Employment of Semantic Web services for workflow systems
Essay in DT8107 Distributed Information Systems

Rune Molden

Department of Computer and Information Science
Norwegian University of Science and Technology (NTNU)
N-7491 TRONDHEIM, NORWAY
E-mail: runemol@idi.ntnu.no

ABSTRACT

The current Web technologies do not satisfy the demands for machine interpretable information resources and Web services. The increase of enterprises collaborating across organizational borders requires interaction in order to do business more efficiently. The Semantic Web provides several concepts which can deal with this scarcity. In this essay, the most important building blocks of the Semantic Web are presented along with the Workflow Reference Model. Some concepts of workflow systems utilizing Semantic Web services are also presented. Finally, the concepts are discussed in relation to some issues concerning standardization and a "global embracing" of the Semantic Web as the future Web.

1 INTRODUCTION

The WWW has emerged as the dominant information repository, and is heavily utilized both in private and professional relations. Everybody can contribute by publishing information and applications to the Web, and this is probably the motive of the rapid growth and success of the Web. Unfortunately, the information on the Web is not organized very well and, thus, the desired information is hard to find. In most cases, information search yields too much, too little or the information is of questionable quality. When it comes to automated information retrieval it is even worse, as the majority of the information available is not encapsulated by a standardized set of descriptions or metadata. Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully [3].

The vision of the Semantic Web is to deal with this inconvenience by bringing in structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users [3]. Hopefully, this will improve both manual and automated information retrieval in the future.

A trend in the global market is the increase of collaboration among enterprises. Enterprises transform themselves into "networked organizations" in order to boost their flexibility and reduce costs. To achieve this, enterprises need to interact with a community of buyers, sellers and partners, and thus enterprise information systems and applications need to be interoperable in order to achieve seamless business interaction across organizational boundaries.

In enterprises generally, information and knowledge is requisite for decision making [2]. Enhanced workflow systems for supporting different business processes and autonomous applications like Semantic Web services can be employed to share functionality across organizational borders to solve appurtenant tasks for the networked organizations. Here, we will give a brief presentation of the building blocks of the Semantic Web, and the concepts of a workflow management system, and finally, look at some of the current approaches for utilization of Semantic Web services for workflow systems.

Section 2 deals with an introduction to the core of the Semantic Web with emphasis on Semantic Web services, section 3 present workflow theory and concepts, section 4 deals with some existing efforts to a coalition of workflows and Semantic Web resources. Section 5 will discuss the concepts and conclude this essay.

2 SEMANTIC WEB

The Semantic Web is not a separate, new Web, but an extension of the current Web [3]. It provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries. It is a collaborative effort led by W3C[1] with participation from a large number of researchers and industrial partners [1]. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming [1]. Figure 2-1 depicts the Semantic Web Tower and its layers. Here, we will not cover all of the layers, but will look at the "RDF + rdfschema", the "Ontology vocabulary" and the "Logic" layers. The "Proof" and "Trust" layers are also of great importance, especially concerning Semantic Web applications and information that has to be trustworthy (e.g. processes and flows in conjunction with B2B interactions, payment transactions, confidential information and other transactions-like processes that has devastating effects if they are abused).

Further, we will present some of the building blocks technologies which constitute the Semantic Web.

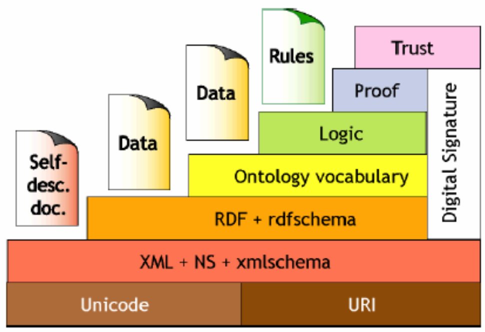

Figure 2-1: The Semantic Web Tower

2.1 RESOURCE DESCRIPTION FRAMEWORK (RDF)

The Resource Description Framework (RDF) is a framework for representing information on the Web. In this, it is meant that RDF is particularly intended for representing metadata about Web resources. In conjunction with this, metadata could be the title of a Web page, the author of it and modification date of the page. RDF can also be used for things that can be identified on the Web, even though they can not be directly retrieved there. Resources refereed to as things, can be items available from an online Web shop. By using RDF for description of goods available in that Web shop, the information about a particularly product is also available for machine interpretation. RDF uses XML as carrier syntax, but do not employ any semantics expressed by XML [10]. Instead, RDF uses own semantics for XML syntax [10].

