Using Semantic Web Services to Integrate Data and Processes from Different Web Portals

Marília T. de Mello[1]
Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre - RS – Brazil
mtmello@inf.ufrgs.br

Mara Abel[2]
Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Porto Alegre - RS – Brazil
marabel@inf.ufrgs.br

Francisco García-Sánchez[3]
Departamento de Ingeniería de la Información y las Comunicaciones – Universidad de Murcia
Murcia - Spain
frgarcia@um.es

Abstract

Integration and exchange of information across the Internet is a universally recognized need, in a wide variety of domains. New promising application fields such as Semantic Web Services can improve the integration process in the Web, dealing with the major difficulty of the semantic heterogeneity in the resources, by means of domain ontologies. In this research, we propose an integration system, combining domain ontologies and Semantic Web Services, to provide an integrated access to the information provided by different Web portals in two application domains: biomedicine and geology. Semantic Web Services were designed to encapsulate Web portals in both domains, allowing the system to accomplish the integration in each domain, separately.

1. Introduction

Strong efforts are being applied by the Artificial Intelligence community dedicated to explore potential approaches to improve the process of data discovery and integration. That happens mainly because Web users demand integrated access to all information resources available through the Internet. New promising application fields such as the Semantic Web and Semantic Web Services are being recognized as the adequate support for integrating data sources and services in the Web.

Nowadays, it is possible to find independent and heterogeneous data resources, accessible through the Internet, covering information related to any kind of domain. Information resources include not only data on the Web but also Web portals which offer services to the users, such as search engines, providing integrated access to dynamic content from a variety of sources. However, only a few Web portals offer their information in the form of Web Services (such as the portal of Amazon.com). The power of the Web Service technology lies in the fact that it establishes a common platform for integrating distributed computing applications, in intranets as well as the Internet at large [4].

The major difficulty when integrating different data sources is the semantic heterogeneity of information sources. Besides being widely distributed across de Internet, and being found in a variety of storage formats, different information sources can use the same object to represent different concepts, or different objects to represent the same concept. In order to solve this problem, applications make use of domain ontologies, which provide a common model of the concepts that are relevant and their allowable relations. The idea of this project is to go beyond this scope and use Semantic Web Services, which make use of ontologies.

The aim of Semantic Web Services is to provide solutions to the challenges associated with automated discovery, dynamic composition, enactment, and other tasks associated with managing and using service-based systems. Several researches explore the use of Web Services for information integration, like [17, 19]. However, only a few studies make use of Semantic Web Services. One of this is the Project DIP[1] (Data, Information, and Process Integration with Semantic Web Services), which intends to provide an environment in which different Web Services can cooperate with each other and be found out automatically.

Moreover, according to [2], current integration proposals do not allow users to take advantage of the services offered by Web portals. This happens because traditional integration techniques are focused just on data integration, ignoring the services provided by the portals.

The present research intends to achieve the integration of data and processes (search engine services), which are offered through different Web portals. The portals offer a

[1] http://dip.semanticweb.org/

variety of functionalities and also permit accessing many data related to a specific domain. The approach is to encapsulate these data sources through Web Services that are semantically annotated based on a global domain ontology, which is specific to each application domain.

A system based on Semantic Web Services and on global domain ontologies was designed to encapsulate the offer of web site services and to allow an integrated access to that information. Once the Web Services are developed and semantically annotated, the system is able to access their functionalities in an automatic way. The main goal is to offer an integrated access to the different functionalities provided by Web portals in specific domains.

The remainder of this paper is organized as follows. An overview of Web Services, the Semantic Web, and Semantic Web Services is given in Section 2. Section 3 presents the proposal of the integration system, giving an overview of the complete project. Section 4 describes the case study realized, using two application domains: biomedicine and geology. The design of a general integration application based on Semantic Web Services is described in section 5. Conclusions are pointed out in section 6.

2. Semantic Web Services Revision

The Semantic Web is a promising evolution of today's Web, providing ways to make the automation process effective and easy. One of its fundamental components will be the markup of Web Services to make them computer-interpretable, use-apparent, and agent-ready [13]. In this section, the main concepts related to Semantic Web Services are presented. First, Web Services are defined, as well as its goals. Then, a general view of the Semantic Web is given. Thirdly, the concept of Semantic Web Services is introduced. Finally, a special attention is given to OWL-S, an approach to the development of Semantic Web Service frameworks.

