Toward a Process Model for GI Service Composition

Udo Einspanier, Michael Lutz, Kristian Senkler,
Ingo Simonis, Adam Sliwinski

Institute for Geoinformatics
University of Münster

{spanier | lutzm | senkler | simonis | sliwinski}@ifgi.uni-muenster.de

Abstract

The composition of individual web services to complex web service chains is a hot topic currently under discussion in the web community. Regarding the geospatial domain, the OpenGIS consortium has published a variety of standards and discussion papers that provide a model to chain geospatial web services. Put into action, it becomes obvious that the proposed model lacks some important features. It does not allow the integration of web services into higher abstraction levels or frameworks. No descriptive language is standardized to define complex chains and executable rules. The OGC discussion does not address semantic aspects at all. This papers aims at presenting concepts to overcome the above-mentioned deficits.

Introduction

Many efforts have been made with regard to standardization in the geospatial domain. Standardized geographic information (GI) services based on Internet technology (i.e. web services) are utilized within many application areas, e.g. in Spatial Data Infrastructure initiatives like GDI NRW (Bernard 2002) or NCGI (Vries et al. 2001). The OpenGIS™ Consortium (OGC, http://www.opengis.org) and ISO/TC 211 (http://www. isotc211.org) have released a number of specifications for interoperable GI services and message encodings. These specifications can be adopted by software vendors for building reliable and sustainable GI services that support users in decision making situations and geoprocessing tasks.

At present, GI services are mostly isolated, stateless applications. However, the greatest value of web services is often seen in their composability, as composition of services can support more complex processing tasks. With these value-added services customers are able to automate, rationalize, and optimize business processes. Unfortunately, concepts for GI service composition (ISO/TC-211 2002) lack some important features. They do not allow the integration of web services into higher abstraction levels or frameworks, no descriptive language is standardized to define complex chains and executable rules, and semantic aspects are not addressed at all.

Outside the geospatial domain, these issues have been addressed by a number of consortia, software vendors, or initiatives. This paper will present the current state-of-the-art approach for each of the limitations outlined above and discuss how they can be applied in the context of geospatial web services.

OGC and ISO RM-ODP

OGC's web service framework and chaining model is based on the "Reference Model of Open Distributed Processing", developed in a joint effort by the international standardization bodies ISO and ITU-T (ISO 1998). RM-ODP specifies concepts and the analytical framework for the normalized description of distributed systems as well as the characteristics that qualify a distributed system as "open". Its objective is the development of standards that allow the distribution of information processing services in an environment of heterogeneous IT resources and multiple organizational domains. This comprises the provision of an underlying infrastructure. The specification of a service includes the representation of data and transport protocols, making the concept of service orientation an essential element of an ODP system.

RM-ODP defines the division of an ODP system into *viewpoints* to simplify its description. This separation of concerns identifies five viewpoints with individual languages:

- *Enterprise Viewpoint*: concerned with business activities and rules;
- *Information Viewpoint*: describes information objects that are saved, processed, and referenced by information sources and sinks;
- *Computational Viewpoint*: describes the system as a set of objects that interact at interfaces and represent information sources and sinks;
- *Engineering Viewpoint*: describes mechanisms to support distribution transparency;
- *Technology Viewpoint*: describes the physical structure of hard- and software components that build up the system.

Each viewpoint is an abstraction that specifies the whole system from its perspective, but is not independent of the other viewpoints. The dependencies are expressed by a set of general concepts and relations that ensure the consistency of the system.