RDF is based on the idea of identifying things using Web identifiers called Uniform Resource Identifiers (URIs). In RDF, resources are described in terms of simple properties and property values. RDF has a simple and easy to understand metadata structure which is based on three assumptions:

- **Resource**: Is anything that can have a URI such as a Web Service or a Web page.
- **Property**: A property of the thing that the statement describes.
- **Statement**: Is a link between the resource and its property.

An RDF statement has the following representation:

- **Subject**: An URI that indicates a resource.
- **Predicate**: The property of the subject.
- **Object**: The value of the property.

[1] World Wide Web Consortium, http://www.w3c.org/

Figure 2-2 below illustrates an RDF graph describing a Web page and its metadata. The nodes represent subjects or objects and the arcs represent predicates, directed from the subject node to the object node.

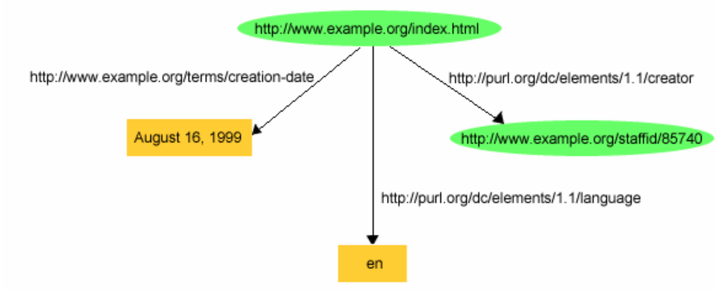

Figure 2-2: An RDF graph describing a Web page [7]

In natural language, the graph above can be represented as: "*http://www.example.org/index.html* has creation date whose value is *August 16, 1999*, has a language whose value is *English* and was created by a person represented by the staff id *85740* within the 'example.org' domain". The corresponding RDF/XML for this example would be:

```
01.  <?xml version="1.0"?>
02.  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
03.              xmlns:dc="http://purl.org/dc/elements/1.1/"
04.              xmlns:exterms="http://www.example.org/terms/">

05.    <rdf:Description rdf:about="http://www.example.org/index.html">
06.         <exterms:creation-date>August 16, 1999</exterms:creation-date>
07.         <dc:language>en</dc:language>
08.         <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
09.    </rdf:Description>

10.  </rdf:RDF>
```

There are several features of RDF, that possibly make it a more powerful metadata framework for identifying resources on the Web, than the previous metadata approaches. Some of the advantages of RDF are extracted below [14]:

- **Based on triples.** The most commonly known metadata formats are based on attribute value pairs. Using triples, makes the subject of the attribute value pair explicit.

- **Distinguishes between resources and properties that are globally qualified.** A globally qualified resource or property can be distinguished from other resources or properties in different vocabularies that share the same fragment name. This is also available in XML by means of XML namespaces.

- **Can be used to make statements about resources.** Resources include documents on the Web, such as relating one URI to another.

- **Easy to encode graphs using RDF.** This is due to that RDF is based on triples, whereas XML documents are trees, thus encoding graphs is more complicated and can be done in several different ways.

- **An explicit interpretation or model theory.** There is an explicit formal interpretation of a RDF model. XML documents do also have interpretations, but they are implicit in the processor or the parser associated with the particular type of the XML document.

For more thorough information on RDF, see the W3C RDF Primer [7] or other resources on the topic.

