

Modelling of Service Compositions: Relations to Business Process and Workflow Modelling

Michael C. Jaeger

Technische Universität Berlin, FG Formal Models, Logic and Programming
Sek. FR 6-10, Franklinstrasse 28/29, D-10587 Berlin, Germany
mcj@cs.tu-berlin.de

Abstract. The service oriented architecture (SOA) represents a trend in the IT industry for the development of a flexible and unifying software infrastructure. In an SOA, software components provide their functionality as a service by using uniform interface description and invocation protocols. The provision of software components in an uniform manner allow their efficient composition to form new complex services. Currently, the compositions of services is a popular field of research with many ongoing efforts.

However, the sheer number of existing proposals and efforts to describe service compositions in this field have led to term Web Services Acronym Hell (WSAH) [1] and an obvious confusion. This paper intends to serve as an orientation for explaining what the differences between business processes and workflow control flow languages are and why service compositions are used in this field. It will also introduce past and existing proposals for Web service composition languages for understanding why so many different languages for modelling workflows, business processes and compositions exist.

1 Introduction

Services in an SOA use standardised interface descriptions and uniform invocation protocols. The Web services proposal by the W3C has defined such elements of an SOA by using Internet protocols for the invocation and XML-based description formats for the description and messages in an SOA [2]. A service composition results from combining existing services to realise a new, more complex functionality. The SOA reflects the task-orientation of modern businesses that form a business process. Using services in this setup leads to services that fulfil individual tasks of a process. Arranging them together to a composition provides the implementation of a process. One candidate for describing and defining compositions of Web services is the Business Process Execution Language for Web Services (“BPEL4WS”). In the remainder of this paper, a language like BPEL4WS is named composition language. However, in this field some problems of understanding arise:

- The BPEL4WS proposal carries “Business Process Execution” in its name. Is a service composition equivalent with the realisation of a business process? What is the relation between business processes and compositions?

- Considering the definition of the Workflow Management Coalition (WfMC) provided in their comprehensive workflow reference model saying that “a workflow is the computerised facilitation or automation of a business process, in whole or part” [3]: Does a service composition represent a workflow when realising a business process? Based on this definition it seems odd why there is a “business process execution language” used for service compositions, but not an analogous “workflow execution language” which would be the more consistent name with respect to the definitions of the WfMC.
- Why there are so many different composition languages? What are their origins and in which direction will they evolve in the future?

This work intends to give a clarification and to answer the posed questions in the following way: In Sections 2 and 3, a rough overview about business processes, workflows and their modelling languages is given. In Sections 4 and 5, differences between both are discussed and the main considerations why service compositions match these two fields are introduced. And then, based on the given clarifications, composition languages are discussed which are found most in today’s literature in Section 5.1. After these main three parts, the paper ends with the conclusions in Section 6.

2 Business Processes

In the mid-90s, the work of Hammer and Champy, who introduced business process reengineering as a main strategy to improve existing businesses and organisations [4], brought the IT industry to put more efforts in the development of software systems that facilitate the creation and management of business processes. The basic idea of this strategy is to design a process in the most modern and optimised way leading in a process definition without legacy artefacts. Every business process should have a dedicated customer and thus offer a clear benefit. In addition, a process should also have a defined process owner, who is in charge and responsible in order to provide customers with a defined point of contact.

At that time, different organisational units were divided by their functional responsibilities, and a process typically crossed different units. As a result, processes performed inefficiently and in case of problems, a responsible party was hard to identify. To overcome these lacks of efficiency, the business process reengineering was proposed as an optimisation effort of existing activities. As a side-effect of standardised processes, monitoring and benchmarking tasks produce results that are more suitable for comparison. And based on the defined processes, mistaken activities or misunderstandings between involved actors are reduced and thus the productivity is improved.

Apart from the evident advantages, analysis from the performed business process reengineering efforts have also revealed a couple of disadvantages. Most noticeable are that too radical changes will lead to social problems in an organisation [5, p. 239]. Moreover, a strong focus on the process optimisation also runs the risk of poor improvements on what the individual activities produce.

2.1 Modelling Business Processes

One motivation of modelling business processes is to achieve a clear and common understanding between the management and the involved actors and users of the process. For modelling processes, different graphical and textual languages exist, which can be used to create diagrams or a description of a business process that is ready for interpretation by software systems. Graphical representations can be flow diagrams, block diagrams, or basic graphs. Considering a graph, a node represents usually an activity, an event, or an entity where directed edges represent the (causal) relations between the elements.