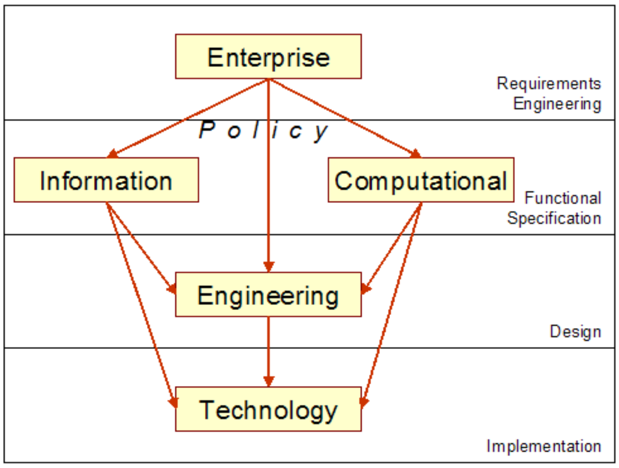

Fig. 1: Relation of RM-ODP Viewpoints to software engineering process

One basic modelling concept is that of an *object* that offers its services at interfaces. The other basic concept is that of an *action*, describing anything of interest that can happen. Actions are linked to at least one object, and can be an *interaction* with other objects or an *internal action*. The collection of actions with a set of constraints when they may occur define an object's *behavior*. An *activity* is a single-headed directed acyclic graph of actions. A *chain (of action*s) is a sequence of actions within an activity. *Subactivities* are subgraphs of activities which are activities themselves. These general concepts are refined and complemented by more specific ones in the different viewpoint languages.

While this framework provides the "big picture" that organizes the pieces of a distributed system as a whole, it does not try to standardize the components of the system or to influence the choice of technology.

The standardization of geospatial processing components is the objective of OGC and ISO 19100. The specifications focus on the information and computational viewpoints. With regard to OGC, common understanding and definitions of RM-ODP related terms are based mainly on the ISO/IEC

14252 report (ISO 1996; Percivall 2002). It is defining service, interface, operation, service chain, and workflow.

A *service* is a distinct part of the functionality that is provided by an entity through interfaces. An *interface* is a named set of operations that characterize the behavior of an entity; an *operation* is a specification of a transformation or query that an object may be called to execute. It has a name and a list of parameters.

Services composed to form a *service chain* are defined as a sequence of services where, for each adjacent pair of services, occurrence of the first action is necessary for the occurrence of the second action. Three types of service chains are distinguished in (ISO/TC-211 2002):

- *User defined (transparent) chaining*. The user manually composes and executes the service chain.
- *Workflow-managed (translucent) chaining.* A pre-defined service chain is executed and controlled by a workflow service.
- *Aggregate Service (opaque chaining).* The service chain appears to the user as a single service.

A *workflow* is an automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules. The term *business process* is used but remains undefined.

Limitations

During the analysis of the chaining model it becomes obvious that it has three deficits that currently prevent effective service chaining:

- No uniform model exists that allows the integration of web services into higher level architectures or business processes.
- No descriptive language exists that allows putting any number of web services into a single or complex service chain. Even using a simple service-service interaction, no rules or execution constraints can be defined.
- There are currently only weak approaches based on shared application schemas for ensuring semantic interoperability between the component services of a service chain.

Outside the geospatial domain a number of approaches exist that address the issues outlined above. In the next sections the state-of-the-art approach for each of these issues will be presented. The comparison of the individual approaches is complicated by the heterogeneous use of technical terms like "service", "action", "activity", and others. Hence, this paper will focus on a presentation of the different approaches and the attempt to map the different usages of the main technical terms and underlying ideas onto the concepts used by OGC and ISO. The results of this lexical comparison can serve as a basis for developing a reference model that allows proper mapping from the different approaches. However, the development of such a reference model is outside the scope of this paper.

XML BASED PROCESS DEFINITION LANGUAGE

Outside the pure web services domain the Workflow Management Coalition (WfMC) has been an active force in defining standard references to facilitate a process definition language, the interchange of process definitions, the interpretation of the process definitions by different workflow management engines, and interoperability across different workflow management systems. The work conducted by WfMC shall allow for developing composite workflow applications across different workflow management systems and organizations which work together as a single logical entity. For this endeavor, WfMC has published the XML Based Process Definition Language (XPDL) (WfMC 2002) and the interoperability specification Wf-XML (WfMC 2001).

