Open Access

Open Journal of Semantic Web (OJSW)
Volume 1, Issue 1, 2014

www.ronpub.com/ojsw
ISSN 2199-336X

BioSStore: A Client Interface for a Repository of Semantically Annotated Bioinformatics Web Services

Ismael Navas-Delgado, José F. Aldana-Montes

Department of Computer Languages and Computing Science,
Higher Technical School of Computer Science Engineering, University of Málaga,
Málaga, 29071, Spain, {ismael, jfam}@lcc.uma.es

ABSTRACT

Bioinformatics has shown itself to be a domain in which Web services are being used extensively. In this domain, simple but real services are being developed. Thus, there are huge repositories of real services available (for example BioMOBY main repository includes more than 1500 services). Besides, bioinformatics repositories usually have active communities using and working on improvements. However, these kinds of repositories do not exploit the full potential of Web services (and SOA, Service Oriented Applications, in general). On the other hand, sophisticated technologies have been proposed to improve SOA, including the annotation on Web services to explicitly describe them. However, these approaches are lacking in repositories with real services. In the work presented here, we address the drawbacks present in bioinformatics services and try to improve the current semantic model by introducing the use of the W3C standard Semantic Annotations for WSDL and XML Schema (SAWSDL) and related proposals (WSMO Lite). This paper focuses on a user interface that takes advantage of a repository of semantically annotated bioinformatics Web services. In this way, we exploit semantics for the discovery of Web services, showing how the use of semantics will improve the user searches. The BioSStore is available at http://biosstore.khaos.uma.es. This portal will contain also future developments of this proposal.

TYPE OF PAPER AND KEYWORDS

Research paper: Bioinformatics, semantic web services, semantic annotation, service discovery.

1 INTRODUCTION

Web services have emerged as a key technology in the development of distributed applications. This technology, based on a set of standards recommended by the W3C, enables applications to communicate and exchange data over the Internet. Web services benefit from object oriented programming techniques because they allow developers to build applications from existing software components. This feature facilitates integration at business level which explains why large companies in the ICT (Information and Communication Technologies) sector are involved in its development. Life sciences is a domain in which Web services have been widely adopted, and numerous approaches are making use of them to provide data retrieval and analysis tasks.

Parallel to the increasing number of Web services available (in general but specifically in bioinformatics), the Semantic Web has also emerged. The idea is that

machines are not only able to present information, but also to understand it. However the Semantic Web is not an entirely new Web, rather, it is an extension of the existing one, within which information has a well-defined meaning thereby enabling computers and people to work in cooperation. In this regard, the first approaches were based on the description of the Web services using external definitions by means of ontologies (WSMO [1] and OWL-S [2]). However, WSMO and OWL-S are heavy proposals that are not needed in some cases. So, a new proposal has been launched in the W3C for lightweight semantic Web services: Semantic Annotations for WSDL and XML Schema (SAWSDL) [3]. It defines how to add semantic annotations to various parts of a WSDL document such as input and output message structures, interfaces and operations. This extension has attributes that fit within the WSDL 1.1 [4], WSDL 2.0 [5] and XML Schema [6] extensibility frameworks. SAWSDL defines an annotation mechanism for specifying the data mapping of XML Schema types to and from an ontology. To accomplish semantic annotation, SAWSDL defines extension attributes that can be applied both to WSDL elements and to XML Schema elements.

In bioinformatics, BioMoby [7] provided an alternative to the Semantic Web technologies at a time when they were unconsolidated, and proposals for annotating Web services did not exist at all. This was the first step towards a way of annotating bioinformatics tools provided as Web services, and a way of registering said Web services in order to make them locatable for users (mainly software developers). Thus, the extensive use of Web services to provide bioinformatics tools at the same time encountered a way of registering these tools to enable their use in complex tools and workflows. However, the emergence of Semantic Web technologies has not been widely taken up by the bioinformatics community for the annotation of these Web services.

So, the sustained growth of available services in Life Sciences has led to an explosion of bioinformatics Web Service registries. In this paper we propose a novel registry in which the consolidated Semantic Web technologies (OWL, RDF and SPARQL) are combined with the emergent proposal for the annotation of Web services (SAWSDL), taking advantage of the reasoning provided by OWL reasoners.

The rest of this paper is structured as follows. Section 2 shows the proposed repository, and how semantics is used to improve Web Service discovery. Section 3 classifies the most prominent registries in relation with their use of the semantics. Finally, Section 4 presents conclusions and future work.

2 SEMANTIC WEB SERVICE REGISTRY

The efforts made to produce standards for the semantic annotation of Web services have guided a lot of the research dealing with the problem of Web services discovery and composition using semantic technologies. However, these studies lack real life large scale scenarios to test their algorithms. In our approach we aim to combine the efforts in semantic Web technologies with the huge amount of syntactic (and real) Web services available in bioinformatics. The idea is to enable the semantic annotation of biological Web services (existing and new). This approach will allow sophisticated algorithms to facilitate the service discovery and composition. However, existing tools working with Web services will keep working as the original Web services will not change at the lower level. This Section will show how the registry has been built (Figure 1), populated with an initial set of Web services and how it can be used through its user interface. The general steps (Figure 1) to perform the service annotation process are divided into four main tasks:

1. *Information retrieval*: the targeted repository is analysed and accessed in order to locate all the information about services (WSDL files, metadata, etc.). The access method will depend on the repository implementation, but available APIs should be used if possible. In this paper we show how a repository of BioMOBY services is accessed to retrieve information about biological services.
2. *Adaptation phase*: This stage includes the transformation of WSDL descriptions to the WSDL 2.0 specification. Available metadata is also analysed and transformed into a common format following an entity-relationship model.
3. *Mapping generation*: if metadata exists, it will be used to automate the annotation process. Otherwise, the correspondences between the syntactic and the semantic level will have to be done by hand. However, regardless the specific mechanism to generate the mappings, this phase must follow a quality-effort trade-off as the accuracy of the results relies on it. This means that even when metadata is available a data curation could be useful here to improve the quality of service annotations.
4. *RDF transformation*: This stage includes the generation of RDF from the annotations created in the previous phase. In this way, enriched service descriptions will be used through a SPARQL API for discovery and composition purposes.