2.1. Web Services

Web Services are well-defined and reusable software components that perform specific encapsulated tasks via standardized Web-oriented mechanisms [8]. They can be discovered, invoked, and the composition of several services can be choreographed, using well defined workflow modeling frameworks.

Web Services were intended to bring a new level of integration to the computing industry and its networked communities [9]. Service-based applications should be able to interoperate despite being developed in different programming languages, at different times, by different people, with designs based on different assumptions. They provide a standard means of interoperating between different software applications, running on a variety of platforms [3].

Currently, a Web Service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML [3]. A Web Service can be accessed via the Internet, through its exposed interface, where its definition and functions can be found out by other software systems, in a controlled manner [4]. These systems may then interact with the Web Service using XML-based messages over Internet protocols. Basically the Web Service interface is presented in WSDL (Web Service Description Language), which describes how other applications can interact with that Web Service, through SOAP (Simple Object Access Protocol) messages. Those messages are mostly transported using http, with XML format, together with a set of other Web conventions.

A Web Service interaction involves two or more software agents (clients and providers) exchanging information in the form of messages. First, the provider registers the service it is offering, using WSDL. The registry is based on the UDDI specification (Universal Description Discovery and Integration), which defines data structures and APIs for publishing (describing) and discovering Web Services [4]. Once the service is registered, a client can query the registry to search for the services it is interested in. After receiving a response, the client may invoke the service it wants, using SOAP.

WSDL is the W3C recommended language for describing the service interface, in an XML notation. It enables specification of data types, operation signatures, port types, message format and transport protocol details, network addresses of different ports, and grouping of different ports into a Web Service [16]. However, WSDL does not enable specification of various constraints on operations and ports in a Web Service. The operations are defined in terms of input and output messages.

Currently, there are only few specifications available for Web Services that are able to provide some kind of formal definition to what the syntactic descriptions of their functionalities might mean. Besides OWL-S [12] used in the present research, another important specification is SAWSDL [10], which allows description of additional semantics of WSDL components. The lack of machine readable semantics requires human intervention for automated service discovery and composition within open systems, thus hampering their usage in complex business contexts [8].

2.2. Semantic Web

The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [18]. It is a collaborative effort led by W3C with participation from a large number of researches and industrial partners. It is based on the Resource Description Framework (RDF).

The Semantic Web is an extension of the current Web where information has well-defined meaning [18]. The Semantic Web project intends to create a universal way of exchanging information, providing knowledge to the contents of Web documents so that it can be understood by computers. Its goal is to transform the Web into a medium through which data can be shared, understood, and processed by automated tools.

A large proportion of today's data on the Web are "understandable" only to humans or custom-developed applications. The Semantic Web will bring 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 [6]. In order to make it possible, there is the need to enrich the Web with ontologies, which capture the domain knowledge.

Technologies from the Semantic Web can make crucial contributions to Web Service frameworks [9]. Semantic Web Services take up on this idea, introducing ontologies to describe, on the one hand, the concepts in the services domains, and, on the other hand, characteristics of the services themselves and their relationships to the domain ontologies. These semantically rich descriptions enable automated machine reasoning over service and domain descriptions, thus supporting automation of service discovery, composition, and execution, besides reducing manual configuration and programming efforts.

2.3. Semantic Web Services

At present, the use of Web Services requires human involvement (information has to be browsed and forms need to be filled in). The existing technologies for Web Services only provide descriptions at the syntactic level, making it difficult for requesters and providers to interpret or represent nontrivial statements such as the meaning of inputs and outputs or applicable constraints [8].

The Semantic Web vision, as applied to Web Services, aims at automating the discovery, invocation, composition and monitoring of Web Services by providing machine-interpretable descriptions of services [5]. A prerequisite to this, however, is the emergence and evolution of the Semantic Web, which provides the infrastructure for the semantic interoperability of Web Services.

