CardioSHARE: Web Services for the Semantic Web

Ben Vandervalk, Luke McCarthy, and Mark Wilkinson

The Providence Heart and Lung Research Institute at St. Paul's Hospital,
University of British Columbia, Department of Medical Genetics

Abstract. Here we present CardioSHARE, a unique framework for querying distributed data and performing data analysis using Semantic Web standards. CardioSHARE's two main innovations are an enhancement to a standard SPARQL query engine, which enables the required data to be retrieved dynamically from web services; and the ability to use this dynamic data to identify instances of OWL classes. This is accomplished by mapping RDF predicates onto web services capable of producing data that satisfy those predicates. Our initial focus has been on integration with the BioMoby project: a set of 1500+ interoperable bioinformatics web services. CardioSHARE effectively brings this established pool of resources into conformance with Semantic Web standards.

1 Introduction

Data integration continues to be a major issue in bioinformatics. On a daily basis, researchers are required to gather, merge, and query data from multiple public databases on the web (e.g. PubMed [1], KEGG [2], UniProt [3], Entrez Gene [4]). This is a highly tedious, inefficient and error-prone process that tends be done either manually or with custom scripts that "screen scrape" data from the source websites.

The problem has inspired the development of many software solutions, which can be broadly classified as either centralized or distributed. Centralized systems, or data warehouses, collect and merge related databases en masse, so that they can queried as a single resource. The data must continually be updated from the original sources, and these sources must be chosen carefully to minimize query times and curatorial burden. Examples of data warehouse systems include SRS [5], Bio2RDF [6], the HCLS Knowledge Base [7], and the Pathway Knowledge Base [8]. Distributed systems, on the other hand, function by breaking a user's query into a number of subqueries and issuing those subqueries against independent resources. In order for such a scheme to be feasible, each resource must implement a web service that serves data according to a common ontology. Examples of distributed systems include BIRN [9] and caCORE [10]. In addition, lower-level distributed systems such as BioMoby [11], SSWAP [12], and myGrid [13] also provide web services, but require that the user coordinates these web services explicitly through the construction of workflows.

Although these systems have been used successfully within their chosen domains (such as neuromedical imaging or cancer research) there is currently no widely adopted infrastructure for data integration in bioinformatics. The CardioSHARE (Cardiovascular Semantic Health and Research Environment) project, presented here, proposes to provide such a framework by building on the established standards of the Semantic Web. Although the project's major focus will be the analysis of clinical data on heart disease, CardioSHARE has been designed for use with any type of data. In order to better illustrate CardioSHARE's generic nature, the submission for this challenge provides semantic access to the existing bioinformatics web services of the BioMoby project.

2 Background: BioMoby

In principle, the use of web services for data integration has numerous advantages over warehousing. Such a system is more reliable, because maintenance of services is distributed across providers without a central point of failure; more efficient, because web service execution can be parallelized; and more scalable, because new services may be added to the system without affecting the performance of existing resources hosted elsewhere. In fact, the distributed approach to data integration subsumes the centralized approach, as data from warehouses may be included via services in the same manner as any other resource. Another major advantage of the web service approach is that data generated by analytical software (tools for sequence alignment, pattern recognition, structure prediction, etc.) can be incorporated in the same manner as static resources.

However, these theoretical advantages come at the cost of establishing a large community of interoperable services, and a system for coordinating their execution. In practice, even manual construction of workflows is difficult, because providers tend to invent their own XML syntaxes for input and output datatypes. Even in cases where syntaxes are identical, it is impossible to ascertain (in an automated fashion) whether two input/output objects really represent the same type of data. It is difficult for a program to determine, for example, whether a service that inputs and outputs a single string is performing a BLAST search, a keyword search for database accession numbers, or any number of other operations.

BioMoby is a web service framework that was created to address this issue. The most important aspect of the system is that all participating services are required to specify their inputs and outputs as instances of objects in a shared ontology of datatypes. This ontology incorporates common bioinformatics objects such as amino acid sequences, DNA sequences, SNPs, GO terms, and so on. The ontology not only specifies the semantic relationships between datatypes, but also the serialization of each datatype into XML, thus creating a community of truly interoperable services. Further, users may freely add new datatypes to the ontology as needed.