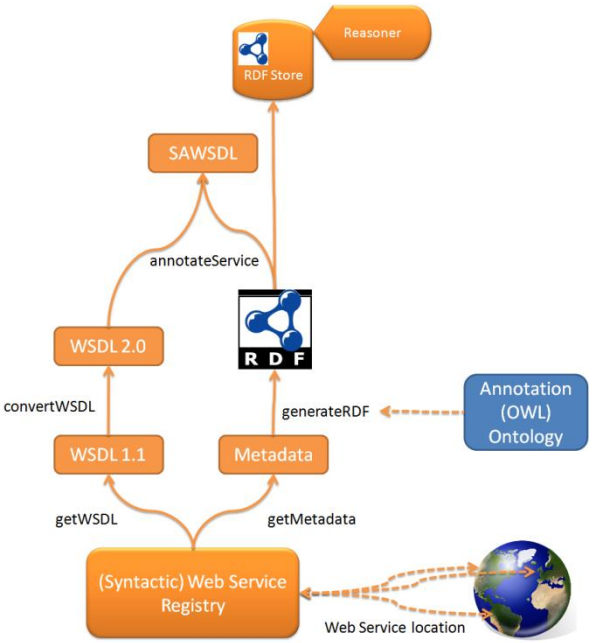

Figure 1: Service Discovery, Annotation and Use

2.1 BioSStore Service Discovery and Annotation

The first part of the process as indicated in Figure 1 is to retrieve services, to initiate the registry. In this case, we have started the work from BioMOBY Web services. However, we are working on adding new services and enabling user registration of Web services.

BioMoby [7] is a project with the goal of producing an open-source, simple, extensible platform to enable the discovery, representation, integration, and retrieval of biological data from widely disparate data hosts and analysis services. The most useful characteristic of BioMOBY is that there are several BioMOBY central repositories providing a set of more than 1500 real services that can be used to test discovery and composition algorithms.

The MOBY-S system defines a protocol for the communication between clients and a server to enable the discovery of biological services. The main goal is to provide a standard way of discovering and using biological services. In this sense, BioMoby-compliant Web services are registered in instances of Moby Central. Thus, client interfaces and applications can use a defined API to ask for the list of services and the annotations available. BioMOBY services are annotated in two different dimensions:

- *Service Type*: this is a hierarchy of service types, so each service is classified depending on its type. Thus, users can discover services using this hierarchy.
- *Object Type*: The inputs and outputs of the services are annotated with respect to a hierarchy of data types. Thus, it is possible to know which, in two services, can be connected by these data types.

BioMOBY provides a set of simple services described using metadata, and the goal in this step is to retrieve the WSDL description files of these services and their corresponding metadata. Using the BioMOBY API we have accessed the set of 1500+ services, and we have downloaded their WSDL descriptions and matched each service with its BioMOBY metadata. This process was automated and consequently the update process of the system is as simple as repeating the access, and downloading the new, available information.

In this case, we can extract metadata from the Web services, which will be used to facilitate the annotation process. In the cases where metadata is not available semi-automatic processes can be applied to try to discover the semantic annotations.

However, the WSDL 1.1 descriptions obtained for each service registered in a BioMOBY Central include information about the input and output XML data types. These descriptions are not enough because they do not include semantics of the data. For this reason, although two services can be compatible due to the syntactic data types, they may be incompatible on a conceptual level.

Therefore, the tools have to retrieve all the WSDL files for these services, and transform them to WSDL 2.0. This is automatically performed, producing a set of services for which the two WSDL files are stored. This conversion enables the use of SAWSDL to annotate the Web Service descriptions using ontologies.

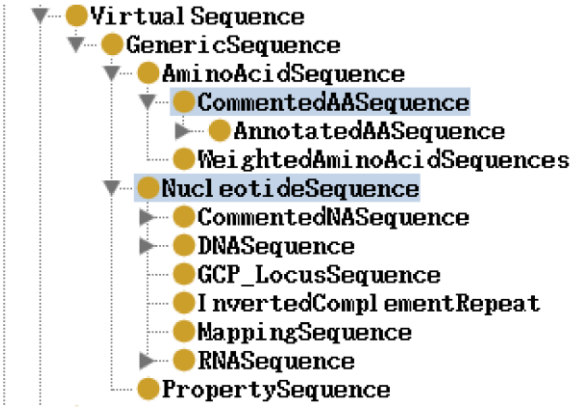

Figure 2: Part of the ontology that shows the concepts used to annotate the input and output types for the runEmbossGetorfFromSequence service.

Later, annotating these services semantically is our aim. For this task, we need an OWL ontology with enough expressivity to allow us to take advantage of reasoning tasks. In order to annotate the services we have developed an annotation ontology, which is divided into two parts: biological concepts of the domain and service related concepts. The biological concepts are an extension of and improvement on the BioMOBY Objects ontology. The BioMOBY concepts have been generalised and properties have been added to improve the quality of service annotations.

