A Security Infrastructure for Cross-Domain Deployment of Script-Based Business Processes in SOC Environments

K.P.Fischer[1,3], U. Bleimann[1], W. Fuhrmann[1], S.M Furnell[2,4]

[1] Aida Institute of Applied Informatics, University of Applied Sciences Darmstadt, Germany
[2] Network Research Group, University of Plymouth, Plymouth, United Kingdom
[3] Digamma Communications Consulting GmbH, Darmstadt, Germany
[4]School of Computer and Information Science, Edith Cowan University, Perth, Australia
e-mail: K.P.Fischer@digamma.de

Abstract

This paper addresses security aspects arising in service oriented computing (SOC) when scripts written in a standardized scripting language such as WS-BPEL (formerly: BPEL4WS or BPEL for short), BPML, XPDL, WSCI in order to implement business processes on top of Web services are deployed across security domain boundaries. It proposes an infrastructure and methods for checking the scripts deployed, prior to execution, for compliance with security policies effective at the domain in which a remotely developed script-based business process is to be executed.

Keywords

Security Policy, Policy Enforcement, Service Oriented Computing (SOC), Service Oriented Architecture (SOA), Business Process Management, Web Services, Web Services Business Process Execution Language (WS-BPEL)

1. Introduction

Service oriented computing (SOC) is currently considered one of the most promising new paradigms for distributed computing (Papazoglou and Georgakopoulos, 2003). Though comparatively new, a significant amount of research has already been dedicated to this area (e.g. Deubler *et al.* 2004). Web services, and the composition or orchestration of them, play a central role in current approaches to service oriented computing (Berardi *et al.* 2003). Service orientation is also expected to have an important influence in the area of grid computing, where the provisioning of computing resources within a conceptual huge network of collaborating computers and devices can also be fostered by services (so called grid services in this context) provided by different nodes (Tuecke *et al.* 2003).

In service oriented approaches using Web services a layered architecture for composing new services from existing services or for executing processes based on existing services has emerged (Medjahed *et al.* 2003). The request for fast adaptation of enhanced services and processes to changing requirements as well as the request to avoid dependency on certain platforms (vendor lock-in) lead to the specification of platform independent, standardized process definition languages for the definition of enhanced Web services or business processes in the top layer of this architecture. However, several different standardization approaches for such a language have been taken, 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) (Arkin *et al.* 2004), Business Process Modelling Language (BPML) (Arkin, 2002), XML Process Definition Language (XPDL) (Workflow Management Coalition 2002), Web Services Choreography Interface (WSCI) (Arkin *et al.* 2002), and ebXML Business Process Specification Schema (Malu *et al.* 2002). Though the existence of several parallel standards aiming at the same goal detracts from the very purpose of standardization, the different standards at least have some obvious commonalities, as all languages are script based using XML 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. It should be noted that a business process defined using one of these languages can itself be considered a Web service from the point of view of external communication parties.

The existence of several business process languages gave rise to research as to which extent these languages are comparable with respect to their semantic expressiveness (Aalst *et al.* 2002; Shapiro, 2002; Wohed *et al.* 2002). In particular Aalst *et al.* (2002) and Wohed *et al.* (2002) 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 a result these languages could be expected to be convertible to each other as has already been shown in an exemplary manner for XPDL and WS-BPEL by Fischer and Wenzel (2004). Given the fundamental similarity of the different languages used for business process management, without loss of generality we will concentrate our proposition on one particular representative, namely WS-BPEL propagated by the Organization for the Advancement of Structured Information Standards (OASIS). For the remainder of this paper we will use BPEL as a short-hand for WS-BPEL.

2. Security Issue in Cross-Domain Business Process Definition

As security already is an important issue in distributed applications in general, this topic is also of paramount importance for the application of business process languages. Security of Web services is well studied and several approaches for access control to Web services exist (e.g. Nadalin *et al.* 2004; Abendroth and Jensen, 2003; Dimmock *et al.* 2004). Koshutanski and Massacci (2003) and Mendling *et al.* (2004) are considering security aspects in the context of employing business process languages, in particular BPEL. While access control related aspects are predominant with Web services and are, of course, also an issue with business process languages, further security aspects arise from the employment of standardized script languages such as BPEL. From their nature of being standardized and platform-independent, these languages involve the capability of defining business processes across platforms. Use of this capability introduces new security issues that have not been present in Web services before the business process languages came in. By employing standardized business process languages it will be feasible to define a business process at one location and execute it at a different location. It is conceivable that the two locations belong to different security domains within the same or different organisations or corporations. The new security issue arising from this approach leads to, but is not limited to, the following questions:

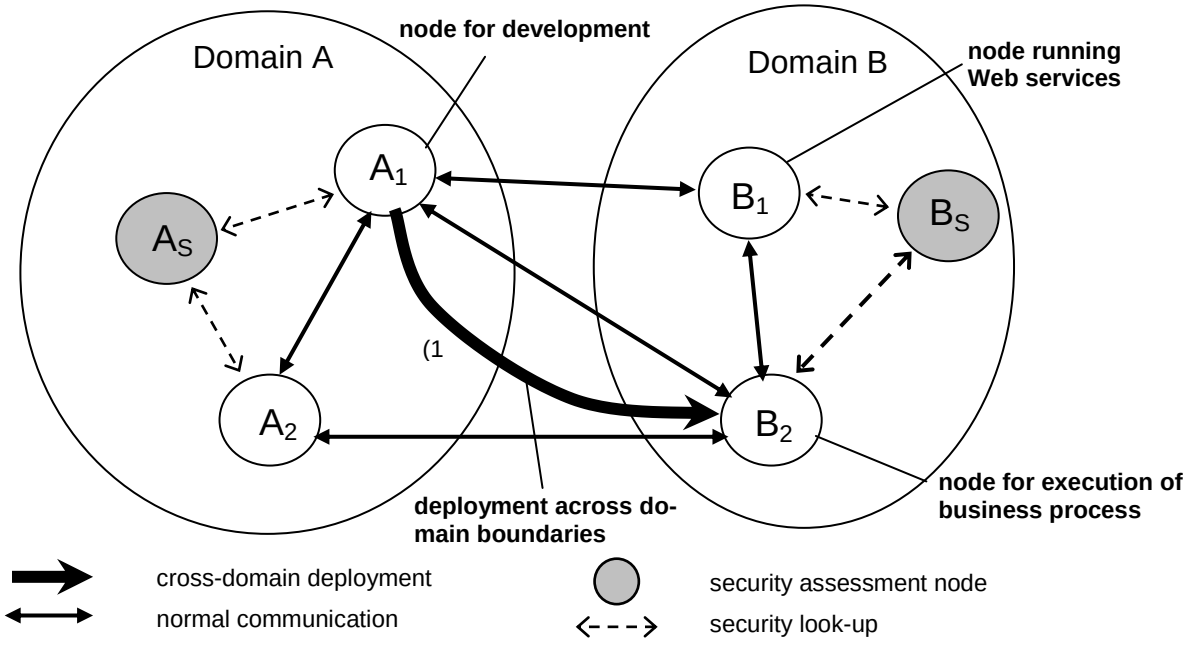
- Are the semantics of a remotely defined business process or enhanced Web service compatible with the security policy effective at the node where it is to be executed?
- Which classification, with respect to access control, is required for the Web service offered by the remotely defined business process or enhanced Web service in order to be compliant with the security policy in the domain the executing node belongs to?

While the second question again arises in the context of access control, albeit from a different point of view to that which usual access control approaches address, the first point addresses a completely new security issue, that, by its nature, had not needed to be considered in the context of Web services as it is not relevant with their basic incarnation.

In this paper we propose an infrastructure and a methodology for coping with the first one of these novel security aspects arising from the employment of standardized business process languages. We consider semantic aspects of the business processes defined by their respective scripts. The methodology proposed makes use of the fact, that business process languages offer little or no 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 other XML standards such as Xpath (Berglund *et al.* 2004).

3. Infrastructure for Distributed Development and Execution of Business Processes: An Example

In an SOC environment we consider the situation where the task of defining business processes and enhanced Web services using BPEL is concentrated at a particular node and distributed to other nodes for deployment and execution.



- Web service W_1 running on node B_1 being restricted not to be invoked from outside domain B
- Security policy of domain B allows for carrying a subset of information I_A provided by W_1 to domain A

Figure 1 - Distributed Development and Execution of Business Processes in SOA

Figure 1 illustrates an exemplary environment for the distributed development and execution of a BPEL script with six nodes residing in two different domains A and B. We further suppose that each node depicted in Figure 1 is capable of running BPEL-defined processes.

