Extending the UN/CEFACT Modeling Methodology and Core Components for Intra-organizational Service Orchestration

Philipp Offermann[1], Christian Schröpfer[1],
and Maximilian Ahrens[2]

[1] Faculty of Electrical Engineering and Computer Sciences,
Technische Universität Berlin, Germany
{Philipp.Offermann, Christian.Schroepfer}@sysedv.tu-berlin.de
[2] Deutsche Telekom Laboratories, Berlin, Germany
maximilian.ahrens@telekom.de

Abstract. When creating a company's IT structure based on a service-oriented architecture (SOA), it is necessary to first analyze the business domains and process areas of the company, then to model the business processes to be supported by the SOA and finally to convert the models into a service orchestration description. Currently, few methodologies exist to support this. At our department, we have proven that the UN/CEFACT Modeling Methodology (UMM) can be used for intra-organizational process integration. In this article we analyze if the UMM is sufficient for SOA, which artifacts are missing and how the UMM could be extended. The UMM was created to model the collaboration between different legal entities to perform collaborative business processes. There exist methods to convert these models into executable service choreography descriptions expressed in the Business Process Specification Schema (BPSS) or the Business Process Execution Language (BPEL). However, the business process models can also be used as a basis for an intra-organizational service orchestration. By extending the UMM it is possible to enable the automated generation of service orchestrations using Core Components and the Business Process Modeling Notation (BPMN).

Keywords: UN/CEFACT Modeling Methodology (UMM), Core Components (CC), Business Process Execution Language (BPEL), Business Process Modeling Notation (BPMN), Service Orchestration, Service-Oriented Architecture (SOA).

1 Introduction

Service-oriented architecture (SOA) is a new design paradigm for software systems and IT governance. While the technical concepts behind the SOA are well established, comprehensive methodologies are still rare. At our department we have proven, using action research, that the UN/CEFACT Modeling Methodology (UMM) can be used for intra-organizational process integration.[1] In this article we analyze if the UMM methodology is sufficient for SOA and how it could be extended.

D. Georgakopoulos et al. (Eds.): ICSOC 2006 Ws, LNCS 4652, pp. 154–165, 2007.

1.1 Service-Oriented Architecture

Service-oriented architectures are a current trend in the IT industry. Many big companies like IBM, Microsoft, BEA and SAP are supporting and developing standards for or are converting their products to an SOA.[2-5] Additionally, organizations like the World Wide Web Consortium (W3C), OASIS and the Object Management Group (OMG) are developing and publishing standards related to SOAs.[6-8]

The Gartner Group defines an SOA as follows: "SOA is a software architecture that builds a topology of interfaces, interface implementations and interface calls. SOA is a relationship of services and service consumers, both software modules large enough to represent a complete business function. So, SOA is about reuse, encapsulation, interfaces, and ultimately, agility."[9]

The most common implementation environment for an SOA is Web services. A Web service is a program that offers its functionality through a defined interface over open protocols.[6] The difference to classical modularization of program logic is that the functionality encapsulated by Web services is derived from the business functions composing the business processes and not from the IT systems.[10-12]

Therefore, the concept of an SOA is not restricted to the technical side, but also reaches out to business process management (BPM).[13] By designing services in a way that they represent business functionality, it is possible to align the structure of business processes with the organization of the IT infrastructure supporting these processes.

1.2 Service Orchestration

To be able to align the IT structure with the business processes, it is necessary to describe the sequence in which Web services are called and to make sure that this sequence is structurally equivalent to the sequence of business functions described in the business process model.