Semantic annotations are the base for the automatic processing of Web pages. In order to achieve automation, Web Services will be augmented with rich formal descriptions of their capabilities, such that they can be utilized by applications or other services without human assistance or highly constrained agreements on interfaces and protocols [8]. The semantic description of a Web Service should provide specific information such as: data and metadata associated with a service together with specifications of its properties and capabilities; the interface for its execution; and the prerequisites and consequences of its use [13].

A Semantic Web Service is defined through a service ontology, which enables machine interpretability of its capabilities as well as integration with a domain knowledge. The service ontology aggregates all concept models related to the description of a Semantic Web Service, and constitutes the Knowledge-level of the information describing and supporting the usage of the service [8].

According to [8], three main approaches have been driving the development of Semantic Web Service frameworks: IRS-II, WSMF and OWL-S. IRS-II (Internet Reasoning Service) is a knowledge-based approach to SWS, which evolved from research on reusable knowledge components. WSMF (Web Service Modeling Framework) is a business-oriented approach to fully enable e-commerce by applying Semantic Web technology to Web Services. OWL-S, in its turn, is an agent-oriented approach, providing fundamentally an ontology for describing Web Service capabilities. The following section describes the OWL-S approach in more detail.

2.4. The OWL-S approach

OWL-S is an OWL ontology with three interrelated subontologies, known as the service profile, service model (process model), and service grounding [12], as shown in Figure 1. A service profile is the description of the offerings and requirements of a service. A service model describes how a service works. The service grounding specifies details of how an agent can access a service. It will specify a communication protocol and port numbers to be used while contacting a service.

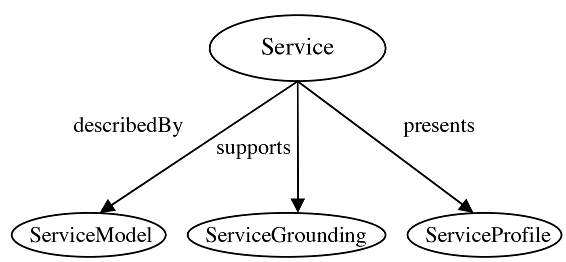

Figure 1. Service ontology

OWL-S provides the major Semantic Web Service description language for specifying the functions (preconditions and effects) of an operation and semantic types for each of the inputs and outputs of the service. It offers to extend the capability of OWL to form the required ontology for Web Services, making them machine understandable while supporting automated Web Service composition and interoperability.

A service profile in OWL-S is described through the properties: input, output, precondition and effect. It provides the following information: a human-readable description of the service and its provider; a specification of the functionalities provided by the service; and additional information, such as expected response time and geographic constraints [5].

Service models are based on the key concept of a process, which describes a service in terms of inputs, outputs, preconditions, effects, and its composition of component subprocesses. A powerful feature of OWL-S is the ability to model composite processes [9]. A composite process is constructed from subprocesses. The control flow of a composite process is defined using control constructs, such as If-Then-Else, Sequence, and Repeat-Until.

The (OWL-S/WSDL) grounding uses OWL classes as the abstract types of message parts declared in WSDL. The central function of the OWL-S grounding is to show how the inputs and outputs of an atomic process are realized as messages in transmittable format [3]. An OWL-S atomic process corresponds to a WSDL operation. The OWL-S description relates to a WSDL file through the grounding.

Since the OWL-S service ontology is public and does not prescribe a framework implementation, it has been used as the starting point of individual efforts towards Semantic Web Services [8].

3. Proposal

Whenever users need to look up information related to a given domain, they have to access one or more Web portals, making queries in each one, until they find the answer they are looking for. However, people want to save time, improving the quality of their work. Therefore, an integrated access to the portals of interest would facilitate the user's task, providing them with a unique tool, where they can access all the available definitions with just one query.

In this work, we propose the development of a domain independent integration system to provide users with an integrated access to data and processes, offered by Web portals that may attend their goals. Two application domains will be used as case studies in this project: the biomedicine and the geological domain, each one providing its own domain ontology. The domain ontologies will help users expressing their needs and will provide information integration by means of Semantic Web Services.

The biomedical domain is supported by the 'Oncogen' ontology [14], which is an ontology of biological terminology for genetic oncology conceived, as part of the project, to support integrated consultation over data and information contained in the biological, medical and information areas. The knowledge model in the geological field is represented by the existing ontology of the PetroGrapher project [1], which was refined in order to become more specific to the present application. This ontology represents the vocabulary applied by geologists in order to make the description of rock samples during the analysis for petroleum reservoir evaluation.