Figure 2 shows the hierarchy of concepts in the neighbourhood of the concepts "*NucleotideSequence*" and "*CommentedAASequence*", which will be used in the examples that follow. However, the service related concepts are based on the BioMOBY Services ontology and the WSMO Lite Service Ontology [8]. In this case, services can be annotated using concepts related to the biological domain to annotate inputs and outputs or concepts related to the service type domain to annotate types. Figure 3 shows the concept "*Translating*", which can be used to annotate the functionality of some services. The developed ontology is available at http://biosstore.khaos.uma.es/owlim/ontology/MyBiomobyOnt.owl.

The third step of the process consists of using our ontology and the metadata downloaded to add annotations to the WSDL 2.0 files. In this case this step has been automated since the WSDL 2.0 files have the same elements, e.g., input, output and interface and the BioMOBY metadata can be mapped to our ontology. The result of this step is a set of SAWSDL descriptions that will allow algorithms to take advantage of semantics to discover and compose services.

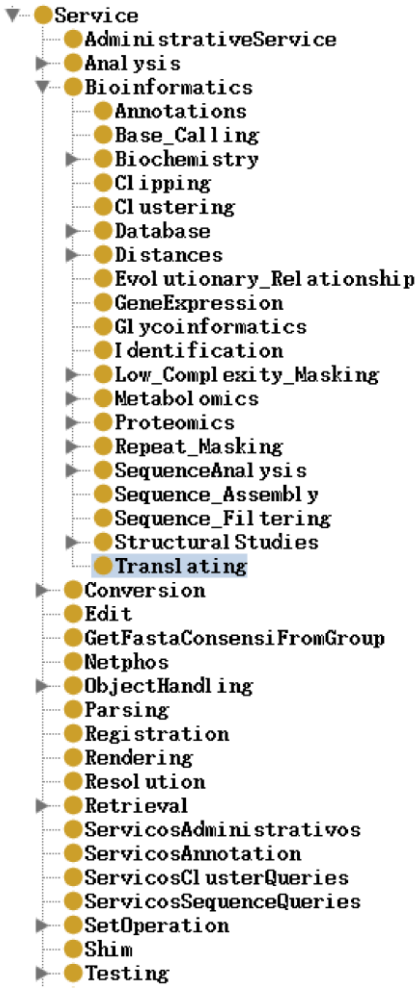

Figure 3: Part of the ontology that shows the concept used to annotate the service interface for the *runEmbossGetorfFromSequence* service.

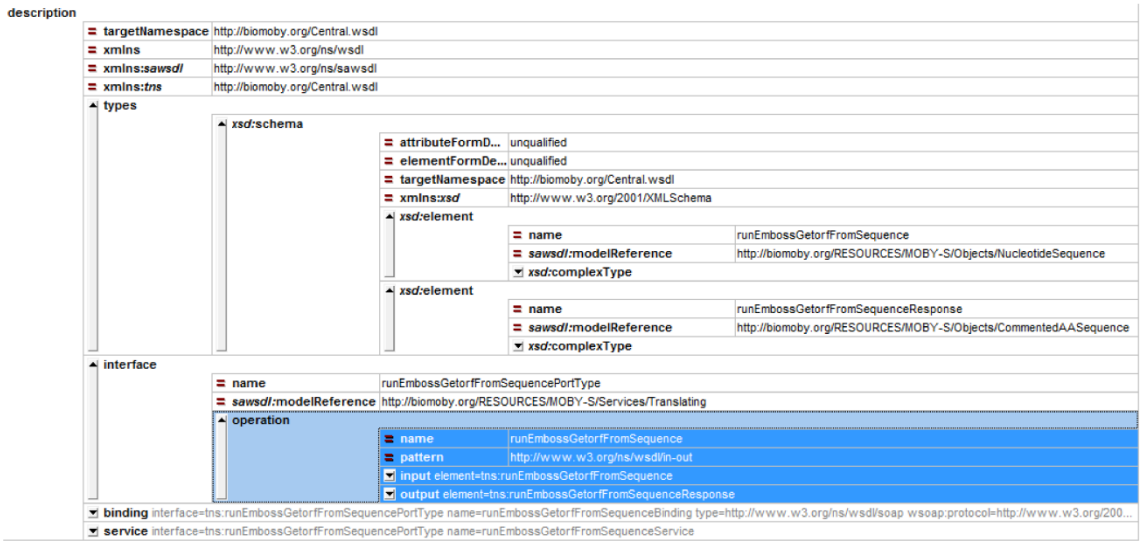

Figure 4: SAWSDL description for *runEmbossGetorfFromSequence* service.

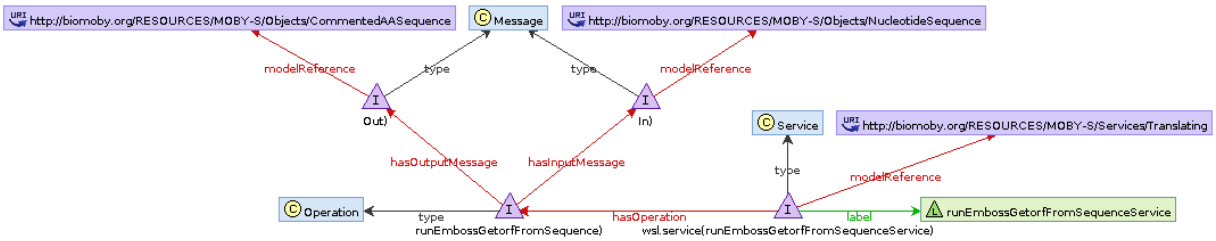

Figure 5: RDF graph for *runEmbossGetorfFromSequence* service.