Concepts

XPDL belongs to the family of graph-structured process definition languages. The overall goal of XPDL is to specify a basic set of object types that ensure the interchange of simple business process definitions. Additional object types can be added to extend the XPDL formalism. According to WfMC, a process definition is a formal representation of a business process that can be enacted by a workflow management system.

The meta model of XPDL utilizes the concept of activity (*workflow process activity*) as the core component of the process definition (*workflow process definition*). The workflow process definition is build from single workflow process activities that are related together to form a control flow via transitions which are governed by *transition information* (see Fig. 2).

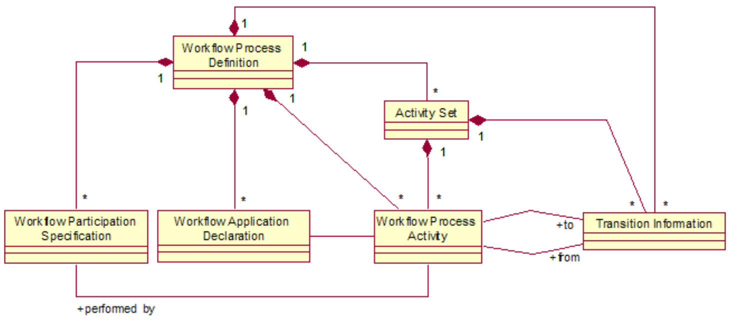

Fig. 2: Excerpt from the XPDL meta model

Following the flow nature of directed graphs, XPDL describes a workflow process definition in terms of what is to be done, when it has to be done, under what conditions, and by whom or what. The latter issue relates to *workflow application declaration* and *workflow participant specification* which are assigned to a *workflow process activity*. The *workflow application declaration* provides descriptions of application interfaces which may be invoked by the workflow enactment service.

The attribute `<ExternalReference>` can be assigned to *workflow application declaration* in order to point at the external definition of an application, e.g. a web service (see the following XPDL code snippet). The *workflow participant declaration* provides a description of human resources which can take part in the process (human in the loop).

```
<DataField Id="abcPO" Name="abcPurchaseOrder" IsArray="False">
    <DataType>
        <ExternalReference xref="PO" loca-
       tion="http://abc.com/services/poService.wsdl"  name-
       space="poService/definitions/types"/>
    </DataType>
</DataField>
```

Limitations

The major limitation of XPDL can be assigned to its mature meta model that has roots in the early 1990s. No doubt, XPDL addresses an array of aspects of enterprise business processes as they appear in the physical world. However, it has few understanding of concepts that are of utmost importance to build business processes on web service architectures, e.g. transaction management or exception handling. This limitation (without making necessary extensions to XPDL) can be claimed to make this language less expressive and powerful than the recently developed business process definition languages, e.g. BPEL4WS (see next section).

Relationship to other specifications

The interaction between applications across different workflow management systems is based on the WfMC's interoperability specification Wf-XML. This specification utilizes a loosely coupled, message-based approach to interoperability. Interaction is referred by WfMC to as bilateral exchange of XML messages between two Wf-XML enabled services. Three types of interaction are supported: request, acknowledgement, and response. These interaction types support both synchronous and asynchronous communication as well as the exchange of individual and batch messages. A request is applied by a service to invoke an operation (and hand over parameters) in another service. A response is used by the invoked service to send the results of an operation to the requesting service. An acknowledgement is applied in asynchronous communication by the invoked service in order to inform the requesting service that the message has been received. The XML dialect specified by Wf-XML can be used to implement three models of interoperability between processes and its services. These include chained services model, nested sub-process model, and parallel synchronized processes model (Hollingsworth 1995).

BUSINESS PROCESS EXECUTION LANGUAGE FOR WEB SERVICES

The Business Process Execution Language for Web Services (BPEL4WS, henceforth BPEL) is a language for web service orchestration released by IBM, Microsoft, and BEA. It supersedes process definition languages XLANG and WSFL and provides means to model the behavior of web services in a business process interaction (Weerawarana and Curbera 2002). It is a XML-based process definition language that allows businesses to describe sophisticated business processes that can both consume and provide web services.

