

Semi-automatic generation of Semantic Web Services for relational biological databases

Julien Wollbrett¹, Pierre Larmande² and Manuel Ruiz^{1§}

¹ CIRAD, UMR AGAP, F-34398 Montpellier, France

² IRD, UMR DIADE, Montpellier, France

[§]Corresponding author

julien.wollbrett@cirad.fr
pierre.larmande@ird.fr
manuel.ruiz@cirad.fr

Abstract: In recent years, a large amount of “-omics” data has been produced. However, these data are stored in many different species-specific databases that are managed by different institutes and laboratories. Biologists often need to find and assemble data from disparate sources to perform certain analyses. Searching for these data and assembling it is a time-consuming task. The Semantic Web helps to facilitate interoperability across databases. A common approach involves the development of wrapper systems that map a relational database schema onto existing domain ontologies. However, few attempts have been made to automate the creation of such wrappers. We developed a framework, named BioSemantic, for the creation of Semantic Web Services applicable to relational biological databases. This framework makes use of both Semantic Web and Web Services technologies and can be divided into two main parts: (i) the generation and semi-automatic annotation of an RDF view; and (ii) the automatic generation of Semantic Web Services. We have used our framework to integrate genomic data from different plant databases. BioSemantic is a framework designed to speed the development of Semantic Web Services for existing relational biological databases. Currently, it creates and annotates RDF views that enable the automatic generation of SPARQL queries. Web Services are also created and deployed automatically, and the semantic annotations of our Web Services are added automatically using SAWSDL attributes. BioSemantic is downloadable at

<http://southgreen.cirad.fr/?q=content/Biosemantic>.

Keywords: Semantic Web Services, ontology driven data integration, SPARQL query formulation

1 Introduction

Plant biologists and breeders often need to access several databases to perform tasks such as locating allelic variants for particular genetic markers in several crop populations and in a given environment or investigating the consequences of a particular mutation at the transcriptome, proteome, metabolome and phenome levels. However, biological data integration faces syntactic and semantic heterogeneity challenges. In their reviews, Stein [1] and Goble C & Stevens R [2] provide a fair criticism of the lack of integration approaches and provide a similar vision for the future, that is, that the Semantic Web (SW) can aid in data integration.

There are currently existing efforts to describe Web Services with semantic annotations by using ontologies, such as SSwap [3], SADI [4] and BioMoby [5]. The SADI framework provides a Protégé plug-in to simplify the coding of SADI Web Services; however, the provider has yet to develop a complete business logic [4]. The implementation of new Semantic Web Services (SWSs) can be time-consuming and requires the developer to know how to manipulate SW and WS standards. To our knowledge, there are currently no ongoing efforts in the context of the automation of Semantic Web Services creation both specific to relational databases and based only on W3C standards.

Our goal is to develop a framework for the creation of Semantic Web Services for the biology field by using both Semantic Web and Web Services technologies. We aim to create links between the relational database management systems that are widely used to store, manage and query biological data. To make the process of Web Service development as easy as possible, we have developed a semi-automated framework to accelerate the development of Semantic Web Services for relational biological databases.

We will detail below the entire process for generating a BioSemantic SWS, which can be divided into two main parts: (i) the generation and semi-automatic annotation of an RDF view (Fig. 1); and (ii) the automatic generation of the Semantic Web Service (Fig. 2).

2 Generation and semi-automatic annotation of an RDF View

A local RDF view of the database schema is automatically created for each relational database to be integrated. Then the RDF view has to be manually annotated by experts with terms from existing bio-ontologies. The RDF views, both created and annotated, are stored in a RDF repository (Fig 1).

2.1 Relational database-to-RDF mapping

The research in the domain of mapping between databases and ontologies is very active and corresponds to various motivations and approaches [6]. In BioSemantic, we use the mapping as an intermediate layer between the user and the stored data. This layer provides an abstraction of the database and allows the user to query databases without knowledge of the database schema. These characteristics correspond to the motivation known as “data access based on ontology”. For that purpose, we found only two tools that strictly use Semantic Web standards: Virtuoso [7] and D2RQ [8–10]. We have chosen D2RQ because this tool is open source and free. In addition, some bioinformatics projects have successfully used D2RQ. With D2RQ, we can automatically generate a mapping file that provides an RDF view of the database schema.

