Database Interoperability through Web Services and Ontologies

Timos Sellis, *Member, IEEE*, Dimitrios Skoutas and Konstantinos Staikos

***Abstract*—This paper reports on efforts towards database integration and interoperability, based on Web services and ontologies. The development of Web Services software for three Biological Institutes is discussed. The software was created considering three major biological institutes which possess Databases with heterogeneous information. The goal was to make this information available over the Internet, in the form of Web services so that they can be used from other applications over the Web. Moreover, we discuss how to extend this functionality through the use of ontologies so as to allow for more effective and automatic data integration.**

I. Introduction

Recent years have seen an explosive growth in biological data. Today, bioinformatics information systems deal with large data sets in the order of terabytes. This data is stored in large Databases, whose number is over 1000, and mostly heterogeneous in terms of data types and formats used, as well as in terms of their design. The need of collecting and integrating data from different Molecular Biology Databases is an issue of increasing importance in Computational Biology. Therefore bioinformatics is an emerging scientific discipline that needs information technology to organize, analyze, and distribute biological information in order to answer complex biological questions. Drawbacks like data awareness and data retrieval should be overcome. All these reasons make the issue of database integration through well designed web-based interfaces as a premier issue and an active area of research in Molecular Biology Databases and hence in Bioinformatics.

In order to facilitate universal access to bioinformatics data and analysis software, Web services have much to offer. Web services are a type of service that can be shared by and used as components of distributed Web-based applications. They commonly interface with existing back-end applications, such as customer relationship management systems, order-processing systems, and so on. Web services are defined to share the following properties that make them easily accessible from heterogeneous environments:

1. they are accessed over the web
2. they describe themselves using an XML-based description language
3. they communicate with clients (both end-user applications and other Web services) through XML messages that are transmitted by standard Internet protocols, such as HTTP, SOAP, etc.

A number of online bioinformatics databases and services are currently available. Given that Web services allow programmatic access to data, the data providers would register their services in a formalized service registry, and researchers' scripts would no longer need to be concerned with the interface details of the different databases, but they would use Web services instead to access required data.

Major benefits of using Web services in Bioinformatics would be:

1. Interoperability among distributed applications that span diverse hardware and software platforms. Interoperability could be in terms of providing distributed access to multiple bioinformatics services, aggregating data from multiple sources, providing a centralized registry for finding new services etc.
2. Easy, widespread access to applications through firewalls using Web protocols.
3. A cross-platform, cross-language data model (XML) that facilitates developing heterogeneous distributed applications.

On the other hand, resolving semantic issues among the data that is provided and exchanged across distributed and heterogeneous sources is an imperative task in order to allow for effective integration and interoperability. To this end, ontologies play a critical role. An ontology is often defined as a formal and explicit specification of a shared conceptualization. It provides a way for describing the meaning and the relationships of the terms in a domain. More specifically, it describes the knowledge of a domain in terms of classes, i.e., groups of individuals, and properties, i.e., attributes or relationships between them.

In the following section, we report on the use of Web services in integrating the databases of three major biological institutes [1]. Then, we discuss the role and benefits of ontologies in database integration. Finally, we give an overview of Semantic Web services, which leverage the power of both Web services and ontologies, to achieve interoperability both at the syntactic and at the semantic level.

Manuscript received July XX, 2008.

T. Sellis is with the Institute for the Management of Information Systems and the National Technical Univ. of Athens, Athens, Greece (phone: +30-210-6990522; fax: +30-210-6990552; e-mail: timos@imis.athena-innovation.gr).

D. Skoutas is with the National Technical Univ. of Athens, Athens, Greece (e-mail: dskoutas@dblab.ece.ntua.gr).

K. Staikos is with European Dynamics SA, Athens, Greece (e-mail: kstai@eurodyn.com).

II. Integrated Web Services in Bioinformatics

There are more than 1000 molecular biology databases. In spite of the recent surge of interest in Molecular Biology databases, these databases are rather unknown outside Computational Biology and Molecular Biology. Computer scientists and database experts are rarely knowledgeable about these databases and their uses. This is regrettable because there is a considerable need for further work and more database expertise in Computational Biology. Especially traditional database issues such as data modelling, data management, query answering, database integration as well as novel issues such as data mining and knowledge discovery deserve more consideration in Computational Biology.