Figure 4 shows a graphical representation for the SAWSDL file of *runEmbossGetorfFromSequence* service. The "*NucleotideSequence*" concept is used to annotate the input type, the "*CommentedAASequence*" concept is used to annotate the output type, and the "*Translating*" concept is used to annotate the service interface.

2.2 Semantic Store

Once the SAWSDL descriptions have been generated, they are transformed to RDF by means of a tool in order to store them in a RDF store. The RDF corresponding to the SAWSDL description in the example in Figure 4 is shown in Figure 5.

The repository is a RDF database of semantically annotated bioinformatics Web service descriptions. The repository has been built using OWLIM [9] so the repository can be queried using SPARQL, and includes the use of a reasoner to infer results.

This repository processes SPARQL queries for filtering services by the service type, the input data type, the output data type or any combination of these three elements. This ensures the search is more precise, and accurate results are obtained. For example, a user can specify that he/she wants to search services that receive a Nucleotide Sequence, using the concept "*NucleotideSequence*", as input. In this query the system provides the user with 29 services: 11 services that receive "*NucleotideSequence*", 17 that receive "*DNASequence*" and 1 that receives "*MappingSequence*". Note that "*DNASequence*" and "*MappingSequence*" are specialisations of "*NucleotideSequence*" in the ontology. Thus, the user would have obtained only 11 services without the use of a reasoner.

If the user wants to filter the services in more detail he/she can add, for example, the restriction that the service performs a "*Translating*" operation. The corresponding SPARQL query for filtering by input data type and service type is as follows:

```
PREFIX wsmolite:
   <http://www.wsmo.org/ns/wsmo-lite#>
PREFIX sawsdl:
   <http://www.w3.org/ns/sawsdl#>
PREFIX rdfs:
   <http://www.w3.org/2000/01/rdf-schema#>
PREFIX mss:
   <http://biosstore.uma.es/bios2tore/
      annotationOntology>
PREFIX mso:
   <http://biosstore.uma.es/bios2tore/
      annotationOntology>
SELECT ?svc
WHERE {
   ?svc a wsmolite:Service .
   ?svc sawsdl:modelReference ?s.
   ?s rdfs:subClassOf mss:Translating .
   ?svc wsmolite:hasOperation ?o .
   ?o wsmolite:hasInputMessage ?x .
   ?x sawsdl:modelReference ?y.
   ?y rdfs:subClassOf mso:NucleotideSequence.
}
```

In this case the user obtains only two services that fit the search parameters:

- *runEmbossGetorfFromSequence*
- *runEmbossTranseqFromSequence*

2.3 Web User Interface

The repository can be accessed by different clients simply by using discovery methods that allow the call to a set of methods of its API:

- *searchServicesByFunctionalPropertyAND*.
  This method allows us to provide concepts for input, output and service type (as an intersection), and it will return the services annotated with these types or any of their descendants (taking advantage of the reasoner). It corresponds to the advanced search selecting the operator "AND".
- *searchServicesByFunctionalPropertyOR*.
  This method allows us to provide concepts for input, output and service type (as a union), and it will return the services annotated with these types or any of their descendants (taking advantage of the reasoner). It corresponds to the advanced search selecting the operator "OR".
- *getServiceAnnotations*.
  This method searches all the annotations for a given keyword. It corresponds to the generic search in the user interface.