The main goal of this project is having a system, based on global domain ontologies and on Semantic Web Services, that encapsulates the offer of different Web portal services and allows an integrated access to the functionalities provided by them. We apply this system to both domains in order to evaluate the capability of information integration in a domain-independent way. In order to provide those functionalities, a complete system architecture was designed, making use of all the techniques described in the previous section. The integration system comprises all the components illustrated in Figure 2.

The *user interface* allows users to express their query using an ontology guided tool (based on the global domain ontology), which assists them in expressing their goals. The domain ontology is loaded through the Protégé-OWL API, and its main concepts are used to form a simple menu where the user can choose the type (class) of the object they are looking for.

Through the *query component*, the system searches and selects the most appropriate Web Services by accessing their semantic description. This is done by matching the term from the domain ontology with the semantic description of the services capabilities. The matching is done by using the criteria proposed by Paolucci et al. [15], but here only exact matches are considered (where both terms denote exact the same concept or the term from the request is a subclass of the concept specified in the service being offered). The query component returns a list with the services that attend the user's need.

The *invoker*, in its turn, uses the grounding of the Web Services, which is part of the semantic description, to invoke each service selected by the query component. It returns the result of each service execution to the core component.

The main component is the *core* of the system, which acts as an intermediate application among the other components. Once the results are returned by the invoker, they are integrated and given to the user by the core of the system.

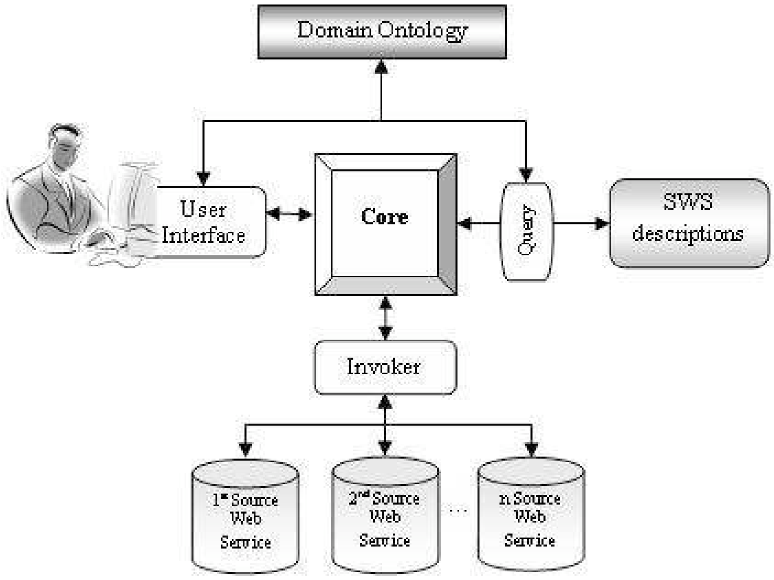

Figure 2. Integration system architecture

4. Case Study

As the present study aims at providing a global domain independent solution to the problem of Web portals integration, the biomedicine and the geological domains will be used as case studies, each one providing its own domain ontology.

The development of the complete integration system starts with the design of Semantic Web Services to encapsulate different Web portals in each domain. The idea is to select a few web portals from the ones that provide information about genes, proteins, and their relations, in the biomedicine domain; and about minerals and rocks, in the geological domain. Next step is to design a Web Service to each web portal, offering access to its information. However, to make use of a Web Service, a software agent needs a computer-interpretable description of the service and the means to access it. In order to make it possible, the developed Web Services must be semantically described, based on a domain ontology, so that the automation can be achieved.

4.1. Data Sources

People who work in research fields, such as biomedicine and geology, need the source of information to be available whenever they need a specific definition. Therefore, we analyzed Web portals in both domains, choosing the ones that provide search mechanisms or glossaries with definitions and descriptions of interest. From the Web portals that were analyzed, four were used as data sources for biomedicine and two were used for geology. The selected sources are described as follows.