2.2 RDF SCHEMA

RDF Schema provides a type system for RDF, but does not provide a vocabulary for application-specific classes [7]. Instead, it provides the facilities needed to describe classes and properties, and which classes and properties that are expected to be used together [7]. RDF Schema is similar in some respect to the type systems of object-oriented programming languages such as Java [7]. As for RDF, RDF Schema also uses own semantics for XML syntax [10]. The classes and properties of the RDF Schema definition are [11] lined up below in Table 2-1 and Table 2-2 respectively:

Class name	Comment
`rdfs:Resource`	The class resource, everything.
`rdfs:Literal`	The class of the literal values, e.g. textual strings and integers.
`rdf:XMLLiteral`	The class of XML literal values.
`rdfs:Class`	The class of classes.
`rdf:Property`	The class of RDF properties.
`rdfs:Datatype`	The class of RDF datatypes.
`rdf:Statement`	The class of RDF statements.
`rdf:Bag`	The class of unordered containers.
`rdf:Seq`	The class of ordered containers.
`rdf:Alt`	The class of containers or alternatives.
`rdfs:Container`	The class of RDF containers.
`rdfs:ContainerMembershipProperty`	The class of container membership properties, rdf:_1, rdf:_2, …, all of which are sub-properties of 'member'.
`rdf:List`	The class of RDF lists.

Table 2-1: RDF Schema classes [11]

Property name	Comment	Domain	Range
`rdf:type`	The subject is an instance of a class.	`rdfs:Resource`	`rdfs:Class`
`rdfs:subClassOf`	The subject is a subclass of a class.	`rdfs:Class`	`rdfs:Class`
`rdfs:subPropertyOf`	The subject is a subproperty of a property.	`rdf:Property`	`rdf:Property`
`rdfs:domain`	A domain of the subject property.	`rdf:Property`	`rdfs:Class`
`rdfs:range`	A range of the subject property.	`rdf:Property`	`rdfs:Class`
`rdfs:label`	A human-readable name for the subject.	`rdfs:Resource`	`rdfs:Literal`
`rdfs:comment`	A description of the subject resource.	`rdfs:Resource`	`rdfs:Literal`
`rdfs:member`	A member of the subject resource.	`rdfs:Resource`	`rdfs:Resource`
`rdf:first`	The first item in the subject RDF list.	`rdf:List`	`rdfs:Resource`
`rdf:rest`	The rest of the subject RDF list after the first item.	`rdf:List`	`rdf:List`
`rdfs:seeAlso`	Further information about the subject resource.	`rdfs:Resource`	`rdfs:Resource`
`rdfs:isDefinedBy`	The definition of the subject resource.	`rdfs:Resource`	`rdfs:Resource`
`rdf:value`	Idiomatic property used for structured values.	`rdfs:Resource`	`rdfs:Resource`
`rdf:subject`	The subject of the subject RDF statement.	`rdf:Statement`	`rdfs:Resource`
`rdf:predicate`	The predicate of the subject RDF statement.	`rdf:Statement`	`rdfs:Resource`
`rdf:object`	The object of the subject RDF statement.	`rdf:Statement`	`rdfs:Resource`

Table 2-2: RDF Schema properties [11]

2.3 OWL WEB ONTOLOGY LANGUAGE (OWL)

The first level above RDF is an ontology language that can formally describe the meaning of terminology used in Web documents. Ontology, a term borrowed from philosophy, refers to the science of describing the kinds of entities in the world and how they are related [15]. OWL adds more vocabulary for describing properties and classes, and provides relations between classes (e.g. disjointness), cardinality (e.g. exactly once), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes [8].

OWL is a semantic markup language for publishing and sharing ontologies on the World Wide Web [8]. It is intended to be used when information contained in a document needs to be processed by applications. OWL can be used to represent the meaning of terms in vocabularies and the relationships between those terms. Such a representation of terms and interrelationship is called an ontology. OWL is a revision of the DAML+OIL ontology languages which incorporates lessons learned from the design and application of DAML+OIL [8].

An ontology is a knowledge representation, and differs in that way from XML schema, which is a message format [15]. The majority of industry based Web standards include a combination of message formats and protocol specifications [15]. These formats have been given an operational semantics, and they are not designed to support reasoning outside of the transaction context [15]. With OWL ontologies, tools will have the ability to reason about them [15]. The tools will provide generic support that is not specific to the particular subject domain [15].