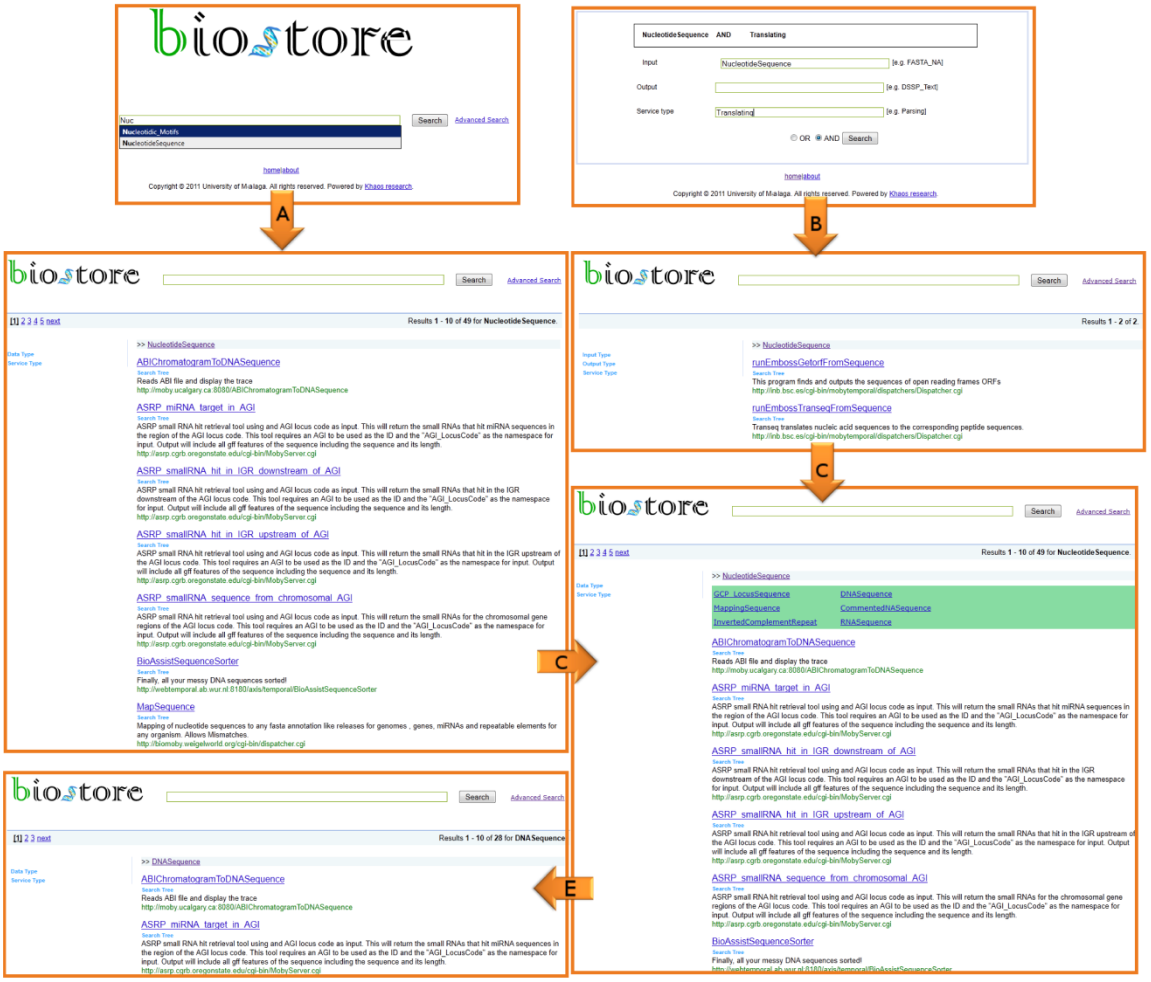

Figure 6: Search Process in BioSStore. The user can perform a general search (in case A, the user is searching for services annotated with NucleotideSequence), or an advanced search (in case B, the user is searching for services with NucleotideSequence as Input and Translating as the Service Type). The result is a list of services (49 for search A, and 2 for search B), that can be reduced by reducing the scope of the search (C). The result of the filtering is a reduced list (in case A it reduces the list from 49 to 28 services).

However, end users will not need to use this programmatic interface, so a Web client (Figure 6) has been developed to provide a powerful and easy to use interface to allow users to locate services.

The user interface helps users to make queries by indicating a set of keywords, which are then matched with the service semantic annotations or the service name. In this sense, the user interface tends to provide an easy to use interface by biologists and bioinformaticians. That is, when a user introduces a word, the user interface generates similar terms of the ontology to be selected by the user (A in Figure 6). For example the user can introduce "*NucleotideSequence*" to locate services that receive or produce a nucleotide sequence, and this will return a list of services that match this search (A search in Figure 6). The search returns a list of services showing, for each one, the service name, description and a graph with the annotations. This graph allows users to see the relationship between the search and the service annotations.

However, the list of results could be quite long, and so to increase usability, BioSStore provides an advanced user interface. The advanced user query interface allows the user to select a set of words to be used to search in the service input, output or service type. Thus, the results will be filtered by the input, output or service type in contrast to the generic search that will look for the search terms in the three elements. This makes the search more precise resulting in more accurate results being obtained. Expanding on the previous example, now