Different description languages, called web service orchestration languages, have evolved to describe the sequence of Web services and make this sequence executable. Service orchestration languages are used to support intra-organizational processes through a central coordination, while service choreography languages support inter-organizational processes without a central coordination.[14] Out of the service orchestration languages, the Business Process Execution Language (BPEL) has most support from industry and research.[15]

For representing the structure of business processes, different modeling notations exist, e.g. Event-Process Chain (EPC), ICAM Definition Language (IDEF) and UML Activity Diagrams. There are ways for converting these process models into executable process description languages, e.g. UML to BPEL.[16]

Additionally, special business process modeling notations that are based on the orchestration languages have evolved to better support the conversion from the business process model into the executable service orchestration. One of these is the Business Process Modeling Notation (BPMN).[8]

1.3 The Missing Methodology

What has been missing until now is a methodology giving a guideline how to analyze businesses and model its processes for orchestrating services. For achieving a perfect alignment of the business processes with the IT structure, in a first step the business processes have to be identified, analyzed and explicitly written down, preferable in the form of diagrams. This process usually starts with a business domain analysis to identify different fields of business processes before modeling the business processes in detail. Then, several refinement layers of the business process activities have to be modeled to describe the business process on a granularity level suitable for a support through Web services.

Finally, the business processes described in a modeling notation have to be converted into an executable description. This also includes binding the services to the business activities. For finding the right Web services, the requirements have to be included as precisely as possible in the business process models. In a first step, the inputs and outputs of the business activities should be specified. For a more detailed description, pre and post conditions as well as non-functional requirements can be included too.

The UMM has been subject to thorough research at our department. It provides a methodology that can be used for intra-organizational process integration. [1] This is why we are interested in analyzing if the UMM is sufficient for SOA, which artifacts are missing and how it could be extended. We will first introduce the UMM, then show its limitations and propose extensions so that required artifacts for intra-organizational service orchestration are created.

2 UN/CEFACT Modeling Methodology

The UN/CEFACT Modeling Methodology (UMM) provides a methodology and models to describe inter-organizational business processes. The idea behind this is to have a unified methodology for identifying and modeling the information exchange between participants when executing a business to business (B2B) process that spans different companies.[17]

2.1 The Methodology

The UMM is based on three views, the Business Domain View, the Business Requirements View, and Business Transaction View. These views present the different perspectives that are used at different stages of the modeling process. The models are based on UML 1.4.2.[18]

Business Domain View. The first view in the process of modeling the business collaboration is the Business Domain View. It takes a very high level perspective. For this view, in a first step, business areas and process areas are identified, using diagrams based on the UML Package Diagram. Then for every process area business processes and stakeholders are identified, using diagrams based on the UML Use Case Diagram.