One early introduced graphical language for the modelling of processes is the event-driven process chain (EPC) [6]. As its name suggests, the basic element of an EPC is the event, which is a defined condition and thus can be the result of a process, a function, or an external event. Contrary to events, a function is an active element which changes a state and thus is the object between different events. Events and functions, can be combined with combination operators, like ANDs or XORs. EPCs are suitable for the modelling of control flows that define the order of occurring events and executed functions. To model the data flow of a process (or also the flow of goods) extensions are proposed that appear in literature as so-called extended EPCs [5, p. 221].

The Business Process Management Initiative (BPMI) has introduced the Business Process Modeling Language (BPML) as a textual language for describing business processes [7]. The BPMI represents a non-profit organisation with the goal to support and coordinate the activities in the field of business processes among its members. The BPML is intended to serve as a comprehensive description of a business process, which also covers aspects of implementing business processes. It consists of different elements to describe the control flow as well as the data flow of a process. The standard textual representation used for BPML documents is XML. The BPMI has also released a graphical notation called Business Process Modelling Notation (BPMN) to provide a set of symbols and layout conventions for drawing business process models [8]. In the recent past, the BPMI has merged their efforts with the business process modelling activities of the Object Management Group (OMG), which represents a non-profit organisation for pursuing interests in the area of object-oriented software technologies.. As a result from this merger, the work focusses on new releases of the BPMN adopted by the OMG.

The Business Process Specification Schema (BPSS) has a slightly different focus than the BPML. The BPSS is part of the ebXML suite which supports establishing agreements to facilitate electronic businesses on an inter-organisational level. The motivation background for this effort is to provide a specification for helping developing countries to participate in electronic commerce without being dependent on technologies offered by particular vendors. The ebXML is driven by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and a Technical Committee at Organisation for the Advancement of Structured Information Standards (OASIS). The first release that can be traced back was submitted in 2001. At that time, compositions of services in an SOA were not mentioned at all.

The Business Process Execution Language (BPEL), also named BPEL for Web Services (BPEL4WS) represents another proposal for specifying business processes. This proposal covers the modelling of processes and the description about the involvement of concrete Web services. At the moment, a committee at the OASIS coordinates the development of BPEL. Before, BPEL was carried out by a joint venture of the software industry. Originally, it resulted from a merger of the Web Services Flow Language by IBM (WSFL, [9]), which shows influences from IBM's MQ Series workflow software [10], and XLANG [11], which was intended to serve as the process modelling language in Microsoft's BizTalk software.

Besides the mentioned EPC, BPMN, BPEL, WSFL, and XLANG, other approaches to model business processes exist. The given selection represent the commonly mentioned proposals in papers discussing the area of service compositions. In addition, the WSFL, XLANG and BPEL4WS proposals directly cover the realisation of processes with services by forming service compositions, which also indicate that the border between business processes and service compositions becomes blurred. Another popular proposal for modelling business processes is the Unified Modelling Language (UML) from the OMG, for example by using activity diagrams [12, cf. section 2.13.2.1]. Originally intended for "software-intensive" systems as the foreword of the UML specification explains, this modelling language is already implemented by different software products for the development of service compositions.¹

3 Workflow Management

The management of workflows has got a different origin from the management of business processes. Workflow management systems (WFMS) were first applied for specific application cases. One of the systems mentioned as the first steps is the so called OfficeTalk which came as a part of the Xerox Star computer system [15]. The Xerox Palo Alto Research Centre (PARC) has developed this system in the 70s to support the work with documents in a typical office environment. This did not represent typical WFMS of today. However, OfficeTalk reflects the way humans would work on documents without computers – a non-computer-supported workflow. Clearly, workflow has in this sense a strong relation to support collaboration and document management. By covering this application scenario, the workflow management systems brought a couple of benefits. The most important advantage is the automated execution of tasks, which do not require an interactive handling, in order to accelerate the workflow. By the automated processing of data, the application of a WFMS anticipated that documents are kept in the software systems by bundling different systems together. Cumbersome re-entry of data which is time-costly should be avoided. And, when workflow management systems coordinate the work, also data can be derived that indicates the performance of the performed processes. Clearly,

¹ Examples are the Oracle BPEL Process Manager [13], or the WebSphere Integration Developer by IBM [14].

this represents an analogy to the main benefits identified for the business process reengineering.