OWL provides three sublanguages designed for use by specific communities of implementers and users [15]:

- **OWL Lite** support primary needs for a classification hierarchy and simple constraints. It only provides cardinality values of 0 or 1, but it should be simpler to provide tool support for OWL Lite than its more expressive relatives.

- **OWL DL** provides a maximum of expressiveness without loss of computational completeness and decidability of reasoning systems. Further, it includes all OWL language constructs with restrictions such as type separation, i.e. a class can not also be an individual or property and a property can not also be an individual or class. OWL DL has its name due to its correspondence with description logics, hence the DL acronym. It as designed to support existing Description Logic business segment and has desirable computational properties for reasoning systems.

- **OWL Full** makes maximum expressiveness and the syntactic freedom of RDF with no computational guarantees possible. Here, a class can be treated simultaneously as a collection of individuals and as an individual in its own right. Other significant differences from OWL DL is that an *owl:DatatypeProperty* can be marked as an *owl:InverseFunctionalProperty*. Further, it allows an ontology to augment the meaning of the predefined (RDF or OWL) vocabulary. However, it is unlikely that any reasoning software will be able to support every feature of OWL Full.

2.4 OWL BASED WEB SERVICE ONTOLOGY (OWL-S)

OWL-S, originally called DAML-S, is an ontology of services that describes Web services in terms of what they can do, how they work and how they can be caught up [9]. OWL-S builds on the Ontology Web Language (OWL) recommendation composed by the Web-Ontology Working Group at the World Wide Web Consortium (W3C) [9]. OWL-S is expected to enable to following tasks and features [9]:

- **Automatic Web service Discovery**: automatic location of Web services that provide a particular service that adhere to requested constraints.

- **Automatic Web service invocation:** automatic execution of an identified Web service by a computer program or an agent.

- **Automatic Web service composition and interoperation:** automatic selection, composition, and interoperation of Web services to perform some task, given a high-level description of an objective.

- **Automatic Web service execution monitoring:** provide status of single or composite Web service requests.

The last item in the point list above has unfortunately not been supported by the prevailing versions of OWL-S, but nevertheless it is regarded as an important issue. The scopes above, except from the latest one, are performed by OWL-S by description of the services with a set of classes and properties.

Figure 2-3 below depicts the OWL-S infrastructure and its relations to other components of the Web service infrastructure. The OWL-S Web service requires the specification of the modules Service Profile, Process Model and Service Grounding which are described more thorough below. The OWL-S Service description connects the three mentioned modules [16].

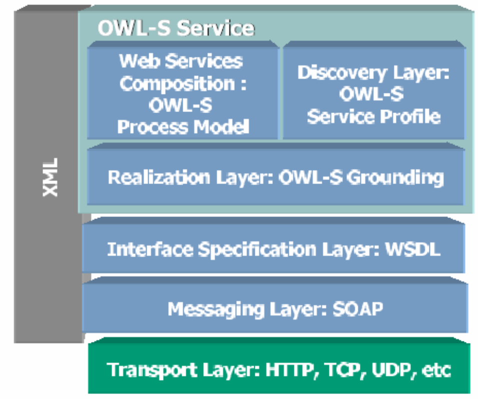

Figure 2-3: The OWL-S infrastructure [16]

OWL-S can be divided into three main building blocks [16]:

- **Service Profile**: a high level view of a Web Service. This covers provenance, capabilities of the Web Service and discovery related information which provides help for discovery of the service. Service Profile is the OWL-S analog to UDDI in the traditional Web services architecture, but there are some differences like UDDI doesn't provide capability-related information. UDDI describes information of the ports of the Web Service, and in OWL-S this is relegated to Grounding and WDSL description.

- **Process Model**: description on what the Web service does. This includes specification of tasks, control flow, order of task-performance and consequences of each task described as input, output preconditions and effects. The needed choreography can be derived from the Process Model, i.e. patterns of message exchange with the Web service by figuring out what inputs the Web services expects, when it expects them and what outputs it reports and when. This is quite similar as the emerging standards such as BPEL4WS and WSCI, but Process Model maintains a stronger focus on the semantic description of service choreography and the effects of the execution of the different components of the Web service.