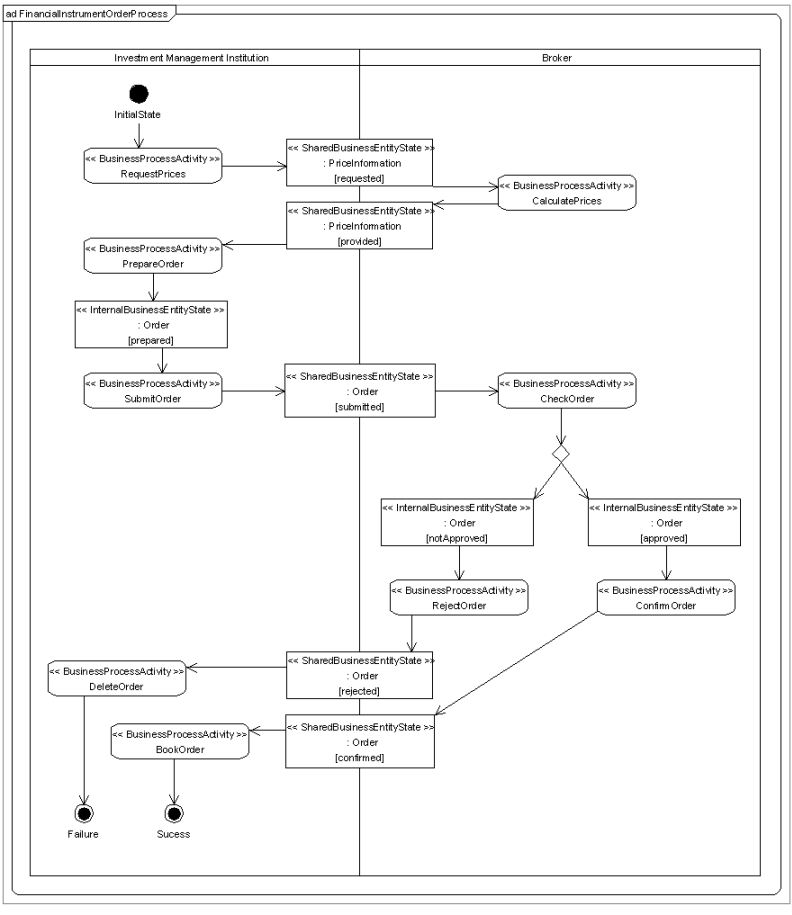

Fig. 1. UMM Business Process View example, from [17]

Business Requirements View. After finishing the Business Domain View, following the methodology, the Business Requirements View has to be completed. It differentiates three views: Business Process View, Business Entity View, and Partnership Requirements View.

The *Business Process View* contains business processes. These processes are modeled as Business Process Activity Models based on the UML Activity Diagram notation. In the process models, entity states are used as defined in the Business Entity View. The *Business Entity View* describes all entities that are used in the Business Process Activity Models, using a diagram based on the UML Class Diagram, and the lifecycles of the entities, using diagrams based on the UML Statechart Diagram. An entity state can either be an Internal Business Entity State for an entity state that is internal to a business process of a single partner, or a Shared Business Entity State for an entity state that is shared between different partners participating in a process. An example Business Process View is displayed in figure 1. Based on the exchange of Shared Business Entity States, requirements for

collaborations between different partners can be identified. Each transaction that transfers information from a partner to another and optionally back will be modeled in more detail in the *Partnership Requirements View*, using diagrams based on the UML Use Case Diagram.

Business Transaction View. After modeling the "Business Requirements View", the "Business Transaction View" is the last stage of the methodology. It consists of three views: Business Choreography View, Business Interaction View, and Business Information View.

The *Business Choreography View* describes the flow of collaborative business activities that have to be performed during a business process involving several partners. The *Business Interaction View* describes the information exchange between two partners to perform a single collaboration. This is where the actual interaction between two organizations is defined. The information entities that can be exchanged are described in the Business Information View. In the *Business Information View* all Information Entities that are exchanged between the partners during a collaboration are modeled. The Information Entities can be mapped to a more formal definition by using Core Components.

2.2 Core Components

Core Components (CC) are, like the UMM, a UN/CEFACT standard.[19] CC are used to semantically describe information that can be exchanged between different business partners. The aim of CCs is to define a common set of information entities that can be reused between different organizations. There are libraries of CC available to semantically harmonize the information entities.[20]

Based on CC, Business Information Entities (BIE) are defined, specifying restrictions on the CC. BIE put CC in a business context, refining the CC according to their specific use.

CC and the UMM are integrated in the way that BIE realize the UMM information entities.

2.3 Service Choreography

Based on the business collaborations modeled in the Business Choreography View and concrete transactions modeled in the Business Interaction View, it is possible to generate a service choreography for the parties involved. The Business Process Specification Schema (BPSS) has been specially developed for this purpose.[21]

What has not yet been envisioned by the UMM is the generation of a service orchestration for a single company. The advantage of this would be to use a single methodology for describing the orchestration of services within a company together with the choreography necessary for the collaboration between companies. This limitation is due to the aim of the UMM.

2.4 Limitations of the UMM

The UMM is aimed at modeling the information exchange between different parties involved in a business process. Until modeling the business processes and information

flow in the Business Process View and the Business Entity View, the methodology is general and can be used either for analyzing the business processes of a single company or the collaboration between different partners. But starting with the Partnership Requirements View, the methodology specializes on the collaboration of the different parties involved. That results in very detailed diagrams about the collaboration, down to the level of a concrete request and an optional response.