In addition to providing a common ontology, BioMoby also provides a large central registry of services. Programs can query the registry for services that consume or generate a given datatype, thus allowing for the semi-automated construction of workflows. There are currently over 1500 BioMoby web services which perform a variety of bioinformatics tasks including BLAST searches, retrieval of GO annotations, identification of protein domains, retrieval of database records, etc. Despite the success of the project, BioMoby has a number of shortcomings: as BioMoby predates the advent of the Semantic Web, it uses a custom XML format that is incompatible with other data frameworks. Also, even though BioMoby provides sufficient semantics to establish the input and output type of a service, it does a poor job of describing the actual operation that a service performs. This forces users to oversee the construction of their workflows, to ensure that they perform the desired analysis.

The CardioSHARE project builds on the strengths of BioMoby and addresses many of its weaknesses by replacing the boutique BioMoby syntax with one based on Semantic Web ontologies.

3 CardioSHARE: Querying Web Services with SPARQL

The key observation behind the CardioSHARE query engine is that when a web service generates an output, it is in effect generating an RDF triple. The subject of this triple is the input, the object is the output, and the predicate is the relationship between the input and output, as determined by the service. For example, a BLAST service generates triples with the “hasHomolog” predicate, as depicted in Figure 1. Considered as a whole, the collection of BioMoby services represents an enormous *virtual graph* of unrealized triples.

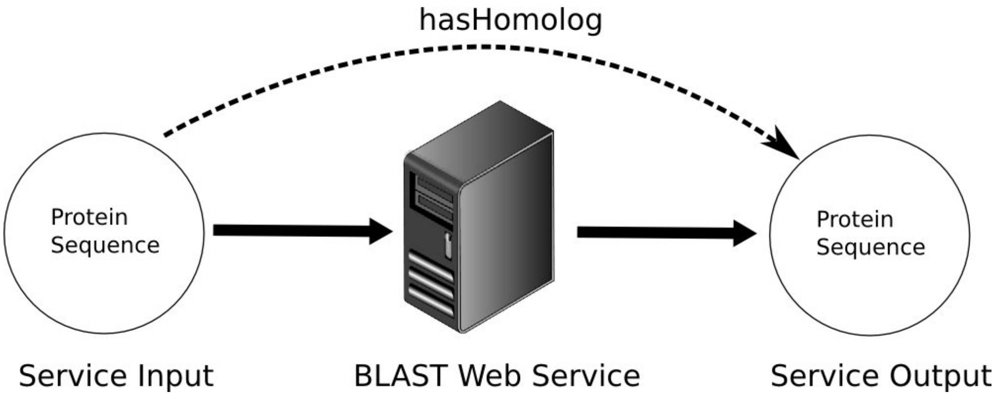

Fig. 1. The key idea underlying the CardioSHARE query engine. When web services generate output, they are dynamically generating a triple where the subject is the service input, the object is the service output, and the predicate is the relationship between the input and output, as determined by the service. In the example shown, the service runs a BLAST sequence similarity search, and the implicit predicate is “hasHomolog”.

The main functionality of CardioSHARE is the ability to issue queries against this virtual graph, as if it had been computed in its entirety and downloaded into a local triple store. A CardioSHARE query is resolved according to the following steps:

1. Each predicate in the query is examined and any matching services are retrieved from the registry.
2. The services are invoked, the results are converted to RDF, and the data is stored in the local triple store.
3. The query engine is executed as normal against the local triple store.

This approach allows for any locally stored data to be queried in conjunction with data in the virtual graph.[1] In addition, CardioSHARE queries are syntactically identical to ordinary SPARQL queries. An example query is shown in Figure 2, which asks "What proteins are the subject of PubMed article 14633995, and what organisms do these proteins belong to?".

```
PREFIX up: <http://uniprot/>

SELECT ?protein ?organism
WHERE
{
  <http://biomoby.org/PMID/14633995> up:isPaperAboutProtein ?protein .
  ?protein up:belongsToOrganism ?organism .
}
```

Fig. 2. An example CardioSHARE query, which asks: "What proteins are discussed in PubMed article 14633995, and what organisms do these proteins belong to?". To resolve this query, the CardioSHARE engine first finds services that are annotated with "isAboutPaperProtein" and "belongsToOrganism". Since the first triple has only one variable, this triple is resolved first, by invoking one or more "isPaperAboutProtein" services with the input http://biomoby.org/PMID/14633995. The outputs of these services are subsequently provided (one at a time) as input to "belongsToOrganism" services, in order to produce the final results. This query may be executed by typing it into the web form at `http://cardioshare.biordf.net/cardioSHARE/query`.

