to provide, even though the gearbox manufacturer usually would only allow a specific amount of discount to this car manufacturer.

2.3 Security Issues Related to Cross-Organisational Deployment of CBP

As can be seen from the foregoing discussion, many restrictions refer to non-disclosure of information passed between Web services within the `control` process to destinations outside the domain of the gearbox manufacturer, in particular to the car manufacturer. The latter restriction is of special interest facing the fact that in the example above this very process is defined by exactly the car manufacturer who is restricted to get some of the information handled by this process.

Other restrictions mandate that the values passed to a Web service originate from a particular other Web service or lie in a particular range of values.

Access control to Web services³⁰ and in particular role-based access control (RBAC)³¹ may only cover part of these restrictions. If addressed by access control

30 e.g. Abendroth and Jensen, 2003; Dimmock *et al.*, 2004

means alone, enforcement of non-disclosure of information outside the local domain would imply that access to the particular information would not be granted to any principal outside the local domain. In this example, the car manufacturer residing outside the local domain of the gearbox manufacturer and, as a consequence, also the BPDF script defined and invoked by the car manufacturer, would not be granted access to visibility restricted information thus preventing the control process of this example being remotely defined and deployed by the car manufacturer.

Relaxation of access restrictions, such as granting access provided the values passed to a Web service come from a particular source (for instance values returned from a specific Web service), require information flow (in backwards direction) to be considered for making decisions upon granting access or not. Thus, information flow analysis has to be applied in addition to purely access control-oriented approaches, in order to cope with this kind of restrictions.

Restrictions with respect to visibility of values returned from a Web service are also not covered by access control measures alone. For this purpose, information flow (in forward direction) has to be analysed, that means the future use of the values returned has to be taken into consideration.

With the approach proposed here, these restrictions derived from the security policies at the location where a remotely defined BPDF script will be executed can be enforced using the methods proposed, including the restriction that the author of the BPDF script is precluded from getting knowledge about information handled by the business process he has specified. At first sight, the last one may seem to be a restriction contradictory to itself. However, it will turn out during the course of further consideration that it is possible to grant access in such a fine-grained manner to a remotely defined business process and enforce the implied information flow restrictions in a straightforward manner at the executing site.

The security issues arising when remotely defined BPDF scripts are to be executed can be condensed into the following questions:

- 1) Will the business process defined by the BPDF script comply with the security policies of the executing domain?
- 2) Which access privileges are required in order to grant access to the business process defined by the BPDF script (*i.e.*, to possess the proper privileges for accessing the resources, particularly the Web services, encountered in the course of its execution)?

The second question is addressed by research considering access control³², even though not necessarily in the context of BPDF script execution³³. The first question,

31 Ferraiolo *et al.*, 2001; Peng and Chen, 2004

32 Koshutanski and Massacci, 2003; Mendling *et al.*, 2004

however, is comparatively novel since execution of remotely defined BPDF scripts seems currently not to be in practical use very much and, therefore, security aspects inherent in this way of using BPDF have not found a lot of attention in research, yet.

The research leading to the results presented in the following chapters has strived to propose a framework for coping with these novel security aspects arising from the employment of standardised business process languages. To this end, semantic aspects of the business processes defined by their respective scripts written, for instance, in BPDF, are considered at the time, a new script is to be deployed to a node across security boundaries. However, the analysis of the semantics of code written in programming languages is a well-known difficulty³⁴. Therefore, the need to analyse the semantics of a BPDF-defined business process with respect to involved security-relevant semantics would make the approach to specify a business process remotely from the location of execution impractical unless this analysis can be provided automatically, at least to a large extent. The methodology proposed will make use of the fact, that business process languages offer little to none means for defining data processing or computational tasks as part of the language itself, but rather have to invoke Web services for these purposes or must import constructs from expression languages defined in other XML standards such as XSLT³⁵, XQuery³⁶, or XPath³⁷.

The extended use of business processes as proposed in Section 2.1 and parts of the framework for assessing compliance of remotely defined BPDF scripts described in chapters 4 through 6 have been published in several workshop, conference, and journal papers³⁸.

2.4 Limitation of Scope to WS-BPEL without Loss of Generality

For the definition of Web services, Web Services Description Language (WSDL) 1.1³⁹ (expected to be gradually substituted by its newer version WSDL 2.0⁴⁰) has been established by the World Wide Web Consortium (W3C) as a single standard broadly accepted for the definition of Web services.

³³ Joshi *et al.*, 2001

³⁴ Cousot 1999

³⁵ Kay, 2007

³⁶ Boag *et al.*, 2007

³⁷ Berglund *et al.*, 2006

³⁸ Fischer *et al.*, 2005, Fischer *et al.*, 2006, Fischer *et al.*, 2007a, Fischer *et al.*, 2007b, Fischer *et al.*, 2007c

³⁹ Christensen *et al.*, 2001

⁴⁰ Chinnici *et al.*, 2007

In contrast, for business process definition languages (BPDs) several approaches to standardisation have been taken by different vendor groups and standardisation organisations, leading to a plurality of standards:

- Web Services Business Process Execution Language (WS-BPEL), formerly known as Business Process Execution Language for Web Services (BPEL4WS or BPEL for short)⁴¹, propagated by the Organisation for the Advancement of Structured Information Standards (OASIS),
- Business Process Modelling Language (BPML)⁴², propagated by the Business Process Management Initiative (BPML.org); since merger between BPML.org and Object Management Group (OMG) in 2005, standardisation activities for BPML has been dropped in favour of BPEL,
- XML Process Definition Language (XPDL)⁴³, propagated by the Workflow Management Coalition,
- Web Services Choreography Interface (WSCI)⁴⁴, propagated by the World Wide Web Consortium (W3C),
- ebXML Business Process Specification Schema⁴⁵, propagated by UN/CEFACT and OASIS, and
- Business Process Model And Notation (BPMN), Version 2.0⁴⁶, initially propagated by BPML.org; after merger with OMG in 2005, BPMN is supported and has been standardised as Version 2.0 by OMG.