We consider the case where in domain A there is a need for a business process, e.g. in a supply chain application, requiring information I_A offered by a Web service W_1 at node B_1. However, because of restrictions imposed by security policies in domain B this information cannot be accessed directly from outside 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_2 at node B_2. W_2 would access the information required from Web service W_1 and offer the non-restricted part of the results, i.e. I_A, to nodes in domain A across domain boundaries. While it would be possible not to use any business process language for this purpose, we assume that W_2 is defined by a BPEL script S_2.

Since the need for the particular business process in this example arose in domain A, there is some probability 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_2, it would be conceivable that W_2 running on behalf of any node in domain A will be defined at node A_1 and the defining BPEL script S_2 will subsequently be brought to execution on node B_2 as indicated by the arc (1) from A_1 to B_2.

This approach would greatly facilitate the adaptation of W_2 in domain B to changing requirements originating in domain A. However, it would induce severe security weaknesses in domain B, if S_2 would be executed in domain B without particular precautions. Prior to running S_2, it has to be determined whether the semantics of W_2 as defined by S_2 comply with security policies effective in domain B.

The analysis of the semantics of code written in programming languages is a well-known difficulty (Cousot, 1999). Therefore, the need to analyse the semantics of W_2 with respect to security-relevant semantics will make this approach of cross-domain definition and execution impractical unless this analysis can be provided automatically, at least to a large extent.

Fortunately, the nature of BPEL (as well as of other business process languages) accommodates this analysis, further supported by the fact that no thorough analysis of each and every particular aspect of the semantics will be required, but instead only a direct search for features violating the security policy of the target domain. To further facilitate this analysis the security policy of the target domain may be expressed with respect to potentially security critical features of the language being used, i.e. BPEL in our example. Given these preconditions, the task of analysis becomes appropriate to be performed automatically, at least in cases when it is sufficient to express the security policy of a domain in the way exemplarily described in the following section.

4. Security Policy Definition for Business Process Analysis

Based on concepts developed in conformance testing methodology of open systems as defined in a series of International Standards (CTMF, Conformance Testing Methodology and Frame-

work, ISO/IEC 9646, part 1–7; particularly (ISO, 1994)), after analysis of the security-relevant features of BPEL a checklist comparable to a protocol implementation conformance statement proforma (PICS proforma) (ISO, 1994) can be provided, which allows for definition of security policies with respect to execution of remotely defined BPEL scripts. This checklist is called a security policy statement (SPS) proforma and will be used to indicate allowed features of BPEL in compliance with the security policy. Hence, while CTMF is dedicated to black box test environments, we extend the concepts of CTMF to the situation where code inspection being a specific method in white box testing will be employed in order to analyse the security- relevant semantics of BPEL scripts. In order to discuss our approach a short example presenting some typical information contained in a SPS proforma is shown in Figure 2.

Security Policy Statement		
For domain:		
…		
(additional identification information not of interest in this example)		
Relating to BPEL scripts from domain:		
Invocation of Web services outside current domain allowed?		Y/N
If yes, indicate allowed external Web services:		
URL:	Ref. to EWSRS	
URL:	Ref. to EWSRS	
Indicate restricted Web services in own domain:		
URL:	Ref. to IWSRS	
URL:	Ref. to IWSRS	
Indicate unrestricted Web Services in own domain:		
URL:		
URL:		
… *(statements with respect to other security relevant language features may follow)*		

Figure 2 - Example of Security Policy Statement Proforma

After indication of identification information concerning the domain to which the SPS relates and the domain, which is allowed to provide remotely defined BPEL scripts, there is an indication as to whether invocation of Web services in foreign domains will be allowed or not. If invocation of such Web services is not prohibited in general, there may be indications of particular foreign Web services each identified by its respective URL that are allowed to be invoked in a BPEL script. For each allowed Web service a so-called External Web Service Restriction Statement (EWSRS) may be referenced that contains further information concerning restrictions with respect to the particular Web service. In this example there is only room for up to two such Web services, but it is understood that the proforma may be extended to accommodate any number of list-type elements present in this example.

After indication of external Web services that are allowed, there are two further groups of indications in Figure 2 concerning Web Services of the current domain: one concerning Web Services for which invocation is restricted and one concerning Web Services that may be invoked without any restrictions. While the last type of entry does not require any further information besides the URL of the particular Web service, for each restricted Web service there is a field for indicating a reference to a so-called Internal Web Service Restriction Statement (IWSRS) similar to the indication for external Web services above. Both the group of EWSRS and group of IWSRS referenced in Figure 2 are considered part of the SPS as a whole. The SPS is understood to indicate all of the security-relevant semantics accepted in a BPEL script for cross-domain deployment. Therefore, all other security-relevant semantics not explicitly stated in SPS as being allowed, will be prohibited.