As in other web service based architectures, CardioSHARE has numerous advantages over data warehousing: services can provide uniform access to data that may have diverse ownership, networks locations, and formats; responsibility for maintenance of these data sets and services is distributed; query performance can be optimized by parallel execution of services; and new resources may be added without the limitations inherent to storing and indexing a monolithic database. In addition, CardioSHARE is unique in the sense that it does not rely

[1] In the current prototype implementation, no data is stored in the triple store prior to the query, and no data is cached afterwards.

on a single unifying ontology for integration. Service providers need not scour a comprehensive ontology such as BIRNLex or NCI Thesaurus in order to add a service into the system; they can choose the predicates from any ontology they are familiar with.

4 CardioSHARE: The Prototype Interface

An early prototype of the CardioSHARE system is accessible at `http://cardioshare.biordf.net/cardioSHARE/query`; this web application provides two paths to access the data: a web form for issuing SPARQL queries, and a browser to explore the virtual graph. For the purposes of the demonstration, a number of BioMoby services have been annotated with predicates. These predicates connect UniProt proteins to related data about publications, GO terms, KEGG genes, and so on. There is a help page to assist with the construction of queries, which includes a diagram depicting the relationships between the available predicates and the URIs they consume/produce. Several sample queries are also provided.

The graph browser offers an alternative way to access the available data. Starting with a single object (a UniProt protein, a GO term, a KEGG pathway, etc.), a user can see the available predicates relating to that object. By selecting a predicate, the user can invoke the appropriate service(s) and add the resulting output to the graph. This process can be repeated for any of the new output nodes, allowing the entire network of data relating to the initial object to be explored.

5 Acknowledgements

The development of CardioSHARE is made possible by the support of the Heart and Stroke Foundation of British Columbia and Yukon. MDW is funded for CardioSHARE through an operating grant from the Canadian Institutes of Health Research. BioMoby was developed under support from the Genome Canada/Genome Alberta bioinformatics Platform. Hardware for both projects has been provided by Sun Microsystems and IBM. Core laboratory funding is provided by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Pubmed Home. http://www.ncbi.nlm.nih.gov/pubmed/ (31 July 2008, date last accessed).
2. Kanehisa, M, Goto, S. KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Research 2000;28(1):27-30.
3. Apweiler, R,Bairoch, A, Wu, C. UniProt: the Universal Protein knowledgebase. Nucleic Acids Research 2004;32(Database Issue):D115-D119
4. Maglott, D, Ostell, J, Pruitt, K, Tatusova, T. Entrez Gene: gene-centered information at NCBI. Nucleic Acids Research 2005;33(Database Issue):D54-D58.

5. Etzold, T., et al.: SRS: Information Retrieval System for Molecular Biology Data Banks. Methods in Enzymolgy 266, 114-128 (1996)
6. Belleau, F., et al.: Bio2RDF: towards a mashup to build bioinformatics knowledge system. In: Proceedings of the WWW Workshop on Health Care and Life Sciences Data Integration for the Semantic Web 2007.
7. Health Care and Life Sciences Interest Group: A Prototype Knowledge Base for the Life Sciences, http://www.w3.org/TR/hcls-kb/ (31 July 2008, date last accessed).
8. Kotecha, N, Bruck, K, Lu, W, Shah, N. Pathway Knowledge Base: Integrating BioPAX Compliant Data Sources. In: HCLS Workshop, ISWC 2006.
9. Grethe, J.S., et al.: Biomedical Informatics Research Network: Building a National Collaboratory to Hasten the Derivation of New Understanding and Treatment of Disease. Stud. Health Technol. Inform. 112, 100-109 (2005)
10. Covitz, P.A., et al.: caCORE: A common infrastructure for cancer informatics. Bioinformatics 19, 2404-2412 (2003)
11. Wilkinson, M.D., Links, M.: BioMOBY: an open-source biological web services proposal. Briefings in Bioinformatics 3(4), 331-341 (2002)
12. Simple Semantic Web Architecture and Protocol (SSWAP), http://sswap.info/ (14 September 2008, date last accessed)
13. Stevens, R.D., Robinson, A.J., Goble, C.A.: myGrid: personalised bioinformatics on the information grid. Bioinformatics 19, i302-i304 (2003)