The *Gene Ontology*[2] project provides a controlled vocabulary to describe gene and gene product attributes in any organism. The *PubMed*[3] database, available via the NCBI Entrez information retrieval system, is a free search engine developed by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine (NLM). *MPact*[4] - the Representation of Interaction Data at MIPS (The Munich Information Center for Protein Sequences) provides a common access point to interaction resources at MIPS [11], which provides resources related to genome information. MPact provides the user with intuitive query forms to quickly retrieve the interactions of interest. The *Swiss-Prot*[5] Protein Knowledgebase is a curated protein sequence database which strives to provide a high level of annotation, a minimal level of redundancy and high level of integration with other databases.

Schlumberger Oilfield Glossary[6] is an instant reference that offers accurate definitions for major oilfield activities, reviewed by technical experts. The *Geologynet*[7], through two of its online geology databases: the Mineral and the Rock databases. They have a search interface providing access to a collection of rock and

[2] http://www.geneontology.org/
[3] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed
[4] http://mips.gsf.de/genre/proj/mpact/
[5] http://www.expasy.ch/cgi-bin/sprot-search-ful
[6] http://www.glossary.oilfield.slb.com/default.cfm
[7] http://geologynet.com/indexa.htm

mineral databases that can be used for education and reference.

4.2. Domain Ontology

A domain ontology is a type of formal specification of a shared conceptualization [7] conceived to capture knowledge of one particular domain, providing detailed description of the domain concepts and their relations. The use of ontologies enables the sharing of common concepts, the specialization of these concepts and vocabulary for reuse across multiple applications, the mapping of concepts between different ontologies, and the composition of new concepts from multiple ontologies [13].

The ontology used in the biomedicine domain, called Oncogen, is a domain ontology for genetic oncology – genes implied in the development of cancer. It is an ontology of biological terminology, which provides a model of biological concepts that can be used to form a semantic framework for data storage, retrieval and analysis tasks. Oncogen was implemented in OWL, through the Protégé ontology editor.

The ontology applied in the geological domain is a refinement of an existing domain ontology, which is part of the PetroGrapher project. It is modeled as a partonomy of terms and was translated to OWL, using the Protégé ontology editor. This ontology represents the vocabulary in the sedimentary petrography area.

The domain ontology provides the best way of helping communities in agreeing on a standard definition of terminology, encoding this definition, then solving the data heterogeneity problem. Each input and output from the operations provided by the portals must be validated against a domain ontology type. This way, the system not only recognizes the parameters as strings, integers, or any other basic type, but also as concepts related to the ontology classes, having an associated meaning. The domain ontology will help users express their needs and achieve information integration by means of Semantic Web Services.

4.3. Semantic Web Services Design

A Web Service must be developed to each data source in order to expose the functionalities provided by the selected Web portals. The implemented services are simple Web Services because they do not rely upon other Web Services nor require further interaction with the user.

Each Web Service is designed as a common class in an object oriented programming language. Then, a WSDL file is generated from the designed class, allowing the specification of the syntax of input and output messages, as well as other necessary details to the service invocation [12]. The service that provides global access to the PubMed database (for example) consists of two methods (or operations): getProtein and getGene, which return the description of a protein and a gene, respectively, given its name. An extract of the WSDL file that describes the capabilities of the PubMed service is shown in Figure 3.

```
<wsdl:definitions targetNamespace="http://klt.inf.um.es/">
  <wsdl:documentation> PubMed Service </wsdl:documentation>
+ <wsdl:types> </wsdl:types>
+ <wsdl:message name="getProteinMessage"> </wsdl:message>
+ <wsdl:message name="getProteinResponse"> </wsdl:message>
+ <wsdl:message name="getGeneMessage"> </wsdl:message>
+ <wsdl:message name="getGeneResponse"> </wsdl:message>
+ <wsdl:portType name="PubMed_WebServicePortType"> </wsdl:portType>
– <wsdl:binding name="PubMed_WebServiceSOAP11Binding" type="axis2:PubMed_WebServicePortType">
     <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
   + <wsdl:operation name="getProtein"> </wsdl:operation>
   + <wsdl:operation name="getGene"> </wsdl:operation>
   </wsdl:binding>
+ <wsdl:binding name="PubMed_WebServiceSOAP12Binding" type="axis2:PubMed_WebServicePortType">
  </wsdl:binding>
+ <wsdl:binding name="PubMed_WebServiceHttpBinding" type="axis2:PubMed_WebServicePortType">
  </wsdl:binding>
+ <wsdl:service name="PubMed_WebService"> </wsdl:service>
</wsdl:definitions>
```

Figure 3. PubMed service WSDL file

However, the WSDL description by itself does not provide enough information to support the decision of which service better suits the client necessities. No software system can read and utilize the WSDL interface without human assistance, because the WSDL specification language provide no means of including representations of the semantics of the defined operations and associated messages elements [12].