Concepts

By definition, BPEL supports both block structured and graph-structured control flow, introduction flow and link construct. BPEL builds on top of WSDL, with WSDL defining the specific operations allowed and BPEL defining how the operations can be sequenced. The interaction model supported by WSDL is a stateless model of synchronous or uncorrelated asynchronous interactions. Models for business interactions typically assume sequences of peer-to-peer message exchanges, both synchronous and asynchronous, within stateful, long-running interactions involving two or more parties. To define such business interactions a formal description of the message exchange protocols used by business processes in their interactions is needed (Curbera et al. 2002). BPEL leverages WSDL in three ways (Peltz 2003):

- Every BPEL process is exposed as a web service using WSDL that describes the entry and exit points for the process
- WSDL data types are used to describe the information being past within the process
- WSDL might be used to reference external services required by the process

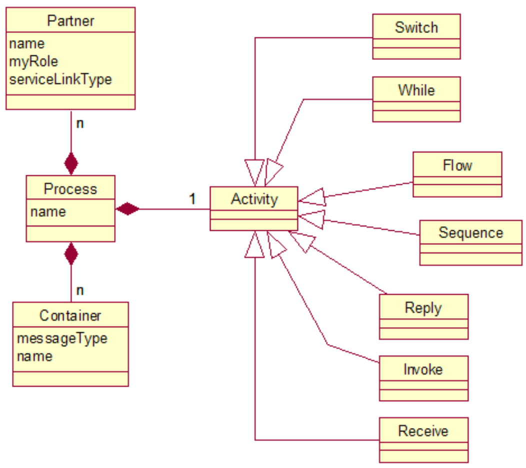

Fig. 3: Excerpt from the BPEL meta model

BPEL supports two kinds of business processes. *Executable business processes* model actual behavior of participant in a business interaction. *Business protocols*, in contrast, use process descriptions that specify the mutually visible message exchange behavior of each of the parties involved in the protocol, without revealing their internal behavior. Business protocols are often defined as *abstract business processes*.

Fig. 3 shows the essential types and relationships of the meta model of BPEL. According to this model, a BPEL process has three main parts: activities (i.e. an operation in a business process), partners (i.e. either a service the process invokes or those that invoke the process) and containers (provide means to store messages that constitute the state of the business process). An activity is by itself decomposed in various elements, e.g. <receive>, <reply>, <invoke>, <sequence> or <flow> (see Curbera et al. 2002 for more details).

As mentioned earlier, BPEL supports a hybrid model that provides both block-structured and graph structured *control flow* mechanisms. The model uses "links" to establish the dependencies between block definitions. The links are defined inside the flow and are used to connect a source activity to a target activity. This means that the interaction with each partner occurs through web service interfaces, and the structure of the relationship at the interface level is encapsulated in service links.

Data flow is handled by containers. Information between different activities is passed in an implicit way through the sharing of data containers. Containers are currently only specified in a global scope within a BPEL process.

Message flow is handled by three types of activities: receive, reply, and invoke. These activities build on top WSDL message types: receive and reply may be assigned to the same operation in case of request/response. Invoke is used to define a solicit response operation type.

Limitations

Since, on the one hand, BPEL offers a broad variety to describe a business process, it is, on the other hand, a very complex language with sometimes too many overlapping structures. Another point is that semantics of BPEL is not always clear. The precise semantics of advanced concepts like serializable scopes leave room for multiple interpretations, thus complicating the adoption of the language (Wohed et al. 2002).

Relationship to other specifications

Next to its core elements BPEL makes use of two further specifications called WS-Transaction and WS-Coordination that deal with how one coordinates the dependable outcome of both short- and long-running- business activities (Snell 2002).