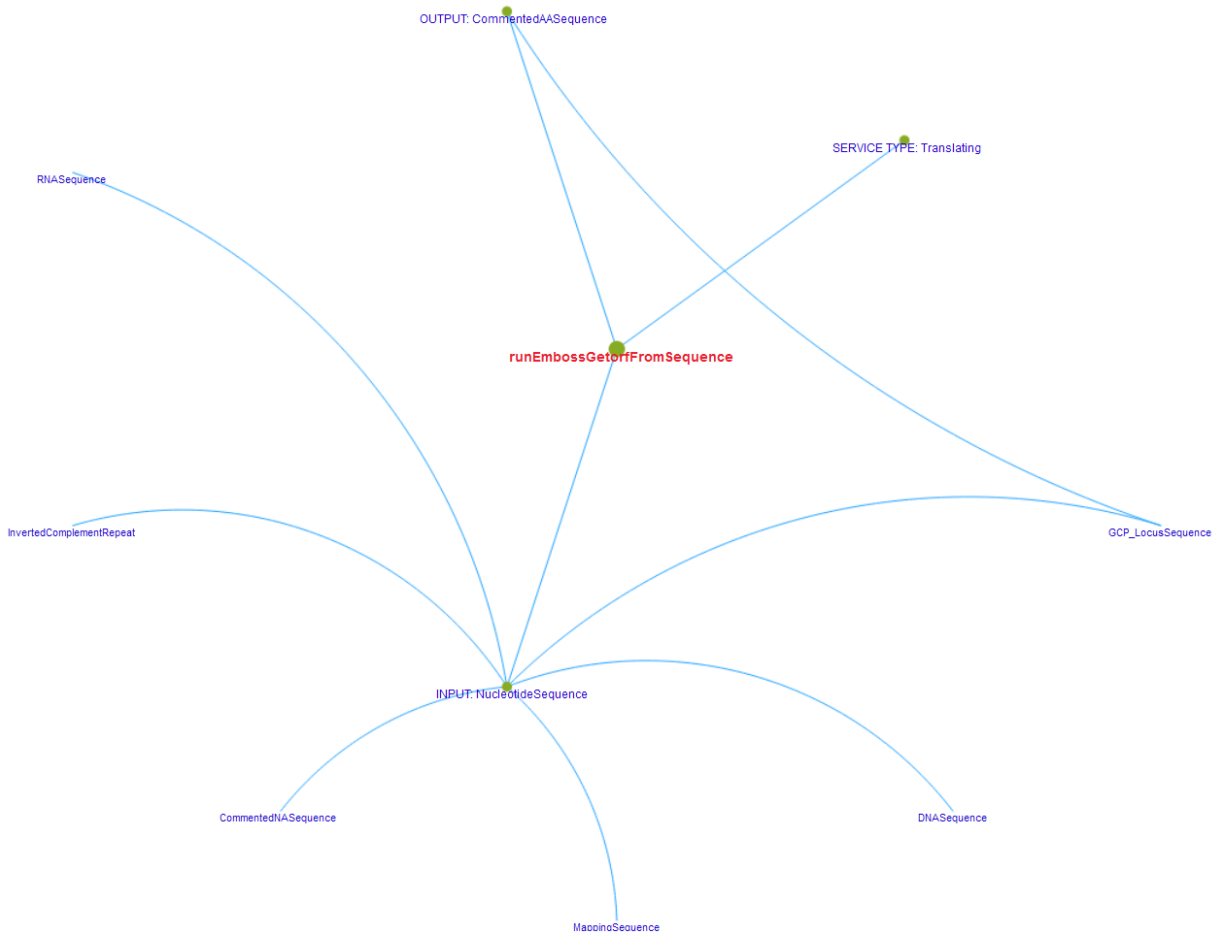

Figure 7: Service Graph. Service Graph for runEmbossGetorfFromSequence.

the user can indicate that he/she wants to search services the inputs of which is a "*NucleotideSequence*" and that performs a "*Translating*" operation (Figure 6). In this case we obtain only two services that fit the search parameters (B search in Figure 6).

Reasoning in the ABox[1] is based on the facts introduced into the repository. BioSStore Client User Interface takes advantage of ABox reasoning by providing suggestions in the search, filtering for the users, improving their search experience. For example, when the user selects a service type, the system can infer the possible input or output data types, and allows the user to select the one that best fits with his/her enquiry.

[1] The terms ABox and TBox are used to describe two different types of statements in ontologies. TBox statements describe a system in terms of controlled vocabularies, for example, a set of classes and properties. Abox are TBox-compliant statements about that vocabulary. TBox statements are sometimes associated with object-oriented classes and ABox statements associated with instances of those classes.

The use of a reasoning service allows the interface to locate services under the ontology perspective. For example, if we search for services with a "*GenericSequence*" as input, most of the systems only retrieve the services annotated with this information. By contrast, BioSStore is able to retrieve not only those services annotated with this input (e.g. "*fromGenericToAminoAcidSequence*") but also those annotated with any of its descendants (e.g. those with an input annotated as "*AminoacidSequence*", such as "*ConvertAAtoFASTAAAService*").

The list of results can be reduced by an additional filtering by restricting the input type, output type or service type. This can be done by selecting one of the descendants of concepts of the ontology indicated in the initial user query. For example, if the user locates the services with receive as input a nucleotide sequence (*NucleotideSequence*) then this search (C in Figure 6) can be reduced by selecting descendants of this semantic concept. This selection is done by clicking on the *DNASequence* link in the user interface (D in

Figure 6 shows the result of this action). In this case, the filtering is reduced from 49 to 28 services.

For those users with little knowledge of the domain ontology, a graphical representation of each ontology facet (data type and service type) is provided to locate services (Figure 7). This representation shows the concepts that describe a service.

3 RELATED WORK

Web service technologies have been widely adopted by the bioinformatics community, producing a huge amount of repositories with real services, such as BioMoby [7], EMBRACE Service Registry [10], BioCatalogue [11], DAS Service Registry [12] and Magallanes [13]. Table 1 shows the comparison in the semantic characteristics of these systems. This comparison is based, not only on the use of ontologies in the annotation of the Web services, but also on the use given to these annotations. Thus, the table also shows whether a standard reasoner has been used to infer new knowledge or improve the user's searches.

BioMoby is a project with the goal of producing an open-source, simple, extensible platform to enable the discovery, representation, integration, and retrieval of biological data from widely disparate data hosts and analysis services. In this platform, data and data analysis tools (for analyzing or transforming data) are distributed in Web services. Resources are registered in a central server called MOBY central. BioMoby objects are lightweight XML coded data used as query input and output values. Thus, the primary components of this infrastructure are MOBY Services (bioinformatics software tools), MOBY Objects (input and output data for the services) and MOBY Central (a register of all resources). This proposal also offers Object and Service hierarchies to classify available services, helping users to understand the meaning of the data required by them. The most interesting characteristic of BioMOBY is that there are several BioMOBY central repositories providing a set of more than 1500 real services that can be used to test discovery and composition algorithms. Moby 2.0/CardioSHARE [14] is a RDF-based system that aims to provide a higher level of functionality and reasoning capabilities. In this approach data is interchanged in RDF and queries are expressed in SPARQL. This project expects Web services to be able to consume and produce RDF. Currently, it is based on the use of bio2rdf [15]. The main aim is to enable the use of standard OWL reasoning techniques to improve service discovery.