- **Service Grounding**: binds the description of abstract exchange of information between the Web service and its partners. These descriptions are defined in terms of inputs and outputs in the Process Model, explicit messages specified in the WDSL description of the Web service, the SOAP message and transport layers.

3 WORKFLOW MANAGEMENT SYSTEMS

Workflow management is a fast evolving technology which is increasingly being exploited by businesses in a variety of industries [13]. The primary characteristics of workflow management are automation of processes involving combinations of human and machine-based activities, particularly those involving interaction with applications and tools [13].

The Workflow Management Coalition (WfMC) defines workflow as: "The 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." [4]. Further, they define a Workflow Management System as: "A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications." [4]. The core terminology of the WfMC is shown in Figure 3-1 bellow.

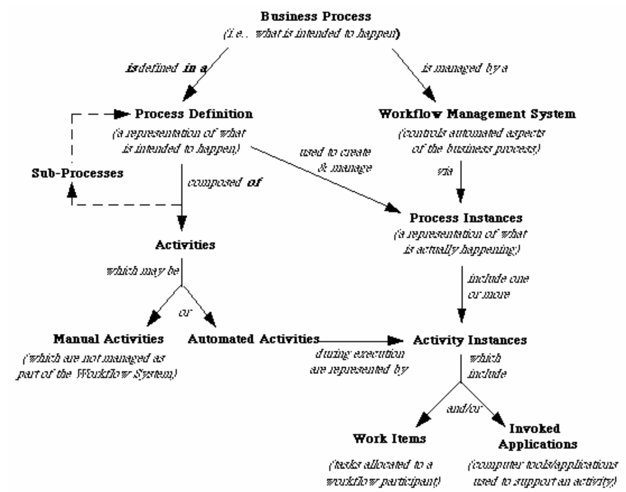

Figure 3-1: WfMC Core terminology

The major components and interfaces within workflow architecture are displayed in Figure 3-2 below. The components in the reference model will be deepened, by explaining their role and purpose within a workflow system and is briefly extracted from the Workflow Reference Model [13]:

Workflow Enactment Service is a software service that may consist of one or more workflow engines in order to create, manage and execute workflow instances. Applications may interface to this service via the workflow application programming interface (WAPI).

Workflow Engine is a software service or "engine" that provides the run time execution environment for a workflow instance.

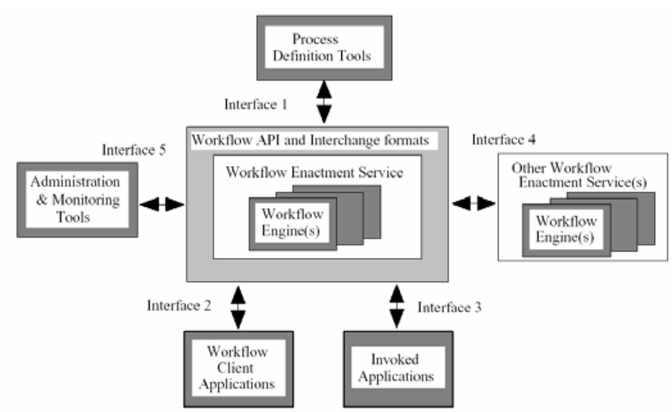

Figure 3-2: Workflow Reference Model - Components & Interfaces [13]

Process Definition Tools are a variety of tools that enables analysis, modeling, descriptions and documentation of a business process. The particular nature of such tools or how they interact during the build time process, are not concerned by the workflow model. The tools can vary from informal ("pencil and paper") to sophisticated and highly formalized, and may be supplied as a part of a particular workflow product.

Workflow Client Applications is software entities which interact with the end-user in those activities which require involving human resources. This software may be supplied as part of the workflow management software or can be developed at the enterprise using the system. In other cases, workflow mat be integrated into a common desktop environment alongside other office services (e.g. mail, work-in-progress folder, unified task managements systems). Thus, there is a need for a flexible mechanism of communication between a workflow enactment service and workflow client applications.