2.2 RDF view description

The RDF view generated by D2RQ contains the elements of the database schema: entities, attributes, keys (primary, foreign) and metadata, such as the database driver and host. The data contained in the relational databases are not included in the RDF view. Consequently, both the D2RQ API and the RDF view are requested when the data are accessed through SPARQL queries.

In the RDF view, the database schema is represented by a graph. Each node corresponds to an entity or attribute in the database, and each edge defines a relationship between two nodes. In RDF format, namespaces are used to uniquely identify each node. Namespaces provide a prefix for each node name. For example, the *map:marker* node indicates the “marker” concept from the “map” vocabulary.

2.3 Automatic semantic enrichment of the RDF view with BioSemantic

The BioSemantic API automatically detects specific information related to the relational database schema and translates it into new properties that can be integrated into the RDF view. These metadata are then used for SPARQL query generation. This step can be seen as a semantic enrichment of the RDF view.

Association tables

For this purpose, we have developed an algorithm that detects association tables:

pk= primary key of R

fk= foreign keys of R

$if((\forall u \in R)(u \in f k \Rightarrow u \in p k)) \{$

$if((\forall u \in R)(u \in p k \Rightarrow u \in f k) \{$

R is an association table

$\}$

$\}$

Arity

We can also detect the arity of association tables, i.e., the number of foreign keys they possess. The algorithm labels association tables in the RDF view with the *dr:associatedTo* property and indicates the arity with the *dr:arity* property.

Inheritance, aggregation and composition

There are many ways to transform inheritance relationships from an object-oriented conceptual model to a relational model [11]. For our algorithm, we detect relationships resulting from the transformation of each class in an inheritance hierarchy into a table. We also detect tables resulting from aggregation or composition relationships by using the identifying algorithm from [12]. We label these relationships in the RDF view with the *rdf:subClassOf* property.

The annotation of the RDF view is performed manually using a text editor and must be conducted by an expert familiar with the database and/or bio-ontology.

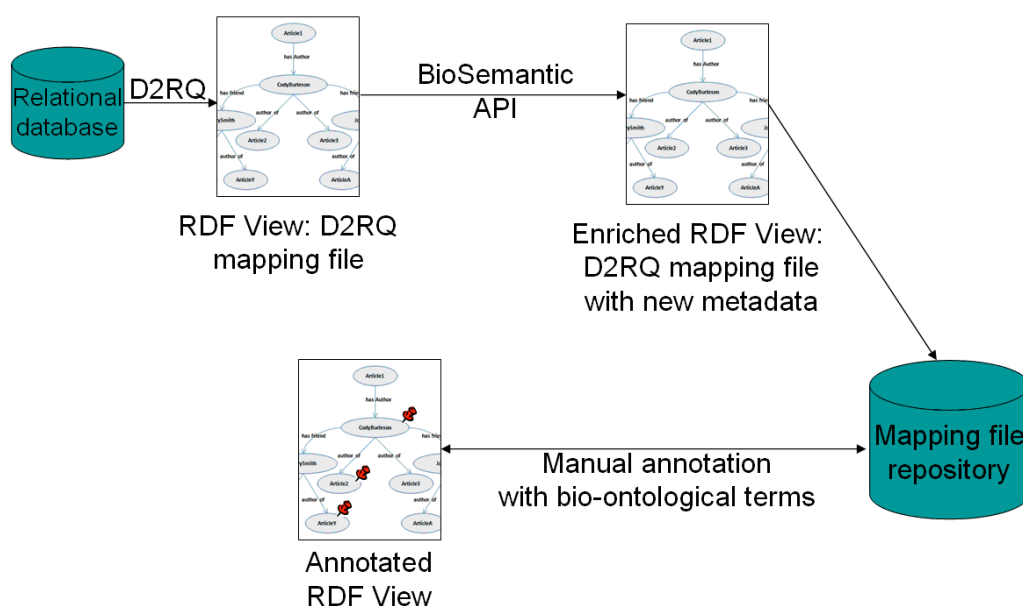


Fig. 1 - Generation and semi-automatic annotation of the RDF view.