EMBRACE Service Registry [10] is a collection of life-science Web services with built-in service testing. This is a prelude to the internationally supported BioCatalogue system that will collect, store, validate, and make web-services available in the biosciences. Users can search or browse the registry for services that match their needs. One important and useful characteristic of this system is that each entry includes live test data. Services are syntactically annotated using BioXSD [16]. BioXSD has been developed by studying the existing Web services, tools and data formats. It is also based on the effort of the community, including indications obtained by consulting the bioinformatics community.

BioCatalogue [11] is a repository of Web services in life sciences. This repository tends to provide a way of locating useful services through easy to use interfaces. Thus, the services are annotated to enable the discovery. However, semantics is not fully used, and inference is not applied to locate services. The service discovery is based on Web 2.0 filtering mechanisms that allow the user to filter by: Service Type, Provider, Submitter and Country. This repository includes interesting proposals for improving the quality

Table 1. Comparative table of semantic characteristics of the tools and registries. Semantic annotations can be of different types to the ones used in OWL (second characteristic). The use of a reasoner for improving the service location is also included in this table. Finally, the description of the services as semantic Web services is determined by the use or generation of OWL-S, WSMO or SAWSDL metadata.

Tool or Service Registry	Semantic annotations	OWL ontologies	Reasoner	Semantic Web services
BioSStore	X	X	X	X
BioMOBY	X			
Moby 2.0	X	X		
EMBRACE	X			
BioCatalogue	X	X		
DAS				
Magallanes				

of the services. Thus, a community based approach is followed to register services and document them. Then, services are monitored to detect their status and inform users about the availability. In this registry, myGrid ontology is used to annotate the services, but a new ontology, the EDAM Ontology, has been developed to improve these annotations.

The Distributed Annotation System (DAS) is a network protocol used in the exchange of biological data [12]. It is usually applied to the annotation of genomes and protein sequences. DAS is based on the idea of reference objects, which are biological data objects with stable identifiers. These objects are the target of annotations. Annotations follow a fixed XML syntax. These annotations include the authority (the institution providing the data), the type (the physical entity referred, such as Chromosome, Clone, Contig, etc.), the organism and in some cases the versioning scheme. DAS follows the REST (Representational State Transfer) paradigm, and the results of the requests are XML documents. A public registry is available of DAS sources with entries from more than 250 distinct sources.

Magallanes [13] is a library of algorithms for simplifying the discovery of bioinformatics Web services. It also provides a way of composing compatible services into workflows. The discovery is based on a Google-like approach, in which the user's keywords are matched with the metadata descriptions of the Web services. These searches are improved by a learning process from the user's feedback.

Taverna [17] is a tool for the design and execution of workflows, which can include not only remote Web Services, but also local tools. However, the location of remote Web Services depends on external repositories. In this case, Taverna relies on the BioCatalogue and Biodiversity Catalogue service registries for the discovery and use of Web Services. The use of semantically annotated services would improve the location of services to be used in workflows. Taverna users can use pre-designed workflows like those published on MyExperiment [18], a registry of biological workflows.

4 EXPERIMENTAL RESULTS

This section describes in more detail, the use cases, described in Section 2, and how they are solved in BioCatalogue. Firstly, we have tried to specify in BioCatalogue the search for services the input of which is a Sequence. The autocomplete function of this system allows us to choose from among different Sequence elements. In our system we selected Generic Sequence, and we have used a similar one in this repository: Biological Sequence. Results listed are: DDBJ_BLAST, BLAST blastp protein similarity search (CNRS IBCP), blastProDom, Gib, Fasta, BLAST (DDBJ) and ClustalWBioSStore returns, in this case, 81 results. If the search is reduced to the Amino Acid Sequence, the list is reduced to 36 results, which is nearer to BioCataloge results: Blast_Against_RefSeq_Complete_Sequenced_Organisms, ConvertAAtoFASTA_AA, EBI_WU_Blast, RPSBlast, etc. To the contrary, if we select just Sequence in this system, the list of results reaches 1071 results, but it is not possible to know whether the sequence is used by the service, produced by the service or part of the natural language comments. This is one of the main differences in the filtering, as Biosstore allows filtering, not only by service type (as BioCataloge does), but also by the service input and/or output.

The second case we have tested in the two systems is the search by service type. BioCatalogue allows us to search for Sequence Analysis, obtaining 515 results. Then, the user can filter the services by the Category Sequence Analysis, obtaining 97 services. This behavior is a bit strange as the generic search is able to obtain more services that are maybe annotated with Sequence or Analysis in their documentation. Then the user can select a more specific service type from the service list (for example Protein Sequence Analysis). In this case, the results are a bit strange as we obtain 163 services for a more specific search. BioSStore allows the user to select the Service Type Sequence Analysis, resulting in 36 services. Later, the user can filter them by adding a more specific type such as Nucleotidic_Motifs, Repetitions, Restrictions, Primers, Protein_Motifs, Mutation or Composition. Selecting, for example, Composition the list is reduced to three results. The use of a formal classification and a reasoning service enables the consistent filtering of the services when selecting more specific inputs, outputs or service types.

5 SUMMARY AND CONCLUSIONS

In this paper we have presented how a novel approach, to deal with the annotation of Web services using semantics, can be used to produce a semantic-based user interface for improving bioinformatics service discovery. Thus, our main goal has been to take advantage of this semantics (by means of reasoning) in the discovery process.

Our proposal is based on the use of standards, making it more adaptable to other complementary solutions, aiming to combine efforts in different areas of research in Web services. In this respect, the semi-automatic composition of Web services is another issue for future development. Besides, the use of standard

technologies makes this repository an interesting resource for testing algorithms under development in the Semantic Web Service community.