From this early beginning, WFMS developed further to support other application domains than the work around documents in an office. Also, WFMS became more compatible to coordinate tasks performed by external systems. As an example, Mahling et al. explain the evolution of the Poise system [16]: in its first version, it represented an office information system supporting the handling of documents, such as entering information or realising static workflows. In the mid-80s, a subsequent development called Polymer offered a more sophisticated concept of modelling workflows that resulted in more flexible workflows and in a better coverage of different application scenarios. Based on that, Polyflow was introduced in 1995 as a full-featured WFMS in the sense of the reference model by WfMC [3].

3.1 Modelling Control Flows in Workflows

Since the development of WFMS begun, vendors have provided proprietary modelling languages. Van der Aalst et al. have introduced a set of patterns that describe structural characteristics and functional capabilities of WFMS and involved languages. Their analysis covers about 15 different WFMS [17].

Besides the reference model, the WfMC has introduced the XML Process Definition Language (XPDL, [18]). XPDL was intended to serve as a platform independent description language that allows to share workflow descriptions between tools and execution environments of different vendors. As a consequence, the authors have chosen the XML as its textual notation because of its wide support among different platforms. Besides the efforts from the industry side, research work covers also the modelling and specification aspect. Most approaches consider the application of EPCs, process calculi or (high-level) Petri nets as a foundation. Petri nets, also called Place-/Transition-Nets (P/T-Nets), are named by its "inventor" Carl Petri and serve as a formal language for modelling and specifying discrete events of dynamic systems. They show a high degree of versatility and are also applied to the specification of telecommunication protocols, or used as description of logistic chains. Petri nets were introduced by Petri in the year 1962; the ISO covers Petri nets today as a proposed standard [19].

Janssens et al. have introduced an analysis of existing workflow modelling efforts that use Petri nets [20]. Their analysis covers 12 contributions. The main reasons for using Petri nets for modelling workflows are that they offer a graphical notation and a formally defined semantic of its elements which allow the application of formally proven analysis techniques [20,21]. Among the contributions in this field, van der Aalst et al. have defined a Petri net variant which they call *Workflow Nets* for modelling workflows [22]. Based on this work and the patterns analysis, van der Aalst et al. have developed "Yet Another Workflow Language" (YAWL, [23]). This language extends the concepts of Workflow nets to support the workflow patterns while keeping a formal foundation that allows the anticipated verification of workflow models.

4 The Workflow Versus Business Processes Discussion

The previous two sections have introduced two fields that seem to have many issues in common. The resulting question is what makes these two different? The two terms business process and workflow modelling are used synonymously sometimes, suggesting that using either the one or the other refers to historic reasons. The two main communities who represent the workflow corner and the business process community, i.e. the WfMC and the BPMI, have discussed this questions in more detail. In a retro-perspective of the workflow reference model published by the WfMC [24], the authors acknowledge the growing momentum of business process management efforts, which have developed in parallel to the workflow-related efforts. The WfMC explains that a difference results from the evolution of the technologies in the workflow area. However, WFMS meet today the requirements of business process management as well. Consequently, their proposal sees the original reference model as the foundation for a new reference model that covers also business process management.

Members from the BPMI promote a different view. Smith and Fingar have initiated a discussion by publishing that a workflow represents just a process expressed using a process calculus [25] and thus is purely concerned with process description. According to their view, this clear focus makes workflows and WFMS less suitable for the new requirements of today's businesses. They explain that workflows are rather static (i.e., application dependent) and do not support changes of the process. Their statement has received a response from the workflow community which has motivated the two authors to publish a clarification [26]. What remains is that according to the viewpoint of the BPMI, a workflow represents just one aspect among the different fields subsumed by the business process management: in addition to workflows, business process management systems cover the integration of different computer systems as well as the non-computerised parts of business processes.

When it comes to the application of the terminology and the referring languages such as for looking at composition languages, a clarification must be achieved to distinguish what is different and what is claimed to be different. Apparently, the difference between both results from their different origins. The underlying problems such as the expressiveness of the modelling languages, how verifications can be applied, or which graphical modelling language is the most efficient, appear for both directions quite similar.