Taking into account the constantly increasing number of Databases, their dissimilarity as far as data is concerned and the fact that a biologist is in general not aware of all the databases relevant to its investigation (he/she uses 3 to 5 in average) the question which arises is simple: *how to find and retrieve the needed data quickly and accurately*. Hence, the goal of this project was to build a Web application able to give direct answers to biologists, without relying on their knowledge on database systems or programming languages.

Another issue is that molecular biology databases are heterogeneous. A widespread practice in Molecular Biology is that a research team first analyzes some data it has generated or collected (e.g. from databases or from the literature), and then makes these data available to the research community through a database. Many Molecular Biology databases have been developed in this manner. As a consequence, Molecular Biology databases are highly distributed and heterogeneous, reflecting the distribution and heterogeneity of the Molecular Biology research community. Collecting and integrating data from different Molecular Biology databases is an issue of increasing importance in Computational Biology, for the detection of similarities between data from distinct origins is prevalent in Molecular Biology. Therefore in this application three heterogeneous databases were used to retrieve the eligible result.

There are several types of molecular biology databases, among them:

- Biological Ontology Databases, like GO, GOA, Ontology Lookup, ChEBI, etc.
- Literature Databases, like MEDLINE
- Microarray Databases, like ArrayExpress
- Nucleotide Databases, like EMBL Nucleotide Sequence Database, Parasites, Mutations, etc.
- Pathways & Networks, like Reactome and BioModels.
- Protein Sequence Databases, like UniProt, UniRef, UniProtKB/Swiss-Prot and many more.
- Proteomic Databases, like PRIDE.
- Structure Databases, like DALI, PDB, FSSP, etc.

In this project we used the EMBL, MEDLINE and Array Express databases. The goal of this particular selection was realistic: for a given Nucleotide Number (EMBL database), find all experiments (Array Express database) and all publications (MEDLINE) which have taken place.

A. *EMBL database*

In Europe, the vast majority of the nucleotide sequence data produced is collected, organised and distributed by the EMBL Nucleotide Sequence Database located at the EBI in Cambridge UK, an Outstation of the European Molecular Biology Laboratory (EMBL) in Heidelberg, Germany. EMBL is a data repository, accepting nucleic acid sequence data from the scientific community and making it freely available. The databases strive for completeness, with the aim of recording every publicly known nucleic acid sequence. These data are heterogeneous, they vary with respect to the source of the material (e.g. genomic versus cDNA), the intended quality (e.g. finished versus single pass sequences), the extent of sequence annotation and the intended completeness of the sequence relative to its biological target (e.g. complete versus partial coverage of a gene or a genome). EMBL is distributed free of charge over the internet.

B. *MEDLINE database*

MEDLINE is the biggest online bibliographic database to medicine, nursing, health services, etc. It was founded in 1966. It is produced by the National Library of Medicine and contains bibliographic records of journal articles (citations). It is monthly updated and covers subjects as: Aging; Athletics; Biochemistry; Biology; Exercise Science; Food Sciences; Gerontology; Health Sciences; Medicine; Microbiology; Nursing; Nutrition; Physiology; Public Health; Speech and Hearing; Sports Medicine. Coverage is worldwide, but most records are from English-language sources or have English abstracts.

C. *Array Express database*

ArrayExpress is a public repository for microarray data, which is aimed at storing well annotated data in accordance with MGED recommendations. It is a public database of gene expression experiments. The data relating to each microarray project in the ArrayExpress database is subdivided into two main components: the *Array*, which refers to information about the design and manufacture of the array itself, and the *Experiment*, which provides information on the experimental factors and the actual data obtained. In addition to these, a third component, *Protocol*, describes the procedures used in the production of the array or the execution of the experiment.

A Web application was built for the communication of these three institutes/databases with the non-realistic "Nucleotide Experimental and Bibliographic Information Centre (N.E.B.I.C)". Therefore physically we dealt with 4 totally different applications, with the three of them (database institutes) offering their data to the fourth (N.E.B.I.C) one. This communication was based on the

SOAP protocol and used open source technologies, such as XML, EJB (Entreprise Java Beans), XDoclet , Axis, Servlets – JSPs, Struts and Ant.

The project described above was pursued in 2004 only to show how novel technologies at that time could be brought together to solve the interoperability problem among various databases. We proceed next to describe how this process can be enhanced through the use of ontologies.

III. Using Ontologies to Enhance Integration

By standardizing the format and the structure of the interfaces and of the exchanged messages, Web services provide interoperability among distributed and heterogeneous applications and platforms. However, apart from this interoperability at the syntactic level, semantic interoperability is also a crucial, and often even more challenging, requirement.