For a service orchestration, it is necessary to also specify the business processes internal to the partners involved in more detail. Unfortunately, as the internal business processes are not interesting for the collaboration, it is not foreseen in the methodology to model them in any greater detail. While the UMM is sufficient for modeling collaborative business processes, it comes to its limitations when trying to orchestrate services. A different methodology is necessary for modeling internal business process as a basis of a service orchestration.

3 Modeling Processes for Service Orchestration with the UMM

It would be of great advantage to use a single methodology to model the processes inside a company as well as to model the collaboration between companies. From such models, in addition to the service choreography, a service orchestration could be derived. Additionally, when all the processes are modeled uniformly, it is easier to in- or outsource parts of the process.

3.1 Using the Methodology on a Finer Granularity Level

One idea to enable service orchestration using the UMM is to use the methodology on a finer granularity level. If, for example, the different partners involved in a business process are not different companies, but different departments inside the same company, the transactions analyzed in the Business Transaction View are transactions inside the company. Still, this would result in a service choreography, only this time inside the company and not between companies.

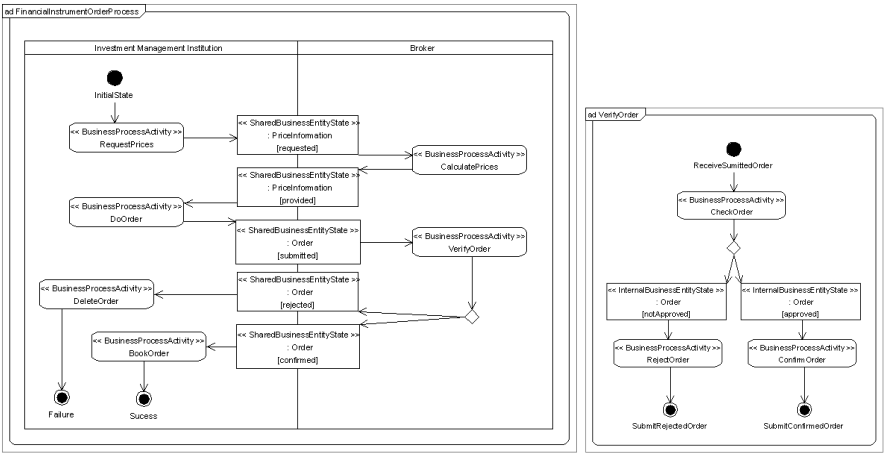

Fig. 2. Modeling business processes with refinements

To model the information exchange necessary for every single business function, and therefore for every single service supporting the business process, every service would have to be interpreted as a different partner involved in the process. While this would be possible for single services, it clearly reaches its limitation when every service has to be modeled as a partner. On the one hand, the models in the Business Domain View and especially in the Business Requirements View would be too complex. On the other hand, modeling every single service as a partner would lead to an explosion of the number of models in the Business Transaction View.

Therefore, for generating a service orchestration out of a UMM model, using the methodology on a finer granularity level does not seem to be a feasible solution.

3.2 Using the Process Model for Service Orchestration

Another approach to generate a service orchestration is to use the process model from the Business Process View. The Business Process Activity Model is based on UML Activity Diagrams. These diagrams are well suited for business process modeling, and a transformation into a service orchestration described in the BPEL exists.[16]

As a Business Process Activity, the activity used in Business Process Activity Models, may contain another Business Process Activity Model, a refinement of activities is possible in the UMM. This can be used to refine process models down to a level where they can be used for service orchestration.

A service orchestration can only be generated for a control flow within a partition (concerning only one partner). Otherwise, it would not be an orchestration but a choreography. Additionally, modeling the activities on a level necessary for a service orchestration is of no use for identifying the need for a collaboration. It would be a better solution to model a process internal to a partner on the highest level as only one activity. This is sufficient for identifying the need for a collaboration. At the same time when refining the activity, it contains a whole process part being performed by one partner. Hence the refinement can be used to generate a service orchestration.