Administration & Monitoring Tools is tools for performing administrative and monitoring tasks. This involves user, role, and audit management. Further, it involves resource control operations for process or activity levels and interrogation of resource control data, process supervisory functions (e.g. changing the operational status of workflow process definitions, enabling of disabling particular versions of a process definition) and process status functions (e.g. opening/closing a process or activity instances query, setting optional filter criteria, fetching details of process or activity instances).

Invoked Applications is applications enabled to directly be invoked by the workflow engine, or applications that is invoked by the workflow engine through an application agent.

Workflow Definition Interface (Interface 1) is an interface between the modeling and definition tools, and the workflow management system. The interface is termed as the process import/export interface, and may support the exchange of a complete process definition or a subset.

Workflow Client Application Interface (Interface 2) is an interface for enabling interaction between workflow clients and the workflow engines and worklists. This can be achieved by WAPI, or, for example by mail standardized interchange formats. However, the interface is to provide command set support, connection/disconnection, process and activity control functions, process status function and worklist manipulation commands.

Invoked Application Interface (Interface 3) is an interface that is intended to be applicable to application agents and applications which have been designed to be "workflow enabled", i.e. to interact directly with a workflow engine.

WAPI Interoperability Functions (Interface 4) is the interface to which is to provide support for interaction between heterogeneous workflow systems. This implies activity and sub-process invocation, process/activity status/control, application/workflow relevant data transfer, synch point coordination and process definitions read and write.

Administration and monitoring interface (Interface 5) provides support for independent management applications to interact with the workflow domain. Alternatively, other implementation scenarios are also feasible, for example the management application may be an integral part of one enactment service, although capable of managing various functions across additional (heterogeneous) workflow domains. The typical functions areas are user/role/audit management, resource control and process supervisory function, but may also manage workflow process definitions, acting as a repository and distributing process definitions to the various workflow domains via operations within interface 1.

The given presentation of the workflow reference model has a rather course-grained level of detail, and it is therefore recommended to consult the Workflow Reference Model [13] for a more thorough presentation.

4 WORKFLOWS AND SEMANTIC WEB SERVICES

There have been several initiatives to utilize traditional Web services for workflow systems. One of the enabling technologies is the Business Process Execution Language for Web Services (BPEL4WS). Explained briefly, BPEL4WS/BPEL defines a language for creating service compositions in the form of business processes [6]. Each composition of BPEL is a business process or a workflow that is able to interact with a set of Web services to achieve a certain goal [6]. There has been some effort done trying to adapt the BPEL for the Semantic Web in [12] and [17].

The motivation of utilizing Semantic Web services in coalition with workflow is to support the networked organizations and their ever changing business environment. There are several challenges which have to be solved in order to meet the demands for realization of global processes. Some of the most important of them are [5]:

- Heterogeneity and Autonomy
- Dynamic nature of business interactions

- Scalability

One solution is the use of ontologies to overcome semantic problems that arise from the autonomy, heterogeneity, and distribution of Web services [5].

This section presents example of an architecture [12] for enabling dynamic workflow composition by using Semantic Web Services, and a Semantic Web process lifecycle [5].

In [12], they present an enabling architecture for dynamic workflow composition by utilizing Semantic Web services and this architecture is depicted in Figure 4-1 below. The information flow between the entities in the architecture is explained in detail corresponding to the numbers depicted [12]:

1. When a new Web Service is created, it is advertised by registering the WDSL and DAML-S[2] descriptions to a directory, e.g. UDDI.
2. Workflows are stored in the "Workflow Repository", from where they are brought into the workflow composer agent.
3. Each workflow consists of one or more Web services. These Web services can be situated anywhere on the Web. DAML-S and WDSL descriptions of the Web services are queried from the directory and a semantic matching is applied.
4. The "Workflow Composer Agent" composes the executable workflow, and brings it into the "Workflow Execution Engine.
5. The "Workflow Execution Engine" executes the workflow using Web services instances which constitutes the workflow.