5 Realising Business Processes and Workflows with Service Compositions

The similarities between a process or workflow model and the composition of services in an SOA are evident. Modelling business processes usually involves a description about required activities and a specification about their execution order; this also represents what is basically required for describing a service composition. Also, business processes should have a defined input and output while

the implementation of each task becomes secondary, what poses a similarity to the interface-orientation of services [27,28]. Moreover, an SOA provides different architectural characteristics which are promoted as the requirements on the IT-infrastructure in today's businesses [26]:

- **Technology Independence.** [29,30,31] The basic motivation to promote the SOA in companies is to establish a middleware that ties together functionality offered by different systems regardless of their hard- and software. This resembles the *access transparency* defined by the ISO RM-ODP [32, p. 17]. The heterogeneity of IT systems in companies is the standard in large businesses, because different departments of an enterprise have usually started the application of computer systems in most cases independently from each other.
- **Location Transparency.** [27,30,32] By definition, services offer their interface over a network. Web services uses Internet protocols for descriptions and messaging. Service consumers can use services across the local network or the boundaries of the local organisation. Although an organisation might not want to invoke any service that is just available anywhere in the world, the use of Internet protocols offers a greater level of flexibility.
- **Loose Coupling.** [29,30] The term loose coupling in an SOA is similar to the understanding of dynamic typing in the field of programming languages. Usually, in an SOA, loose coupling means that a service consumer knows what kind of service is required at design time. However, the binding ("coupling") to a real available service takes place at the run-time. Loose coupling enables service consumers to revise existing bindings during the run-time when necessary.

5.1 Modelling Service Compositions

Among the proposals for modelling service compositions, most languages provide direct support for Web services as this represents the major SOA implementation used today. The current situation shows different composition languages that share the same goal what resembles the situation with workflow or business process modelling. Also, some business process modelling languages are considered to create service compositions. To provide a better orientation among these, the following three groups are proposed:

- **Abstract level languages, L1.** These languages are primarily intended to describe a composition with activities in the way that services might realise these activities. However, concrete services are not mandatory to make the description complete. When not mentioning available particular services, such a description handles the activities or involved services as black boxes.
- **Concrete level languages, L3.** On the concrete level, the description covers the invocation of concrete services from a technical view. Usually, a service does not provide an atomic, stateless operation, but also provides different operations and requires to manage a session or a state. A description

must cover states when interoperating with services as well as considering particular operations. These languages do not focus entirely on service composition, because they are also concerned with the interoperation of different parties rather than realising a process.

- **Languages covering both levels, L2.** Some languages are right in the middle of both, not showing a clear process modelling focus and also not a clear interoperation focus. An example for such a language represents BPEL which allows the description of a business process among with an executable description involving available Web services. Compared with L3-languages, L2 would represent typical service composition languages.

The reader should note that this categorisation does not provide a formal basis nor an argumentation like if concept x is found in a language z , then it belongs to level y . This categorisation has the purpose to give an orientation. It resembles the idea of the model-driven architecture (MDA, [38]) which is a proposal for the model-centric development of software systems. This proposal distinguishes

Table 1. Composition Languages Overview

| L | Acronym | Full Name, <i>Reference</i> Supporting Parties, Remarks |
|---|---------|---|
| 1 | XPDL | XML Process Definition Language [33] WfMC, contributing authors were from Global 360, FileNet, Staffware/TIBCO, Prozone and Fujitsu Software |
| 1 | BPML | Business Process Modelling Language [7] BPMI, mentions only one contributing author from Intalio |
| 1 | BPSS | Business Process Specification Schema [7] (<i>Part of the ebXML Suite</i>) UN/CEFACT, an United Nations Body for Electronic Trade and an OASIS TC, including members from Cyclone Commerce, Fujitsu, SAP AG and Sun Microsystems |
| 2 | WSFL | Web Services Flow Language [9] IBM, moved into the BPEL4WS proposal |
| 2 | XLANG | subtitled "Web Services for Business Process Design" [11] Microsoft, moved into the BPEL4WS proposal |
| 2 | BPEL4WS | Business Process Execution Language for Web Services [34] IBM, Microsoft and BEA, moved into the WS-BPEL proposal |
| 2 | WS-BPEL | Web Services Business Process Execution Language [35] An OASIS TC involving about 18 industry parties, among them BEA Systems, IBM, Microsoft, Oracle, Sun Microsystems, SAP AG |
| 3 | WSCI | Web Service Choreography Interface [36] W3C Note submitted by BEA, Intalio, SAP AG und Sun Microsystems |
| 3 | WS-C | Web Service Choreography [37] W3C Working Group, continuing the WSCI proposal |