Another process modelling language that has been released by BPMI.org is the Business Process Modelling Language (BPML). BPML and BPEL share similar roots in web services, take advantage of the same XML technologies, and are designed to leverage other specifications like WS-Transactions. Beyond these areas of commonality, BPML supports the modelling of real-world business processes through its unique support for advanced semantics such as nested and complex compensated processes.

DAML-based Web Service Ontology

The DAML-based Web Service Ontology (DAML-S) is a DAML+OIL ontology for describing web services[2], which has been developed collaboratively by a group of researches from BBN Technologies, Nokia, SRI International and Stanford, Carnegie-Mellon and Yale Universities (DAML 2002). The goal of DAML-S is to provide software agents with computer-interpretable descriptions of web services in order to enable automatic discovery, invocation, composition, and execution monitoring of web services.

Concepts

The ontology of services can be divided in three parts, which are characterized, by the kind of knowledge provided about a service:

- The service profile describes what the service requires of users or agents and what it provides to them. The profile specifies the inputs, outputs, preconditions, and effects as well as additional information such as the provider, the quality, or the classification of the service.
- The service model describes how the service works. For specifying the service model the service is viewed as a process. The DAML-S process ontology is discussed in more detail below.
- The service grounding describes how the service can be accessed. The grounding specifies how the abstract inputs and outputs of (atomic) processes are to be realized concretely as messages using the Web Services Description Language (WSDL).

The top-level class of the process ontology (Fig. 4) is a *process*. A process can have any number of *inputs* and *outputs* (that are conditional on the inputs) as well as *preconditions*, which must hold in order for the process to

[2] Subsequent releases will be based upon the Ontology Web Language (OWL), which is being developed by the Web-Ontology Working Group at the World Wide Web Consortium, and will accordingly be called OWL-S

be invoked, and (conditional) *effects*, which represent physical changes that result from the execution of the process. Also, the participants in the process have to be specified.

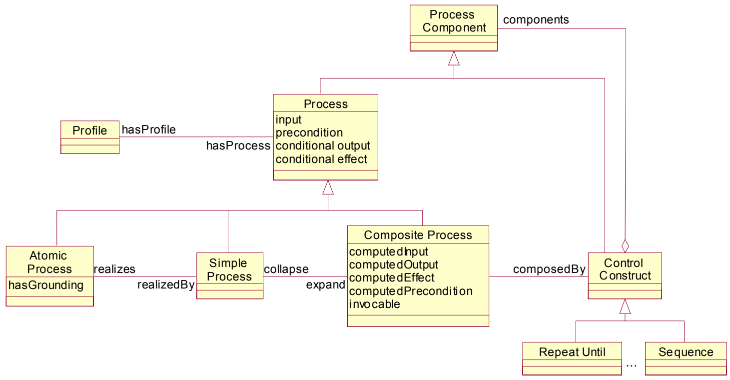

Fig. 4: Top level of the DAML-S process ontology

There are three disjoint subclasses of the class process:

- *Atomic processes* can be directly invoked, have no sub-processes and execute in a single step (from the perspective of the service requester). For all atomic processes a grounding must be specified.
- *Composite processes* can be decomposed into other (atomic or composite) processes, which are linked by control constructs such as *sequence* or *if-then-else*. In contrast to atomic processes, they cannot be directly invoked and are therefore not associated with a grounding.
- *Simple processes*, too, cannot be directly invoked, but, like atomic processes, they are viewed as having single-step executions. Simple processes are used as (specialized or simplified) views of atomic or composite processes for purposes of planning and reasoning.

Limitations

When composing atomic services into composite services it is crucial that the inputs, outputs, preconditions and effects of the component services can be related to each other. In programming languages this is simply done by using variables. DAML+OIL, however, does not provide for the use of variables. In the current version 0.7 of DAML-S this limitation is addressed by introducing a VALUEOF class that is to be used by a specialized DAML-S reasoner to determine which properties should have the same value.

A process control model that would enable an agent to monitor and control the execution of a composite service is also currently missing. However, requirements for such a model are specified in (DAML 2002) and the model is planned to be included in a later version of DAML-S.