Figure 2 illustrates this. On the left side, a business process with only one activity per partner per collaboration is shown, sufficient to identify necessary collaboration. On the right side, one activity is refined. This refinement can later be used to generate a service orchestration description.

4 Using Core Components for Business Entities

What has been considered for the service orchestration until now is the control flow of internal business processes. Apart from the control flow, Business Process Activity Models describe the business entity flow and business entity states too. This information flow can be used for orchestrating services.

4.1 Modeling the Business Entities for Service Orchestration

Information about the business entity flow is used to identify what kinds of objects are exchanged during a collaborative business process. The concrete structure of the information exchanged between partners is defined at a later stage in the Business Transaction View.

When modeling the entity flow in a business process being used as a basis for a service orchestration, the relevant entities are information entities. In contrast to general business processes where physical products could be exchanged, Web services only exchange information. Therefore only the information flow has to be modeled in the regarded business processes.

For a service orchestration, the information about the data flow can be used to describe the services that are necessary to support the business process. It is therefore helpful to extensively use the Internal Business Entity State, the entity stereotype used internally in a business process, for the modeling of a service orchestration.

Unfortunately, in the UMM there is no official link between the entities modeled in the Business Entity View that are used in the Business Process View and the entities modeled in the Business Information View that are used in the Business Interaction View. But only in the Business Information View the entities are specified in detail. Therefore, a link between a Business Entities from the Business Entity View and an Information Entity from the Business Information View should be established to enable a more detailed description of the information exchange in the Business Process Activity Model. This is possible as the Business Entity View and the Business Information View are based on UML Class Diagrams. It could be envisioned to unify the Business Entity and the Information Entity into one single stereotype when modeling business processes for a service orchestration.

4.2 Using Core Components for a Common Semantic

The flow of information entities is relevant for a service orchestration. For modeling this flow, entities being used as input and/or output of business activities should be modeled using a common semantic. In the UMM, this can be realized by using Core Components. As explained above, a Business Information Entity can realize a UMM Information Entity. By this a common semantic is used to describe the information exchange in the Business Transaction View. Unfortunately, the Business Transaction View defines a choreography of information exchanges only, but doesn't help when creating a service orchestration.

By linking Information Entities with Business Entities as proposed above, the semantics of the CC can be used in the Business Process View. The information flow used for creating the service orchestration is semantically based on CC. This is useful when trying to bind the business activities to concrete services, as input and output parameters are described semantically.

5 Using the BPMN for Business Process Activity Models

Modeling business processes for service orchestrations, another UMM extension can be proposed by replacing the UMM Business Process Activity Model with the Business Process Modeling Notation.

5.1 The Business Process Modeling Notation

The Business Process Modeling Notation (BPMN) is an Object Management Group (OMG) standard.[8] While UML Activity Diagrams were developed to describe

object oriented software systems, the BPMN was developed to model business processes.[22] At the same time, a defined mapping between the BPMN and the BPEL exists.[8, 23]

BPMN and UML Activity Diagrams are very similar. The only pattern that can not be modeled in UML Activity Diagrams is the Interleaved Parallel Routing pattern.[24] As the UML and the BPMN are OMG standards, it is possible that the two notations will be merged at some point.[24] Still, modeling business processes with the BPMN has the advantage that due to its extended syntax complex situations can be model much more explicitly. This results in diagrams that are much more readable.

The BPMN can be used to model private (internal) business processes, abstract (public) processes and collaboration (global) processes. Of interest in this article are collaboration processes and private business processes.

Collaboration processes use at least two swim lanes. In figure 3 a collaboration process is shown. For modeling an information exchange between different parties, a Message Flow with a special arrow is used for message exchanges between the parties. This clearly depicts which flows are needed in a collaboration.