Semantic heterogeneity refers to the intended meaning of the information. In order to achieve semantic interoperability in a heterogeneous information system, the meaning of the information that is exchanged has to be understood across the communicating subsystems. Three main causes for semantic heterogeneity are usually identified [1]: (a) "confounding conflicts", which occur when information items seem to have the same meaning, but differ in reality; e.g. owing to different temporal contexts; (b) "scaling conflicts", which occur when different reference systems are used to measure a value; e.g. different currencies or different date formats; and (c) "naming conflicts", which occur when naming schemes of information differ significantly (a frequent phenomenon is the presence of homonyms and synonyms.)

The main challenge that arises is how to identify the appropriate data sources for a given information need, and how to specify the transformations required in order to extract and integrate data from them. The schema of a data source describes the way that data are structured when stored, but does not provide any information for their intended semantics. Therefore, metadata are required to allow for the understanding, management, and processing of these data. Using domain ontologies, it is possible to semantically annotate the involved data sources and infer mappings between them. Exploiting the information conveyed by the ontology and the annotations, it is possible to provide a measure indicating how close semantically the information provided by two data sources is. Furthermore, the use of ontologies allows to identify and construct, in a semi-automatic manner, the processes required to clean and reconcile data coming from different sources [3].

In the recent years, research efforts towards the realization of the Semantic Web vision have lead to the standardization of ontology languages such as RDF(S) and OWL. OWL, and more specifically its OWL DL part, is based on Description Logics, a decidable fragment of First Order Logic, constituting the most important and commonly used knowledge representation formalism. Apart from specifying classes and properties, and organizing them hierarchically, it supports a variety of other constructs such as defining properties as being symmetrical, inverse or transitive, or specifying arbitrary cardinality constraints. These standardization efforts have further facilitated the development of tools for creating, maintaining, and reasoning with ontologies, such as the Protégé ontology editor or the Pellet OWL reasoner. This makes it possible even for non-experts, i.e., without requiring specific programming skills, to incorporate ontologies in their applications and benefit from their use.

Regarding the use of ontologies in information integration, several works have been proposed in the literature, which can be classified in three broad categories: single ontology approaches, multiple ontologies approaches, and hybrid approaches [4]. A single, "global", ontology simplifies the integration process, but it is difficult to create and maintain, especially in the presence of changes in the data source schemas. On the other hand, multiple ontologies provide flexibility; however, comparing the sources becomes considerably more difficult. In hybrid approaches each source is described by its own ontology, using terms from a global, shared vocabulary.

IV. Interoperability Through Semantic Web Services

Combining the advantages of both worlds, Web services and ontologies, has lead to the development of Semantic Web services. These are Web services that are semantically described, i.e., the parameters in their descriptions are annotated by concepts from an associated domain ontology. In particular, three main approaches have been proposed for bringing semantics to Web services: OWL-S, WSDL-S, and WSMO. The main idea in all these approaches is to use appropriate ontologies to semantically annotate the various aspects of a Web service description, such as inputs, outputs, preconditions, and effects, as well as non-functional parameters (e.g., QoS parameters). With Semantic Web services interoperability and integration is further facilitated, as it becomes possible to reason about service descriptions, and thereby to automate tasks such as service discovery, for finding and fetching the required information, and service composition, for combining simpler software components to perform more complex tasks and workflows.

In the following, we give an overview of OWL-S, and discuss how the semantic enhancement of service descriptions facilitates the automation of service discovery and composition. OWL-S is an ontology for describing

Semantic Web Services, built on top of the Web Ontology Language (OWL). It consists of three subontologies, describing, respectively, service profiles, service models and service groundings, which correspond to different aspects and levels of granularity of service descriptions. In general, the service profile is useful for service advertisement and discovery, while the model and the grounding provide information about how an agent can use a discovered service. A more detailed description of these parts is given below.

A service profile gives a high-level description of what the service provides to its clients. Hence, it is used during matchmaking to determine whether the service meets the client's needs. The functionality of the service is specified in terms of inputs and outputs, as well as preconditions, which are requirements that the client should satisfy in order to use the service, and effects that may result from the service execution. It also contains information about other features of the service, such as the entity or the organization that offers it, the category of the service in a given classification system or quality ratings (response time, reliability, etc.).