In Figure 3 there is an example of an IWSRS proforma in order to clarify the typical information contained in this part of an SPS. There may be indications concerning restrictions related to particular input parameters of the Web service by indicating the XML-name of the parameter as well as indications concerning restrictions related to allowed processing with respect to particular output parameters, e.g. type of computations performed or flow control dependent on values of output parameters or assignment to outbound messages allowed.

Internal Web Service Restriction Statement	IWSRS-ID	
Restrictions with respect to invocation parameters		
Parameter:		
Restrictions with respect to processing of output parameters		
Parameter:		
Assignment of output values to outbound messages outside own domain allowed?		Y/N
…(statements with respect to other security relevant use of output values may follow)		

Figure 3 - Example of Internal Web Service Restriction Statement Proforma

In our example (Figure 1), if W_2 would be the only instance in domain B where remotely defined BPEL scripts would be allowed, the SPS for domain B with respect to domain A would state that only invocation of Web services within domain B is allowed, and W_1 is the only internal Web service allowed to be invoked. Further restrictions apply with respect to the output parameters of W_1, since only the information I_A intended for external use is allowed to be carried outside of domain B. This would be specified in an IWSRS for W_1 indicating the restrictions with respect to the output parameters that may not be assigned to outbound messages.

While the examples are given in human-readable tabular format it is obvious that it is straightforward to define appropriate XML schemas in order to be capable of presenting the information in machine-processable format.

5. Analysis and Assessment of Security-relevant Semantics of Business Processes

Specifying the restrictions derived from security policies as indicated in this example makes it easier to analyse a BPEL script than it would be without specifying the security policies in such a SPS. The statements in SPS are focused on the security-relevant elements in BPEL. Therefore, during analysis these elements can be searched for in a straightforward manner.

However, this approach involves consequences with respect to the language features that may be employed:

- If only domain internal Web services are allowed then, in order to allow for checking the domain part of a URL prior to execution, only URLs explicitly predefined within the script (at least the domain part of a URL) are allowed in a BPEL script
- If restrictions with respect to particular Web services apply, also the part of the URLs containing the Web service name has to be present explicitly

Obviously, this may limit the expressiveness of BPEL scripts. However, as is the case in Figure 1 and many other applications, the URLs of nodes involved in foreign domains are

known and fixed throughout the runtime of the script. In many cases this also holds for names of Web services invoked. Therefore, these restrictions, though looking very tight at first sight, still leave room for useful applications. Without going into details it is apparent that the more fine grained analysis is implied by the policy, the more likely is the need for human support during analysis. Therefore, in cases where it would be too complicated to differentiate between allowed semantics for cross-domain defined BPEL scripts and semantics not allowed by the way of indications in an SPS of domain B, it would still be possible to extract allowed semantics from the Web service being remotely defined and encapsulate it into another Web service, say W_3, defined within domain B. Thus, it would be possible to define an SPS indicating even more restricted semantics for remotely defined BPEL scripts, since semantics required for the functionality now present in W_3 may also be excluded. By just adding W_3 to the list of allowed Web services for invocation in the SPS for domain B it would again be possible to provide the originally intended overall semantics of a remotely defined BPEL script.

Even with the definition of SPS in the manner described above the task of analysing security-relevant semantics of BPEL scripts and matching against restrictions imposed by policies still is not, in every case, trivial and it might not always be capable of being performed automatically. Therefore, it may be a promising approach to perform this task at a dedicated node within a domain, say node B_S in our example, instead of performing this task at every node in domain B. Human interaction, when required during the analysis of scripts and comparison to security restrictions, may be more easily provided at a single node (or only few nodes, if single point of failure would be an issue) in a domain compared to the situation when being distributed across the domain. It may also facilitate use of specific software required for this purpose when it only needs to be available at a single instance both with respect to potential license fees and effort for user training.

Again, approaches in conformance testing as described in CTMF (ISO, 1994) gave rise to this approach to concentrate assessment in dedicated nodes within a domain. In our example, if domain A permits acceptance of remotely defined BPEL scripts from domain B, it could also be useful to perform examination of the scripts against the appropriate SPS at one particular node in domain A, say node A_S, independent of where the script is to be executed later on. In such cases of mutual exchange of BPEL scripts, the analysis and assessment could further be centralized to a particular node shared by both domains for this purpose. However, the issue of privacy of information contained in security policies as well as the issue of trust implied in this delegation of assessment have to be considered in order to render this approach possible. Going into details on these aspects is beyond the scope of the present paper.