The WSDL's lack of semantic description associated to the meaning of inputs and outputs makes it impossible to develop software clients that can, without human assistance, dynamically find and successfully invoke a service [12]. Therefore, after developing the Web Services, their semantic description must be done to make it possible to access them in an automatic way.

The semantic annotation was done in a semi-automated way, through the OWL-S language (based on the domain ontology, which is written in OWL) using the OWL-S Editor – as a Protégé plug-in. This tool offers a development environment where the domain ontologies are well integrated with the service descriptions.

The OWL-S Editor allows the generation of a "skeletal" OWL-S description based on a preexisting WSDL file. Using this feature, parts of the OWL-S description were generated automatically, based on the inputs and outputs defined in the WSDL file that was generated from the service class. Afterwards, this simple description was enriched by the semantic part, which is the association of the parameters with the domain ontology classes, together with the definition of preconditions and effects. A piece of the OWL-S semantic description of the PubMed service is presented in Figure 4.

The objective of Semantic Web Services is to support clients that can find and correctly utilize newly discovered service without additional programming. The semantic description allows the automatic access to the services by software agents.

```
<j.1:Service rdf:ID="getGeneService">
  <j.1:supports rdf:resource="#getGeneGrounding"/>
– <j.1:presents>
  – <j.0:Profile rdf:ID="getGeneProfile">
      <j.0:serviceName
        rdf:datatype="http://www.w3.org/2001/XMLSchema#string">getGene</j.0:serviceName>
      <j.0:textDescription    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Get gene description
        from a given gene name or ID</j.0:textDescription>
    – <j.0:hasOutput>
      + <j.2:Output rdf:ID="description"> </j.2:Output>
      </j.0:hasOutput>
    – <j.0:hasInput>
        + <j.2:Input rdf:ID="term"> </j.2:Input>
      </j.0:hasInput>
      <j.1:presentedBy rdf:resource="#getGeneService"/>
    </j.0:Profile>
  </j.1:presents>
–<j.1:describedBy>
 + <j.2:AtomicProcess rdf:ID="getGeneProcess"> </j.2:AtomicProcess>
  </j.1:describedBy>
</j.1:Service>
```

Figure 4. A portion of the OWL-S file for the PubMed service showing the operation 'GetGene' and its parameters

4.4. Results

Having the repositories with semantic descriptions related to both domains, and all the developed components of the application integrated, the system is be able to perform the integration of the sources, based on those repositories, in each domain separately. The answers to the queries are validated against the information in the Web portals. Besides returning the expected results, it is important to notice that only the services of interest are invoked. That is, when a user makes a query, the system does not invoke and execute all the services in the domain to return the expected answer. Instead, it queries the semantic descriptions, building a list with the services that must be invoked, which are only the ones that provide the answer to the user query.

5. Development process of an integration application based on Semantic Web Services

The development process of an integration application based on Semantic Web Services must begin with the collection and formalization of the essential information that should be integrated in the given domain. Based on that analysis, the data design is done in two tasks: Web portals analysis and selection, and the domain ontology specification (or the reuse of an existing one). Next step is to encapsulate each selected Web portal with a Web Service, followed by the semantic description of the designed Web Services, based on the domain ontology.

The tasks described above are specific to each application domain, which means that they must be performed every time a new domain is chosen. After having the semantic repositories, the components of the integration system must be developed and integrated.

The system development comprises the architecture design, the implementation, besides the testing and evaluation of the application as a whole. The complete system must encapsulate all the service offers from the different Web portals, allowing an integrated access to that information, and answering the user's query.