However, this proposal will encounter some limitations on growth because of the limitations of the current reasoners in the amount of instances they can manage. Thus, we are working on adapting and using a large scale reasoner for the search engine. Along these lines, we are analyzing the migration to DBOWL [19], a scalable reasoner based on a relational database.

The annotation process is costly for huge repositories, because the manual annotation could require a lot of human effort. For this reason BioSStore's ongoing work includes several approaches:

- Ontology Alignment: if we have a service annotated with a domain ontology, and we provide an alignment between this ontology and a new one, the tool will be able to provide annotations with this new ontology automatically.
- Text Mining: current semantic annotations and natural language documentation of such services can be used to generate a thesaurus for text mining. Thus, this thesaurus can be the basis for extracting automatic annotations from services using similar language terms.
- Social Curation: Biosstore users can answer simple questions upon entering the tool to validate automatic annotations or provide service annotation (for those that cannot be automatically annotated).

Finally, we would like to stress that our proposal is based on standard technologies, and this enables new ways of combining existing approaches to improve the user's experience in locating services required for their daily work. Usually, tools to build pipelines or workflows rely on external repositories when using remote Web Services. One of the problems that a user has to face is the location of a service that is able to deal with a data not only on a syntactic level, but also on a semantic level. So, the use of formal annotations would help users to select the most appropriate services for a given pipeline or workflow design and execution. Therefore, we plan to study how to combine BioSStore with different Open Source proposals to introduce semantics further into the service searches.

ACKNOWLEDGEMENTS

Supported by Grants TIN2011-25840 (Ministerio de Ciencia e Innovación) and P11-TIC-7529 and P12-TIC-1519 (Plan Andaluz de Investigación, Desarrollo e Innovación).

REFERENCES

[1] Roman, D., Lausen, H. and Keller, U. (2007) Web Service Modeling Ontology (WSMO) WSMO Final Draft 13 April 2005. wsmo.org, 2007.http://www.w3.org/Submission/WSMO/.

[2] The OWL Services Coalition (2004) OWL-S 1.1 Release (2004). http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/.

[3] Farrel, J. and Lausen, H. (2007) Semantic Annotations for WSDL and XML Schema. W3C Recommendation (August 2007). http://www.w3.org/TR/sawsdl/.

[4] Christensen, E. et al. (2001) Web Services Description Language (WSDL) 1.1. W3C Note (March 2001). http://www.w3.org/TR/wsdl/.

[5] Chinnici, R. et al. (2007) Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation (June 2007). http://www.w3.org/TR/wsdl20/.

[6] Thompson, H.S. et al. (2004) XML Schema Part 1: Structures Second Edition. W3C Recommendation (October 2004). http://www.w3.org/TR/xmlschema-1/.

[7] The BioMoby Consortium (2008) Interoperability with Moby 1.0: It's Better than Sharing Your Toothbrush!. Briefing in Bioinformatics, 9, 220-231.

[8] Vitvar, T. et al. (2008) WSMO-Lite Annotations for Web Services. In Proceedings of 5th European Semantic Web Conference (ESWC).

[9] Bishop, B. et al. (2011) OWLIM: A family of scalable semantic repositories. Semantic Web, 2, 33-42.

[10] Pettifer, S. et al. (2009) An active registry for bioinformatics web services. Bioinformatics, 16, 2090-2091.

[11] Bhagat, J. et al. (2010) BioCatalogue: a universal catalogue of web services for the life sciences. Nucleic Acids Research , 38, W689-W694.

[12] Prlic, A. et al. (2007) Integrating sequence and structural biology with DAS. BMC Bioinformatics, 8, 333-333.

[13] Rios, J., Karlsson, J. and Trelles, O. (2009) Magallanes: a web services discovery and automatic workflow composition tool. BMC Bioinformatics, 10, 334-334.

[14] Vandervalk, B.P. et al. (2009) Moby and Moby 2: Creatures of the Deep (Web). Briefings in Bioinformatics, 10, 114-128.

[15] Belleau, F. et al. (2008) Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. Journal of Biomedical Informatics, 41, 706-716.

[16] Kala, M. et al. (2010) BioXSD: the common data-exchange format for everyday bioinformatics web services. Bioinformatics, 26, i540-i546.

[17] Wolstencroft K, Haines R, Fellows D, et al. (2013) The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Res. 2013;41:W557-W561.

[18] Goble, C.A., et al. (2010) myExperiment: a repository and social network for the sharing of bioinformatics workflows, Nucl. Acids Res., 2010.

[19] María del Mar Roldán-García, José F. Aldana-Montes (2012) Evaluating DBOWL: A Non-materializing OWL Reasoner based on Relational Database Technology. ORE 2012.

AUTHOR BIOGRAPHIES

Dr. Ismael Navas Delgado holds the MSc degrees in Computer Science (2002), and the PhD degree in Computer Science (2009). Currently, he is Assistant Professor at the University of Malaga since 2009, and has been involved in a number of national and international research projects as participant researcher. His research activity is focused on the application of database technologies to the Semantic Web, including topics such as middleware development, semantic annotation, ontology location, ontology alignment, the intersection between Big Data and Semantic Web and with a wide experience in the development of bioinformatics applications.

Prof. José F. Aldana-Montes holds the BSc. and MSc degrees in Computer Science, and the PhD degree in Computer Science. He was Dean of the Faculty of Computer Science. Currently, he is a Professor at the University of Malaga, and has been involved in a number of national and international research projects as senior researcher. His research activity is focused on the application of database technologies to the Semantic Web and the intersection between Big Data and Semantic Web.