Relationship to other specifications

Apart from its connection to WSDL, which is used for specifying the grounding of the specified services, there exist several similarities between DAML-S and BPEL. In fact, DAML-S has been designed to be agnostic with respect to a process model formalism in order to remain compatible with an eventually agreed upon standard for process modeling, e.g. BPEL. A more detailed comparison of the two approaches can be found in (McIlraith and Mandell 2002).

COMPARISON OF CONCEPTS

The approaches presented in the previous sections show how the limitations of the current specifications provided by OGC/ISO could be addressed. However, in order to integrate them into the OGC/ISO architecture a mapping between the extremely heterogeneous concepts and terminologies has to be established. In this section, a first attempt at such a mapping is made. It is based on a lexical comparison of the core concepts of each approach with the concepts used in RM-ODP and OGC. This comparison can serve as a basis for a more thorough analysis and the development of a common reference model.

The comparison is based on the core concepts of each of the approaches described, i.e.

- workflow process activity, transition information, and workflow process definition in XPDL,
- process and activity in BPEL, and
- simple, composite, and atomic process in DAML-S.

Tab 1 gives an overview of the RM-ODP and OGC concepts that we consider to correspond most closely to these core concepts. How these correspondences have to be interpreted and how they are possibly constrained is discussed in more detail in the following subsections.

Tab 1: Concepts in RM-ODP, OGC corresponding to the core concepts employed in XPDL, BPEL and DAML-S

		RM-ODP	OGC
XPDL	workflow process activity	activity	operation
	transition information	action	transformation
	workflow process definition	?	workflow
BPEL	process	chain of actions	translucent/opaque service chain
	activity	action	–
DAML-S	simple process	activity	opaque service chain
	composite process	chain of actions	(generic) service chain
	atomic process	activity	operation

XPDL

XPDL workflow process activities and transition information are components of the workflow process definition that can be executed by workflow management systems. The concept of workflow process activity adheres to activity in RM-ODP and to operation from OGC, respectively. Moreover, the transition information can be assigned to the RM-ODP action and to the OGC transformation, respectively. We also assume the XPDL workflow process definition concept to be closely related to the OGC concept of workflow even though the latter is not precisely defined. In contrast, RM-ODP is considered to lack any concept that most likely corresponds to the concept of workflow process definition from XPDL.

BPEL

As shown in Fig. 3, the basic elements of a BPEL process are activities that represent an operation in a business process. This type of operation is different to the type of *operation* defined by RM-OPD. Since BPEL extends WSDL with the ability of stateful web service interactions, an operation in BEPL is a WSDL operation. An activity in BPEL is better compared to an RM-OPD action that carries out some specific behavior during the execution of a BPEL process. Another point is that BPEL activities provide a richer semantic than a simple operation: activities are both control structures (i.e. <sequence> or <flow>) and message flow constructs (<receive>, <invoke>, <reply>). During the execution of an activity the operations of a web service are invoked.

With regard to service chaining, one can specify processes that are comparable with translucent and opaque service chains as defined by OGC: translucent, since a workflow engine can manage service interactions defined by a BPEL process; opaque, since a user has the ability to define service chains in an abstract or executable manner (BPEL abstract and executable processes).

DAML-S

The DAML composite process compares to the generic concept of service chain, i.e. a composition of several simple services. The DAML concept of simple process relates to the specific concept of aggregate services in OGC, i.e. service chains that appear as a single service to the user. Finally, the DAML atomic process representing an actual executable service with a grounding/binding is equivalent to the OGC concept operation. It should also be noted that the focus of DAML-S is on describing service compositions rather than higher-level workflows (possibly with humans in the loop). There are therefore no equivalents to the notions of workflow in OGC or workflow process definition in XPDL.

Conclusion and Future Research

The geospatial domain lacks crucial concepts that facilitate the composition of complex service chains. The paper recognizes three key shortages that are presently exposed by the work of OGC. These shortages need to be challenged in order to facilitate the building of service chains.