In contrast to the collaboration process, a *private business process* uses one swim lane only, enabling the modeling for internal business processes.

5.2 Using BPMN Instead of Business Process Activity Models

The idea is to replace the UMM Business Process Activity Model with a BPMN model. There are four advantages:

1. Many constructs are much clearer in the BPMN than the equivalent UML Activity Diagram notation when modeling down to a granularity level of services, as special denotations exists e.g. for rollback and messages.[24]
2. The BPMN was developed to be transformed into a service orchestration and has a defined mapping to the BPEL.
3. While the OMG designed UML Activity Diagrams for modeling object oriented software systems, the BPMN is the OMG's standard for business process modeling. Hence one should assume that for modeling business process it is better to use the notation that was designed for this task.
4. The collaborations between partners can be identified more easily using BPMN because of its special message flow notation and the possibility to informally group elements. An example for this is drawn in figure 3.

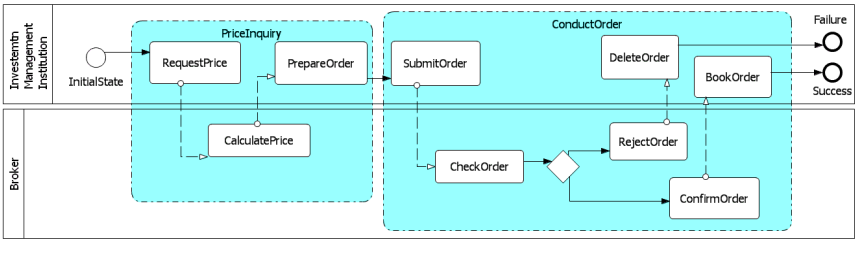

Fig. 3. Using groups for indicating collaboration in the BPMN

Replacing the Business Activity Model by a BPMN model is straight forward as the Business Activity Model is based on UML Activity Diagrams, which BPMN models are very similar to. An example of the business process shown in figure 1 modeled in the BPMN can be seen in figure 3. What is not shown in the example are the BPMN Data Objects that would relate to the Business Entities of the Business Entity View. By relating Business Entities to Information Entities, the BPMN Data Objects can also be based on Core Components, making it possible to semantically annotate the data flow in the same way as in Business Process Activity Models.

6 Summary and Outlook

It is possible to extend the UN/CEFACT Modeling Methodology to better support the modeling and generation of service orchestration. Only few extensions are necessary.

6.1 Propositions

The following extensions of the UMM are proposed:

1. Use the UMM Business Process View model on a granularity level as high as possible for depicting a collaborative business process. Then use refinements to model internal business processes down to a detail level necessary for supporting business activities by services.
2. Establish a link between UMM Business Entities and UMM Information Entities. For a service orchestration only the information exchanged between business activities is of interest. Thus, by establishing the link, the Business Entities can be based on Core Components. This is also useful for a consistent semantic description of the services necessary to support the business activities.
3. Replace the UMM Business Process Activity Model by a Business Process Modeling Notation model. This helps specifying the internal business process, makes it easier to transform the model into an executable service orchestration and even facilitates the identification for necessary collaborations.

6.2 Advantages

By following these extensions of the UMM, the methodology can not only be used to generate service choreographies, but also service orchestrations internal to a business partner. One methodology can be used to model both aspects of a service-priented architecture, the business and its processes as well as the service orchestration and the service choreography. The end-to-end process starting with the business domain analysis and ending with the orchestration and choreography description is supported by one single methodology. This helps businesses, because knowledge of only one technique has to exist and one set of models can be used for business and technical aspects.

6.3 Outlook

There are many points that are still open and subject to further research within this area. The future work of the authors will focus on three of them. The most important

point is evaluating the concept by conducting an empirical study on the applicability of the proposed extensions. For this, secondly, a tool has to be created incorporating all necessary modeling notations to support the methodology. Finally, a semantic description of Web services to be used in conjunction with the proposed modeling technique based on Core Components should be developed.