6. Related Work

Sekar *et al.* (2003) propose an approach to the problem of executing untrusted code, in general, i.e. not specific to business process languages, by deriving information as to the behaviour of the code at the level of system calls from execution monitoring or static analysis at the developing site. The information derived is mapped to a model describing security-relevant behaviour and carried together with the code to the executing site where this information can both be checked against security policies and used during execution in order to monitor potential deviations from the stated behaviour. Though this approach appears to be

very close to the one presented in this paper, there are important differences which tend to make this approach more difficult to be applied, at least in the context of business processes, if it will be applicable to this area at all. The requirement of extensive testing as explicitly stated in this paper, and the need for observation at the level of system calls as well as the monitoring during execution, make application of the approach very complex. By contrast, these requirements are not present in our approach as it is based on code inspection techniques applied at script level.

Mendling *et al.* (2004) present an approach to addressing the second aspect in the list in section 2. By extracting RBAC models from BPEL scripts, and converting BPEL code in a format suitable for a particular RBAC software component, they provide an automated link of access control requirements into business processes defined by the BPEL scripts.

Koshutanski and Massacci (2003) also address access control issues of business processes defined by BPEL scripts, in particular the problem of providing the required evidence of possessing the required access privileges at the right time to the right place during execution of a business process. This approach does not address any issues from the list in section 2.

Fischer and Wenzel (2004) extend the scope of services or processes being defined by business process languages to the area of grid computing and grid services. Among others they provide a conversion of scripts written in one business process language, XPLD (Workflow Management Coalition 2002) in this case, into scripts in another business process language, namely BPEL. The scenarios for using scripts written in business process languages in a grid computing environment add another example of the approach to execute remotely-defined business processes, e.g. in BPEL, to the example discussed in the present paper.

7. Conclusions and Future Research

In this paper, we have presented an approach to one of the security aspects arising from executing business processes or enhanced Web services defined outside the domain of execution, by checking the semantics of the BPEL script defining the process via inspection and comparison with the security policies effective at the site of execution. To this extent, we have proposed a method for defining the security policy based on an SPS proforma focusing on the security-relevant semantics of BPEL. This facilitates analysis of security-relevant semantics of business processes and matching it against restrictions imposed by security policies. Since, in contrast to (Sekar *et al.* 2003), there is no need for testing or monitoring during analysis and execution, the application of our approach is straightforward and well-suited for automated execution in the context of business processes in SOC. This is an important feature considering the lightweight nature of fast changing business applications, the high level of abstraction above system calls and platform-independence of business process languages. Further, an infrastructure has been introduced for separating the task of analysis and assessment out of the particular node executing the business process and delegating it to a dedicated node in a domain. In this way, capabilities and resources required for this purpose may be centralised and reused from throughout the domain. Future work will be dedicated to more exhaustive description of the security-relevant aspects of business process languages. Furthermore, the issue of increasing trust in distributed multi-domain

environments will be investigated, in order to make it viable to tap the full potential of outsourcing the analysis and assessment task, even across domain boundaries.

8. References

Aalst, W.M.P. v.d., Dumas, M., ter Hofsted, A.H.M., and Wohed, P. (2002) "Pattern Based Analysis of BPML (and WSCI)" *Technical report FIT-TR-2002-05*, Queensland University of Technology, May 2002, http://sky.fit.qut.edu.au/~dumas/bpml_report.pdf, last accessed 2004-12-28

Abendroth, J. and Jensen, C.D. (2003) "Partial Outsourcing: A New Paradigm for Access Control" *SACMAT'03*, June 2003, Como, Italy

Arkin, A. (2002) "Business Process Modeling Language", *BPMI.org*, November 2002, http://www.bpmi.org/bpml-spec.htm, last accessed 2005-02-25

Arkin, A., Askary, S., Fordin, S., Jekeli, W., Kawaguchi, K., Orchard, D., Pogliani, S., Riemer, K., Struble, S., Takacsi-Nagy, P., Trickovic, I., and Zimek, S. (2002) "Web Service Choreography Interface (WSCI) 1.0", *W3C*, 8 August 2002, http://www. w3.org/TR/2002/NOTE-wsci-20020808, last accessed 2004-12-28