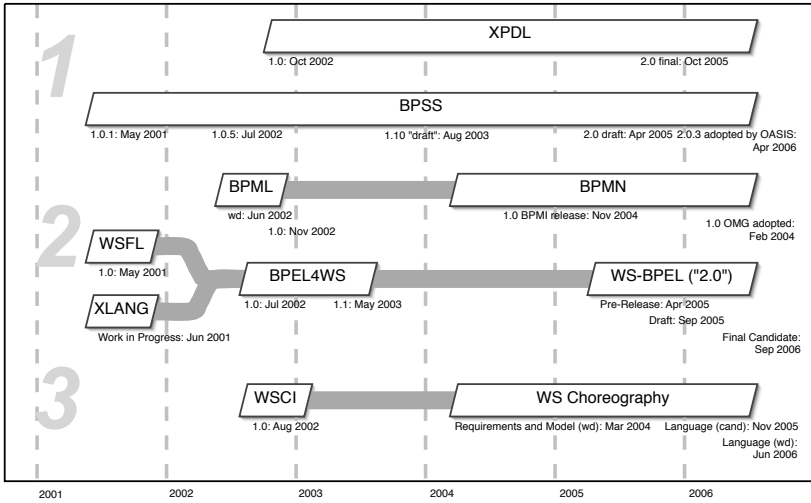


Fig. 1. Release Dates of Process Description and Service Composition Languages

(software) models at different stages of the development process. The two main categories of models are platform-independent models (PIM) and platform-specific models (PSM). PIMs do not contain modelling artefacts resulting from specific technologies, platforms or standards; they represent an *abstraction* from technical details. Based on these, PSMs are derived from PIMs to provide the technical information necessary for the implementation of the software.

Table 5.1 lists a selection of composition languages with their acronyms and their proposed categorisation. In addition, Figure 1 shows a chronological overview about their introduction dates. The two BPEL proposals, WSFL, and XLANG were mentioned already in Section 2.1. These candidates can be clearly called Web service composition languages and fit into the second group.

Regarding the languages at the first level, Table 5.1 mentions XPD, BPML, and BPSS. These languages provide elements to support the invocation of Web services; however, their involvement is not mandatory. The BPSS is completely independent from particular SOA technologies. To provide the ties to concrete SOA implementations like Web services, separate specifications called a Collaboration Protocol Profile (CPP) and a Collaboration Protocol Agreement (CPA) are proposed. A BPSS document can refer to a CPP or CPA to provide a description of the involved services, for example, by referencing to a WSDL description. However, since the BPSS has a clear focus on enabling electronic trade, it is thus not the preferred candidate for implementing business processes with service compositions.

The Web Service Choreography Interface (WSCI) represents the foundation for the third group. The WSCI proposal directly focusses on specifying the required message flow between Web service requesters and providers resulting from their offered operations [36]. Consequently, the authors of BPML mention

explicitly that WSCI and WSDL specifications are complementary to BPML [7, section 1.2]. The WSCI has been submitted to the W3C as a technical note in 2002. In beginning of 2003, a W3C working group named Web Services Choreography has begun its work to take the WSCI proposal as input to elaborate this aspect further [37]. By choreography, the working group refers to the *characteristic of describing linkages and usage patterns between Web services*. The working group uses the term choreography in a similar manner as other proposals use collaboration, conversation, coordination or orchestration.

When comparing the different languages and specifications, not only their focus but also their expressiveness is a useful criteria. Such a comparison has already been conducted by Wohed, van der Aalst et al. They have presented a comprehensive analysis about the structural capabilities of languages to specify Web service compositions for BPEL4WS, XLANG, and WSFL [39,1], BPML, and WSCI [40], and XPDL [10]. Their result is an analysis about which language directly supports which pattern of their workflow patterns. Apart from these languages, other proposals exist which were in previous sections. Considering graphical notations, EPCs and the BPMN are suitable for the description of service compositions as well. However, none of these efforts shows a noticeable impact in this area. For this purpose the UML plays a more important role, as research work has already indicated [41,42].