Outside the geospatial domain there is remarkable work under way aimed at building business processes on web service architectures. This paper addresses three state-of-the-art approaches and puts emphasis on the work conducted by the Workflow Management Coalition (WfMC), a group formed by IBM, Microsoft, and BEA, and The DAML Services Coalition (DAML). Therefore, the fundamental concepts from XPDL, BEPL, and DAML-S are introduced and compared against those from RM-ODP and ISO. This comparison allows us to put forward initial ideas for how the concepts defined and maintained by OGC may be supplemented and strengthened in order to overcome the shortages. However, the comparison in this paper remains at the entity level. We are well aware of the fact that a more detailed analysis of the concepts remains necessary. This, most notably, includes a comparison at the property and relationship level. Also only the general RM-ODP concepts are taken into account. For a complete comparison the viewpoint-specific concepts have to be considered as well. However, such a comparison is well outside the scope of this paper and will be dealt with in future research.

REFERENCES

Bernard, L. (2002): *Experiences From an Implementation Testbed to Set Up a National SDI*. 5th AGILE Conference on Geographic Information Science 2002, Palma de Mallorca, pp. 315-321.

Curbera, F., Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, S. Weerawarana (2002): *Business Process Execution Language for Web Services, Version 1.0*. URL: http://www.ibm.com/developerworks/ library/ws-bpel/ (Last accessed: March 2003).

DAML (2002): *DAML-S: Semantic Markup for Web Services (DAML version 0.7)*. The DAML Services Coalition.

Hollingsworth, D. (1995): *The Workflow Reference Model v1.1*. The Workflow Management Coalition (WfMC).

ISO (1996): *Information technology — Guide to the POSIX Open System Environment (OSE).* ISO/IEC 14252.

ISO (1998): *Open Distributed Processing Reference Model.* ISO/IEC 10746.

ISO/TC-211 (2002): *Geographic Information - Services*. ISO/DIS 19119.

McIlraith, S. A., D. Mandell (2002): *Comparison of DAML-S and BPEL4WS*. URL:http://www.ksl.stanford.edu/projects/DAML/ Webservices/DAMLS-BPEL.html (Last accessed: Mai 2003).

Peltz, C. (2003): *Web Service Orchestration*. URL: http://devresource.hp.com/drc/technical_white_papers/WSOrch/ WSOrchestration.pdf (Last accessed: March 2003).

Percivall, G. (2002): *OpenGIS Abstract Specification. Topic 12: OpenGIS Service Architecture.* Open GIS Consortium.

Snell, J. (2002): *Automating business processes and transactions in Web services.* URL: http://www-900.ibm.com/developerWorks/cn/ webservices/ws-autobp/index_eng.shtml (Last accessed: November 2002).

Vries, M.E.d., T.P.M. Tijssen, J.E. Stoter, C.W. Quak, P.J.M.v. Oosterom (2001): *The GML Prototype of the New TOP10 Vector Model*. http://kartoweb.itc.nl/top10nl/TOP10NL_eng/download/pdf/GML_prototype.pdf (Last accessed: April 2003).

Weerawarana, S., F. Curbera (2002): *Business processes: Understanding BPEL4WS, Part 1*. URL: http://www-900.ibm.com/developerWorks/cn/webservices/ws-bpelcol/part1/index_eng.shtml (Last accessed: November 2002).

Wohed, P., W. M. P. v. d. Aalst, M. Dumas, A. H. M. t. Hofstede (2002): *Pattern Based Analysis of BPEL4WS*. URL: http://xml.coverpages.org/AalstBPEL4WS.pdf (Last accessed: January 2003).

WfMC (2001): *Interoperability Wf-XML Binding*. The Workflow Management Coalition (WfMC).

WfMC (2002): *Workflow Process Definition Interface -- XML Process Definition Language (XPDL)*. The Workflow Management Coalition (WfMC).