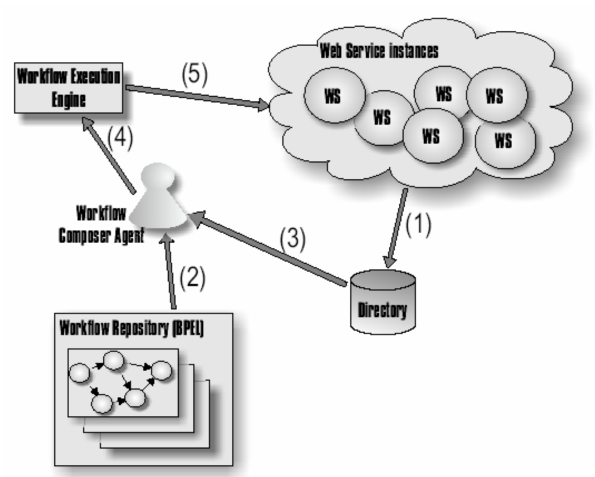

Figure 4-1: Enabling architecture for dynamic workflow composition [12]

Further, [12] propose some general steps to the process of replacement of an unavailable Web services in a workflow and creation of new workflows:

- Identify the required functionality
- Semantic matching of Web services
- Create or update a workflow
- Execution and monitoring of workflows

[2] DAML-S is the predecessor of OWL-S, which is described in section 2.4.

They conclude that their presented architecture requires human interaction in the initial phase of the workflow composition, which is the part concerning identification of required functionality.

In [5], they present the Semantic Web process lifecycle which includes annotation of services, advertisement, discovery, selection, composition and execution. Web processes allows complex interactions among organizations and represents the evolution of workflow technology [5]. Their introduction of the functional semantics for Web service discovery, might handle the demand of human interaction mentioned in [12]. Figure 4-2 gives an overview of this lifecycle process. The different semantic concepts of this Web process lifecycle, which is extracted from [5], will be explained in more detail:

Functional semantics: Web services should be discovered based on their functional requirements, not only by their inputs and outputs. Web services could be annotated with functional semantics, by using an ontology (Functional Ontology [5]).

Data semantics: To perform discovery of services in an efficient way, semantics of the input and output data has to be taken into account. If the data involved in the Web service operation is annotated using an ontology, a matching between semantic representation of input/output data of the Web service and the semantics of the input/output data of the requirements can be performed.

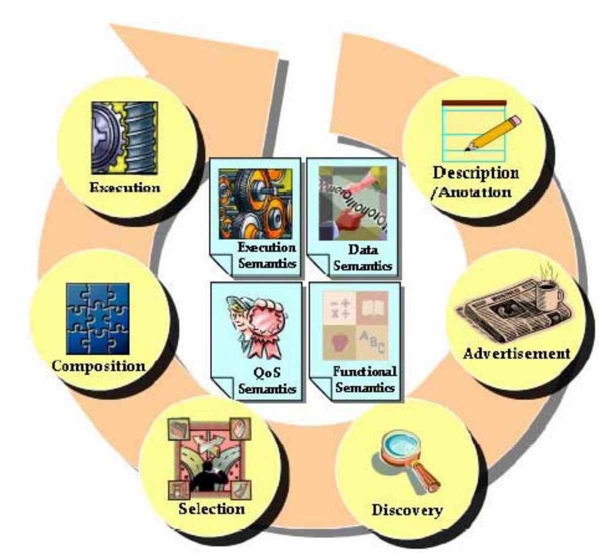

Figure 4-2: Web process lifecycle and semantics [5]

QoS semantics: These semantics helps to select the most suitable service based on aspects of quality. This means that the service that matches the best quality criteria is chosen. This requires QoS metrics management of Web services.

Execution semantics: Encompasses the ideas of instance message sequence, conversation pattern of Web service execution, flow of actions, preconditions and effects of invocation. Execution semantics will not be the same for all services. These semantics will help in dynamically finding services based on operational requirements like long running interactions and complex conversations. It will also provide help for coordinating activities in transactions that involve multiple parties.