A service model describes how the service works, presenting it from the perspective of a process. Given a request, a process may either return some information, specified by its inputs and outputs, or produce a change in the world state, specified by its preconditions and effects. In addition, a process may be either atomic or composite. In the first case, there is a single interaction between the client and the service, i.e., the client sends a single request and receives a single response. In the second case, there is a series of interactions, i.e., of exchanged messages, with the service maintaining some state throughout it. OWL-S provides a set of control constructs for specifying composite processes, such as sequence, split, if-then-else, repeat-while. The specification of data flow between the sub-processes is supported, as well.

Finally, a service grounding provides information about how to access and interact with the service, such as the communication protocol to be used and the structure of the exchanged messages. Essentially, it grounds the service description to a concrete implementation. This is achieved in conjunction with WSDL, which has been chosen due to its widespread use in industry for describing Web services. For example, an OWL-S atomic process is mapped to a WSDL operation.

Once the descriptions of the available Web services have been semantically enhanced as discussed previously, the task of matching a user request with a published service is essentially based on the use of logic inference to check for equivalence or subsumption relationships between the ontology classes annotating the request and service parameters. Typically, the following types of match are identified [5], [6]: (a) exact, if the request is equivalent to the advertisement; (b) plug-in, if the request is subsumed by the advertisement; (c) subsume, if the request subsumes the advertisement; (d) intersection, if the intersection of the request and the advertisement is satisfiable; and (e) disjoint, otherwise. Our recent work extends and elaborates on this matchmaking framework, focusing on ranking the results of the matcher, so as to facilitate the selection of the most suitable candidates for a given request. In particular, our approach presented in [7] uses the measures of recall and precision to evaluate the similarity between the requested and the offered service, and expresses this similarity as a continuous value in the range [0..1]. In addition, efficient service discovery based on the notions of dominance and skyline has been proposed in [8], [9].

However, it is possible that no single service exists that can provide the information or the functionality required by the user. In this case, it should be possible to combine existing services together to perform the final task. Several approaches have been proposed for this purpose. For example, the work presented in [10] leverages the semantic annotation of the service parameters, and proposes an AI planning-oriented model for service composition. It is based on a data structure called causal link matrix, which maintains valid semantic connections between existing services.

Summarizing, the aim of Semantic Web services and techniques such as those described above is to make it possible that a software agent, given the high-level description of a complex task, is able to automatically discover, compose, invoke and coordinate services to achieve this goal.

V. Conclusions

Vast amounts of biological data are distributed in heterogeneous databases. In this paper we have discussed the key role that Web services, ontologies, and their combination, Semantic Web services, can play in integrating such data sources, so that the researcher can effectively and efficiently seek and compose the desired information, and hence benefit from this available wealth of knowledge.

References

[1] K. Staikos, *Integrated Web Services in Bioinformatics*, MSc Thesis, Tehnische Universität München, 2004.

[2] C. H. Goh, *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources*. MIT, 1997.

[3] D. Skoutas and A. Simitsis, "Ontology-based Conceptual Design of ETL Processes for both Structured and Semi-structured Data," *International Journal on Semantic Web and Information Systems, Special Issue on Semantic Web and Data Warehousing*, 2007.

[4] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner, "Ontology-Based Integration of Information.

A Survey of Existing Approaches," *In Proc. of the IJCAI workshop on Ontologies and Information Sharing*, 2001.

[5] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara, "Semantic Matching of Web Services Capabilities," *In Proc. of the 1st International Semantic Web Conference*, 2002.

[6] L. Li and I. Horrocks, "A Software Framework for Matchmaking based on Semantic Web Technology," *In Proceedings of the 12th International World Wide Web Conference*, 2003.

[7] D. Skoutas, A. Simitsis, and T. Sellis, "A Ranking Mechanism for Semantic Web Service Discovery," *In Proc. of the ICWS workshop on Semantic Web for Services and Processes*, 2007.

[8] D. Skoutas, D. Sacharidis, A. Simitsis, and T. Sellis, "Serving the Sky: Discovering and Selecting Semantic Web Services through Dynamic Skyline Queries," *In Proc. of the 2nd IEEE International Conference on Semantic Computing*, 2008.

[9] D. Skoutas, D. Sacharidis, V. Kantere, and T. Sellis, "Efficient Semantic Web Service Discovery in Centralized and P2P Environments," *In Proc. of the 7th International Semantic Web Conference*, 2008.

[10] F. Lécué and Alain Léger, "A Formal Model for Semantic Web Service Composition," *In Proc. of the 5th International Semantic Web Conference*, 2006