References

1. Dietrich, J.: Nutzung von Modellierungssprachen und -methodologien standardisierter B2B-Architekturen für die Integration unternehmensinterner Geschäftsprozesse (unpublished Dissertation). TU Berlin, Fachgebiet Systemeanalyse und EDV (2006)
2. International Business Machines Corporation: developerWorks : SOA and Web services (2006), http://www-128.ibm.com/developerworks/webservices
3. Microsoft Corporation: .NET Architecture Center: Service Oriented Architecture (2006), http://msdn.microsoft.com/architecture/soa/
4. BEA Systems: Dev2Dev Online: Service-oriented Architecture (2006), http://dev2dev.bea.com/soa/
5. SAP AG: SAP - Enterprise Service-Oriented Architecture: Blueprint for Service-Based Business Solutions (2006), http://www.sap.com/platform/esa/index.epx
6. World Wide Web Consortium: Web Services Architecture (2004), http://www.w3.org/TR/ws-arch/
7. OASIS Open: OASIS Commitees by Category: SOA (2006), http://www.oasis-open.org/committees/tc_cat.php?cat=soa
8. Object Management Group: Business Process Modeling Notation Specification (2006), http://www.omg.org/cgi-bin/doc?dtc/2006-02-01
9. McCoy, D., Natis, Y.: Service-Oriented Architecture: Mainstream Straight Ahead. Gartner Research (2003)
10. Szyperski, C.: Component Oriented Programming. Springer, Heidelberg (1998)
11. Schmelzer, R.: Solving the Service Granularity Challenge (2005), http:// www.zapthink.com/report.html?id=ZAPFLASH-200639
12. Foody, D.: Getting web service granularity right (2005), http://www.soa-zone.com/index.php?/ archives/ 11-Getting-web-service-granularity-right.html
13. Leymann, F., Roller, D., Schmidt, M.T.: Web services and business process management. IBM Systems Journal 41, 198–211 (2002)
14. Newcomer, E., Lomow, G.: Understanding SOA with Web Services. Addison-Wesley, Reading (2005)
15. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services Version 1.1 (2003), ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf
16. Skogan, D., Gronmo, R., Solheim, I.: Web service composition in UML. Enterprise Distributed Object Computing Conference, 2004. In: EDOC 2004. Proceedings. Eighth IEEE International, pp. 47–57 (2004)
17. UN/CEFACT: UN/CEFACT's Modeling Methodology (UMM) (2006), http://www.untmg.org/index.php?option=com_docman&task=view_category&Itemid=137&subcat=1&catid=63&limitstart=0&limit=5
18. Object Management Group: Unified Modeling Language Specification Version 1.4.2 (2004), http://www.omg.org/cgi-bin/doc?formal/04-07-02

19. UN/CEFACT: ISO\DTS 15000-5: 2006 Core Components Technical Specification 2nd edn. UN/CEFACT Version 2.2 (2006), http://www.untmg.org/index.php?option=com_docman&task=docclick&Itemid=137&bid=43&limitstart=0&limit=5
20. United Nations Economic Commission for Europe: Core Component Library (UN/CCL) (2006), http://www.unece.org/cefact/codesfortrade/codes_index.htm#ccl
21. UN/CEFACT: UN/CEFACT – ebXML Business Process Specification Schema (2003), http://www.untmg.org/index.php?option=com_docman&task=view_category&Itemid=137&subcat=3&catid=63&limitstart=0&limit=5
22. White, S.A.: Introduction to BPMN. IBM Corporation (2004), http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf
23. White, S.A.: Using BPMN to Model a BPEL Process. IBM Corp., United States (2005), http://www.bpmn.org/Documents/Mapping%20BPMN%20to%20BPEL%20Example.pdf
24. White, S.A.: Process Modeling Notations and Workflow Patterns. IBM Corp., United States (2004), http://www.bpmn.org/Documents/Notations%20and%20Workflow%20 Patterns.pdf