6 Conclusions

This paper has proposed a classification of composition languages and languages for modelling business processes. The classification helps to identify different players in the Web Services Acronym Hell (WSAH) [1]. The classification is roughly based on separating composition languages for abstract process design from languages for defining how to involve concrete Web services. Moreover, it was discussed that composition languages have much in common with the modelling of workflows. By using their workflow patterns, Wohed, van der Aalst et al. give a detailed explanation about the capabilities of composition languages with their problems and – more important – they also explain how previously introduced workflow modelling languages have coped with that.

The overview has identified different parallel efforts working on composition languages, which is still an ongoing process. Apparently, the OASIS-driven WS-BPEL has the biggest momentum in the service composition area. It gathers together main industry players, such as IBM, Microsoft, Oracle, SAP, and Sun Microsystems. Besides, the WS Choreography working group shows also ongoing activity for the definition of service compositions on a technical level.

References

1. van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M.: Web Service Composition Languages: Old Wine in New Bottles? In: EUROMICRO'03. Proceedings of the 29th EUROMICRO Conference New Waves in System Architecture, Belek, Turkey, September 2003, pp. 298–304. IEEE Computer Society Press, Los Alamitos (2003)

2. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web Services Architecture (February 2004), <http://www.w3c.org/TR/ws-arch/>
3. Hollingsworth, D.: The Workflow Reference Model. Technical Report TC00-1003, Workflow Management Coalition, Lighthouse Point, Florida, USA (1995)
4. Hammer, M., Champy, J.: A Manifesto for Business Revolution. Harper Business (1993)
5. Frank, H., Gronau, N., Krallman, H.: Systemanalyse im Unternehmen, 3 edn. Oldenbourg Verlag, München, Germany (October 2000)
6. Keller, G., Nüttgens, M., Scheer, A.W.: Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK). Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) 89, Universität des Saarlandes, Saarbrücken, Germany (1992)
7. Assaf, A., et al.: Business Process Modeling Language (BPML). Technical Report Version 1.0, BPMI.org (2002)
8. White, S.A.: Business Process Modeling Notation (BPMN). Technical Report Working Draft (1.0), BPMI.org (August 2003)
9. Leymann, F.: Web Services Flow Language (WSFL 1.0). Technical report, IBM Software Group (2001), <http://www-4.ibm.com/software/solutions/webser-vices/pdf/WSFL.pdf>
10. van der Aalst, W.M.P.: Don't go with the flow: Web services composition standards exposed. In: Jan/Feb 2003 Issue of IEEE Intelligent Systems, January 2003, pp. 72–76. IEEE Computer Society Press, Los Alamitos (2003)
11. Thatte, S.: XLANG - Web Services for Business Process Design (2001), <http://www.gotdotnet.com/team/xml.wsspecs/xlang-c/default.htm>
12. Object Management Group (OMG): Unified Modeling Language: Superstructure. OMG formal document/05-07-04 (August 2005)
13. Shaffer, D., Dayton, B.: Orchestrating Web Services: The Case for a BPEL Server. Technical report, Oracle Corporation, Redwood Shores, California, USA (June 2004)
14. Lynch, E., Venkatapathy, C.: Sustaining your Advantage with Business Process Integration based on Service Oriented Architecture. White Paper (October 2005)
15. Johnson, J., Roberts, T.L., Verplank, W., Smith, D.C., Irby, C., Beard, M., Mackey, K.: The Xerox Star: A Retrospective. *Computer* 22(9), 11–26 (1989)
16. Mahling, D.E., Craven, N., Croft, W.B.: From Office Automation to Intelligent Workflow Systems. *IEEE Intelligent Systems* 10(3), 41–47 (1995)
17. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. *Distributed and Parallel Databases* 14(1), 5–51 (2003)
18. Marin, M., Brunt, J., Zurek, W., Stephenson, T., Bojanic, S., Gouri, G.: Workflow Process Definition Interface – XML Process Definition Language, Version 1.0. Technical Report WFMC-TC-1025, Workflow Management Coalition, Lighthouse Point, Florida (October 2002)
19. ISO/IEC: ISO/IEC 15909-1: High-level Petri nets – Part 1: Concepts, Definitions and Graphical Notation. Published Standard (December 2004)
20. Janssens, G.K., Verelst, J., Weyn, B.: Techniques for modelling workflows and their support of reuse. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) *Business Process Management*. LNCS, vol. 1806, pp. 1–15. Springer, Heidelberg (2000)