5 DISCUSSION AND CONCLUSION

In this essay a majority of the building blocks included in the Semantic Web notion and Workflow Reference Model has been presented. In addition, some issues and concepts regarding utilization of Semantic Web services for workflow systems has been presented.

The Semantic Web envisions undoubtedly a great improvement of the current Web concerning computerized information retrieval, computerized discovery and selection of Web services. The use of Semantic Web technology to enable computers to reason about information resources and services, will hopefully enable automation of business process across organizational boundaries to some extent. However, human interaction will be needed in some degree, but they will hopefully spend minor effort in manual discovery and retrieval. Their effort will to some extent be to control that the computer systems are performing the reasoning correctly and confirm that the commitment of the process can be carried out.

Moreover, there are some issues concerning standardization. The World Wide Web Consortium (W3C) is conducting most of the work concerning the Semantic Web. However, there are important concepts like the OWL-S[3] and the BPEL4WS[4], which is composed and conducted by other parties. One of the great things about the current Web is its cross-platform compatibility, which enables global utilization and contributions. It is important that the industry join the common effort to urge the development of Semantic Web technology to make sure it will be adopted as a global standard.

The Semantic Web is a promising approach to achieve a better Web community and to face the future demands, but there should be spent more effort on research and development concerning Semantic Web based workflow, security, performance, global ontology agreement, scalability and standardization, and agreement.

REFERENCES

[1] Miller et al. (2004): *Semantic Web*, W3C, http://www.w3c.org/2001/sw/

[2] Debenham, J. (1998): *Controlling a Web-Based Workflow System,* Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference

[3] Berners-Lee, T., Hendler, J. and Lassila, O. (2001): *The Semantic Web*, Scientific American, May 2001

[4] Lawrence, P. (1997): *Workflow Handbook 1997*, Workflow Management Coalition. John Wiley and Sons, New York, 1997

[5] Cardoso J. and Sheth, A (2004): *Introduction to Semantic Web Services and Web Process Composition*, Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg (in print 2005)

[6] Cubera et al. (2003): *The next step in Web Services*, Communications of the ACM

[7] Manola, F. and Miller E. (2004): *RDF Primer, W3C Recommendation 10 February 2004*, W3C, http://www.w3.org/TR/rdf-primer/

[8] Bechhofer et al. (2004): *OWL Web Ontology Language Reference: W3C Recommendation 10 February 2004*, W3C, http://www.w3c.org/TR/owl-ref/

[9] Martin et al. (2004): OWL-S: Semantic Markup for Web Services, http://www.daml.org/services/owl-s/1.1/overview/

[10] Schneider, P. F. and Fensel, D. (2002): *Layering the Semantic Web: Problems and Directions*, Springer-Verlag Berlin Heidelberg

[11] Brickley, D. and Guha, R.V. (2004): *RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004*, W3C, http://www.w3c.org/TR/rdf-schema/

[12] Laukkanen, M. and Helin, H. (2003): *Composing Workflows of Semantic Web Services*, Workshop on Web Services and Agent-based Engineering

[13] Holligsworth, D (1995): *The Workflow Reference Model*, Workflow Management Coalition (WfMC)

[14] Butler, M. H. (2002): *Barriers to real world adoption of Semantic Web technologies*, Technical Report, Hewlett Packard

[15] Smith, M. K., Welty, C. and McGuinness, D. L. (2004): *OWL Web Ontology Language Guide: W3C Recommendation 10 February 2004*, W3C, http://www.w3.org/TR/owl-guide/

[16] Sycara, K. (2004): *Dynamic Discovery, Invocation and Composition of Semantic Web Services*, Springer-Verlag Berlin Heidelberg

---

[3] OWL-S has recently been submitted as an "Acknowledged Member Submission to W3C", see http://www.w3.org/Submission/

[4] BPEL4WS is an initiative by BEA, IBM, Microsoft, SAP AG and Siebel Systems, see http://www-128.ibm.com/developerworks/library/ws-bpel/

[17] Mandell, D. J. and McIlraith, S. A. (2003): *Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation*,