Though the existence of several parallel standards aiming at the same goal, in general, adversely affect the very purpose of standardisation, the different standards at least have some obvious commonalities, as all languages except of BPMN are script-based using XML⁴⁷ and facilitate the composition of business processes by invocation of Web services and definition of the communication with other parties (in particular human participants) involved in a business process.

Among these standards, BPMN plays a special role since it provides a graphical notation for representing business processes to be specified and may be best compared with UML as a modelling tool. Being a graph-based notation for business processes, BPMN is considered better suited for business analysts. However, using it without special tools to draw the graphic required seems to be very complicated and could be considered nearly impossible. Further, in order to become executable, BPMN-based process specifications need to be mapped to another representation such as XPDL or BPEL. The BPMN standard comprises a mapping to BPEL, al-

⁴¹ Alves *et al.*, 2007

⁴² Arkin, 2002

⁴³ Workflow Management Coalition, 2008

⁴⁴ Arkin *et al.*, 2002

⁴⁵ Dubray *et al.*, 2006

⁴⁶ OMG, 2011

⁴⁷ Bray *et al.*, 2006

though it has been argued that not all process modelling supported by BPMN can be mapped to BPEL. For this reason, some people particularly in favour of BPMN argue that BPMN is to be considered superior to BPEL⁴⁸. In spite of these arguments, many authors still rate BPEL as the de facto standard for defining business processes, at least, executable ones⁴⁹.

The fact that several business process languages exist in parallel, gave rise to research as to which extent these languages are comparable with respect to their semantic expressiveness⁵⁰. In particular, Aalst *et al.*⁵¹ and Wohed *et al.*⁵² analysed different languages (*i.e.*, WS-BPEL, BPML, WSCI and some vendor-specific business process languages) with respect to workflow and communication patterns. The results of their work indicate that, to a large extent, the different languages are capable of expressing the same semantics with respect to workflow control and communication behaviour.

As was to be expected from these results, the different languages may be convertible to each other as has been shown in an exemplary manner for XPD L and WS-BPEL by Fischer and Wenzel⁵³.

In other work⁵⁴, a process ontology based on multiple meta-models derived from different existing workflow models is introduced in order to facilitate mappings between choreography descriptions defining possible interactions between different partners of a CBP and internal workflows of the partners defined using different workflow languages, workflow models, and choreography languages. One successful mapping (with manual intervention) between a vendor-specific workflow definition language and Abstract BPEL (specification of abstract, *i.e.*, not executable, business processes defined in BPEL) is reported. Although successful automatic conversion between different workflow definition languages and BPD Ls based on their proposition does not seem to have been published ever since, this approach and the preliminary results reported so far underpin the assumption that mapping between different workflow definition languages and different BPD Ls is possible and conversion of business process specifications based on a particular BPD L may be converted into a semantically equivalent specification based on another BPD L. In addition, this conversion has been shown to be performed automatically, at least up to a certain extent.

Given the fundamental similarity of all different languages used for business process definition and their potential to be converted to each other, the scope of novel

48 Vigneras, 2008; Swenson, 2008

49 *e.g.*, Ouyang *et al.*, 2009, Merouani *et al.*, 2010

50 Aalst *et al.*, 2002; Shapiro, 2002; Wohed *et al.*, 2002

51 Aalst *et al.*, 2002

52 Wohed *et al.*, 2002

53 Fischer and Wenzel, 2004

54 Haller *et al.*, 2006; Haller and Oren, 2006

methods presented here has been focussed on one particular representative, namely WS-BPEL. In 2007, WS-BPEL has been accepted by the Organization for the Advancement of Structured Information Standards (OASIS) as an OASIS standard⁵⁵. Prior to this, through support by prominent vendors like IBM, Oracle, BEA, Microsoft, SAP, and Siebel, WS-BPEL already had emerged as the de-facto standard for business process definition⁵⁶.

According to common practice, for the remainder of the book BPEL will be used as a short-hand for WS-BPEL.

2.5 Summary

In this section, the main goal of the research that led to the results presented in the following chapters, namely the assessment of compliance of remotely defined BPEL scripts with local security policy, has been motivated by discussing some of the security issues arising by executing remotely defined BPEL scripts. A comprehensive example from the field of supply chain management has been introduced for this purpose, which will also be used in other chapters for explication of various aspects of the approaches presented. The example has shown that availability of fine-grained access control and information flow control may enable sensible tasks to be performed by remotely defined business processes without jeopardising local security policy.

Since the different standardised BPDs are comparable with respect to their expressiveness for specifying the process logic of a business process and BPEL is considered the de-facto standard for specification of executable business process as underpinned by related work discussed above, without loss of generality, the scope of consideration will be restricted to BPEL in the remainder of the book.

⁵⁵ OASIS, 2007

⁵⁶ Wang *et al.*, 2004; Mayer and Lübcke, 2006

Information Flow Based Security Control Beyond RBAC
How to enable fine-grained security policy enforcement
in business processes beyond limitations of role-based
access control (RBAC)

Fischer-Hellmann, K.-P. - Bischoff, R. (Ed.)

2012, XXI, 161 p. 25 illus., Softcover

ISBN: 978-3-8348-2617-6