21. van der Aalst, W.M.P.: Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 161–183. Springer, Heidelberg (2000)
22. van der Aalst, W.M.P., van Hee, K.M., Houben, G.J.: Modelling Workflow Management Systems with high-level Petri Nets. In: De Michelis, G., Ellis, C., Memmi, G. (eds.) Proceedings of the second Workshop on Computer-Supported Cooperative Work, Petri nets and related formalisms, pp. 31–50 (1994)
23. van der Aalst, W.M.P., Aldred, L., Dumas, M., ter Hofstede, A.H.M.: Design and implementation of the YAWL system. Technical Report FIT-TR-2003-07, Centre for IT Innovation, QUT (2004), <http://www.tm.tue.nl/it/research/patterns>
24. Hollingsworth, D.: The Workflow Reference Model 10 Years On (extracted from Workflow Handbook 2004). In: Workflow Management Coalition, Lighthouse Point, Florida (February 2004)
25. Smith, H., Fingar, P.: Workflow is just a Pi Process, January 2004. Business Process Trends, Columns and Articles (2004)
26. Smith, H., Fingar, P.: Business Process Fusion Is Inevitable. Business Process Trends, Columns and Articles (March 2004)
27. Bolcer, G.A., Kaiser, G.: SWAP: Leveraging the Web to Manage Workflow. In: IEEE Internet Computing, January-February 1999, pp. 85–88. IEEE Computer Society Press, Los Alamitos (1999)
28. Dijkman, R.M., Dumas, M.: Service-Oriented Design: A Multi-Viewpoint Approach. International Journal of Cooperative Information Systems (IJCIS) 13(4), 337–368 (2004)
29. Huhns, M.N., Singh, M.P.: Service-oriented computing: Key concepts and principles. In: IEEE Internet Computing, January and February 2005, pp. 75–81 (2005)
30. Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. In: WISE'03. Proceedings of the Fourth International Conference on Web Information Systems Engineering, Roma, Italy, December 2003, pp. 3–12. IEEE Computer Society Press, Los Alamitos (2003)
31. Yang, J.: Web Service Componentization. Communications of the ACM 46(10) (2003)
32. ISO/IEC: ITU.TS Recommendation X.902 — ISO/IEC 10746-1: Open Distributed Processing Reference Model - Part 1: Overview (August 1996)
33. Shapiro, R., Marin, M., Brunt, J., Zurek, W., Stephenson, T., Bojanic, S., Gouri, G.: Process Definition Interface – XML Process Definition Language, Version 2.0. Technical Report WFMC-TC-1025, Workflow Management Coalition, Lighthouse Point, Florida (October 2005)
34. Tony, A., et al.: Business Process Execution Language for Web Services Version 1.1. Technical report, BEA Systems, IBM Corp., Microsoft Corp., (2003), <http://www-106.ibm.com/developerworks/webser-vices/library/ws-bpel/>
35. TC, O.W.B.: WS-BPEL Specification Editors Draft (December 2005), http://www.oasis-open.org/committees/download.php/127_91/wsbpel-specification-draft-May-20-2005.html
36. Assaf, A., et al.: Web Service Choreography Interface (WSCI) 1.0. Technical report, W3C (2002), <http://www.w3.org/TR/wsci>
37. Burdett, D., Nickolas, K. (eds.): WS Choreography Model Overview, W3C Working Draft 24 March 2004. Technical report, W3C (2004), <http://www.w3.org/TR/ws-chor-model/>
38. Object Management Group (OMG): Model Driven Architecture. ormsc/2001-07-01 (August 2001)

39. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.: Pattern Based Analysis of BPEL4WS. Technical Report FIT-TR-2002-04, QUT, Queensland University of Technology, Queensland, Australia (2002)
40. van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Wohed, P.: Pattern Based Analysis of BPML (and WSCI). FIT Technical Report FIT-TR-2002-05, Queensland University of Technology, Brisbane, Australia (2002)
41. Skogan, D., Grønmo, R., Solheim, I.: Web Service Composition in UML. In: EDOC'04. Proceedings of the 8th IEEE Intl Enterprise Distributed Object Computing Conf., Monterey, California, September 2004, pp. 47–57. IEEE Computer Society Press, Los Alamitos (2004)
42. Grønmo, R., Jaeger, M.C.: Model-Driven Methodology for Building QoS-Optimised Web Service Compositions. In: Kutvonen, L., Alonistioti, N. (eds.) DAIS 2005. LNCS, vol. 3543, pp. 68–82. Springer, Heidelberg (2005)