6. Conclusions

Web Services, as well as Semantic Web Services, are applications where Artificial Intelligence techniques can be used effectively [5]. Semantic Web Services is a new research paradigm that, by making use of domain ontologies, establishes a common platform for integrating distributed computing applications. Moreover, they provide means to automate service discovery, composition, invocation, and monitoring in the Web environment.

The use of domain ontologies, through the OWL-S tool, made it possible to give richer meaning to inputs and outputs through class hierarchies, properties and property restrictions. The chosen Web portals in each domain could be encapsulated by Web Services, which, in their turn, could be semantically annotated using OWL-S, providing a consistent basis to automate the integration process.

The major contribution of the present work is the provision of an integration system architecture with basic main components, which could be used in more than one application domain. The system was able to provide integration both in the biomedical and in the geological domain. The central components in the system perform the integration according to the application domain specified and linked to the system by the domain ontology, and the semantic descriptions.

As future work, we plan to search and select a few more web portals to improve the testing set. Besides that, we plan on extending the case study, adding more application domains, in order to turn the system into a domain independent solution to the problem of integrating web portals functionalities through Semantic Web Services.

7. References

[1] Abel, M., L.A.L. Silva, et al. "PetroGrapher: Managing petrographic data and knowledge using an intelligent database application", In *Expert Systems with Applications* 26(1 SPECISS), 2004, pp. 9-18.

[2] Acuña, C.J. et al. "Toward Web Portals Integration through Semantic Web Services", In *International Conference on Next Generation Web Services Practices – NWeSP'05*, 22-26 Aug, 2005, 6 pp.

[3] Alesso, H.P., and C.F. Smith, *Developing Semantic Web Services*, A K Peters Ltd, Canada, 2005.

[4] Alonso, G. et al. *Web Services: Concepts, Architectures and Applications*, Springer Verlag, 2004.

[5] Antoniou, G., and F. Van Harmelen, *A Semantic Web Primer*, The MIT Press. London, England, 2004.

[6] Berners-Lee, T., J. Hendler, and O. Lassila, "The Semantic Web", In *Scientific American*, 2001.

[7] Borst, W.N. *Construction of engineering ontologies*, University of Twente, PhD Thesis, 1997.

[8] Cabral, L. et al. "Approaches to Semantic Web Services: an Overview and Comparisons", In *The Semantic Web: Research and Applications - ESWS 2004*, Lecture Notes in Computer Science, 3053, Springer-Verlag, 2004, pp. 225-239.

[9] Elenius, D. et al. "The OWL-S Editor – A Development Tool for Semantic Web Services", In *Proceedings of the Second European Semantic Web Conference*, 2005.

[10] Farrell, J., and H. Lausen. "Semantic Annotations for WSDL and XML Schema", W3C Recommendation, 2007, http://www.w3.org/TR/sawsdl.

[11] Güldener, U. et al. "MPact: the MIPS protein interaction resource on yeast", Nucl: Acids Res, 2006, pp. D436-D441.

[12] Martin, A. et al. "Bringing Semantics to Web Services: The OWL-S Approach", In *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, 2004.

[13] McIlraith, S., T.C. Son, and H. Zeng, "Semantic Web Services", In *IEEE Intelligent Systems,* 2001, 16(2), pp. 46-53.

[14] Muñoz-Garcia, A. et al. *Oncogen Ontology,* KLT Report, Universidad de Murcia, Spain, 2006.

[15] Paolucci, M. et al. "Semantic Matching of Web Services Capabilities", In *Proceedings of the First International Semantic Web Conference, LNCS 2342, Springer-Verlag, 2002, pp. 333-347*.

[16] Tosic, V., K. Patel, and B. Pagurek, "WSOL – Web Service Offerings Language", In *CAiSE 2002 - WES 2002*. Lecture Notes in Computer Science, 2512, Springer-Verlag, 2002, pp. 57-67.

[17] Vinoski, S. "Integration with Web Services", In *Internet Computing*. IEEE, 2003, pp. 75-77.

[18] W3C "Semantic Web", 2003. http://www.w3.org/2001/sw.

[19] Zhu, F. et al. "Dynamic Data Integration Using Web Services", In *Proceedings of International Conference on Web Services*. IEEE, 6-9 July, 2004, pp. 262-269.