Arkin, A., Bloch, B., Curbera, F., Goland, Y., Kartha, N., Liu, C.K., Thatte, S., and Yendluri, P. (2004) "Web Services Business Process Execution Language Version 2.0", *OASIS*, December 2004, http://www.oasis-open.org/committees/download.php/10347/wsbpel-specification-draft-120204.htm, last accessed 2004-12-28

Berardi, D., De Rosa, F., De Santis, L., and Mecella, M. (2003) "Finite State Automata as Conceptual Model for E-Services", *Integrated Design and Process Technology, IDPT-2003*, June 2003

Berglund, A., Boag, S., Chamberlin, D., Fernández, M.F., Kay, M., Robie, J., and Siméon, J. (2004) "XML Path Language (XPath) 2.0", *W3C*, 29 October 2004, http://www.w3.org/TR/ xpath20, last accessed 2004-12-28

Cousot, P. (1999) "Directions for research in approximate system analysis", *ACM Computing Surveys (CSUR)*, Volume 31, Issue 3es , September 1999

Deubler, M., Grünbauer, J., Jürjens, J., and Wimmel, G. (2004) "Sound Development of Secure Service-based Systems", *ICSOC'04*, November 2004, New York, USA

Dimmock, N., Belokosztolszki, A., Eyers, D., Bacon, J., and Moody, K. (2004) "Using Trust and Risk in Role-Based Access Control Policies", *SACMAT'04*, June 2004, Yorktown Heights, New York, USA

Fischer, O. and Wenzel, B. (2004) "Prozessorientierte Dienstleistungsunterstützung: Workflowbasierte Komposition unternehmensübergreifender Geschäftsprozesse", University of Hamburg, http://vsis-www.informatik.uni-hamburg.de/getDoc.php/thesis/177/DA-Wenzel-Fischer-final.pdf, last accessed 2004-12-28

ISO (1994) "ISO/IEC 9646-5:1994 Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 5: Requirements on test laboratories and clients for the conformance assessment process", *International Organisation for Standardization*, Geneve, 1994

Koshutanski, H. and Massacci, F.(2003) "An Access Control Framework for Business Processes for Web Services", *ACM Workshop on XML Security*. October 31, 2003, Fairfax VA, USA

Malu, P., Dubray, J.J., Lonjon, A., Buchinski, E., Chan, A., Mukkamala, H., and Smiley, D. (2002) "ebXML Business Process Specification Schema, Version 1.05", *UN/CEFACT and OASIS*,. http://xml.coverpages.org/ebBPSSv105-Draft.pdf, last accessed 2005-02-25

Medjahed, B., Benatallah, B., Bouguettayaet, A., Ngu, A.H.H., and Elmagarmid, A.K. (2003) "Business-to-business interactions: issues and enabling technologies", *VLDB Journal* (2003) 12: 59-85, April 2003

Mendling, J., Strembeck, M., Stermsek, G., and Neumann, G. (2004) “An Approach to Extract RBAC Models from BPEL4WS Processes”, *Proceedings of the Thirteenth IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WET ICE 2004)*

Papazoglou, M.P. and Georgakopoulos, D. (2003) “Service-Oriented Computing”, *Communications of ACM*, 46(10):25–28, October 2003

Sekar, R., Venkatakrishnan, V.N., Basu, S., Bhatkar, S., and DuVarney, D.C. (2003) “Model-Carrying Code: A Practical Approach for Safe Execution of Untrusted Applications”, *SOSP'03*, October 2003, Bolton Landing, New York, USA.

Shapiro, R. (2002) “A Comparison of XPDL, BPML, and BPEL4WS”. http://xml.coverpages.org/Shapiro-XPDL.pdf, last accessed 2004-12-28

Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maquire, T., Sandholm, T., Snelling, D., and Vanderbilt, P. (2003) “Open Grid Services Infrastructure (OGSI) Version 1.0”, *Global Grid Forum, GGF*, June 2003, http://www.ggf.org/documents/GWD-R/GFD-R.015.pdf, last accessed: 2004-12-28

Wohed, P., van der Aalst, W., Dumas, M., and ter Hofstede, A. (2002). “Pattern-Based Analysis of BPEL4WS”, *Technical report, FIT-TR-2002-04*, Queensland University of Technology, Brisbane, 2002, http://tmitwww.tm.tue.nl/research/patterns/download/qut_bpel_rep.pdf, last accessed 2004-12-28

Workflow Management Coalition (2002) “Workflow Process Definition Interface - XML Process Definition Language, Version 1.0”, October 2002, http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf, last accessed 2004-12-28