3 Automatic generation of the Semantic Web Service

Semantic annotations are used to select inputs and outputs of a query. We are able to find a path in one RDF view by linking the inputs to the outputs. If such a path is found in the RDF view, it is used to create a SPARQL query. To automate the creation of SPARQL queries, we implement an algorithm that is a single-pair variant of the shortest-path algorithm. Given an input graph, a source node and a destination node, it returns a path linking the two nodes through the graph. We add conditions to our shortest-path algorithm according to the types of relationships between the nodes, which can be either of the following: (i) relationships corresponding to association tables; or (ii) relationships resulting from inheritance, aggregation or composition in an object-oriented conceptual model. These conditions correspond to the metadata added to the RDF view during the automatic semantic enrichment step by the BioSemantic API.

The Web Services developer selects the bio-ontological terms to be used as input/output (Fig. 2). All of the mapping files, which are stored in the mapping file repository, are automatically parsed to find a path linking the input and output ontological terms. If such a path is found, it is used to create a SPARQL query. The query is integrated into a semantic Web Service that is then registered in a Web Service registry, such as BioCatalogue.

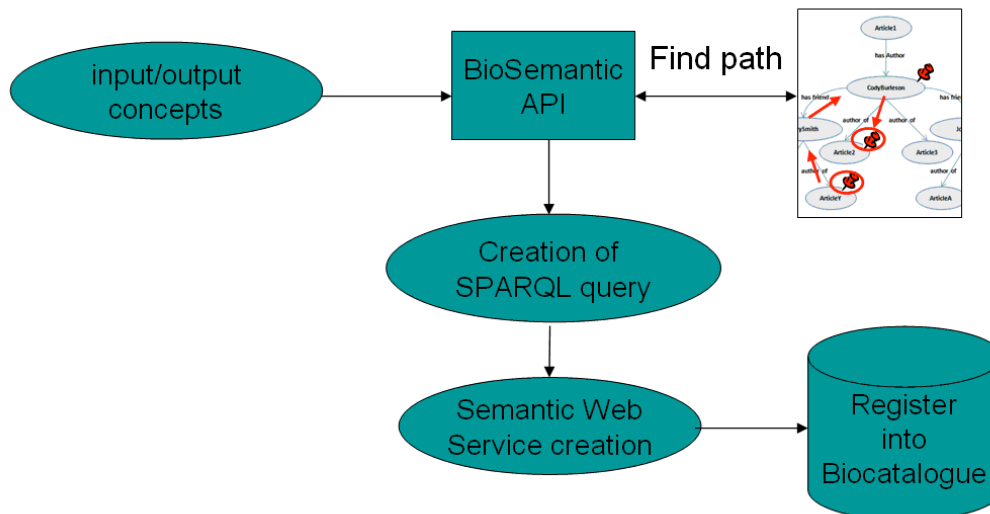


Fig. 2 - Automatic generation of Semantic Web Services.

Acknowledgements

We would like to acknowledge Isabelle Mougnot, Guilhem Sempere, and Frédéric de Lamotte for their assistance.

This work was supported by Region Languedoc-Roussillon and CIRAD.

References

1. Stein LD: Towards a cyberinfrastructure for the biological sciences: progress, visions and challenges. *Nat Rev Genet* 2008, 9:678–688.
2. Goble C, Stevens R: State of the nation in data integration for bioinformatics. *J Biomed Inform* 2008, 41:687–693.
3. Gessler D, Schiltz G, al.: SSWAP: A Simple Semantic Web Architecture and Protocol for semantic web services. *BMC Bioinformatics* 2009, 10:309.
4. Wilkinson M, McCarthy L, al.: SADI, SHARE, and the in silico scientific method. *BMC Bioinformatics* 2010, 11:S7.
5. The BioMoby Consortium: Interoperability with Moby 1.0—It's better than sharing your toothbrush! *Briefings in Bioinformatics* 2008.
6. Spanos D-E, Stavrou P, Mitrou N: Bringing Relational Databases into the Semantic Web: A Survey. *IOS Press* 2011.
7. Erling O, Mikhailov I: Mapping Relational Data to RDF in Virtuoso. 2006.
8. Miles A, Zhao J, Klyne G, White-Cooper H, Shotton D: OpenFlyData: An exemplar data web integrating gene expression data on the fruit fly *Drosophila melanogaster*. *Journal of Biomedical Informatics* 2010.
9. Cheung K-H, Yip KY, Smith A, deKnikker R, Masiar A, Gerstein M: YeastHub: a semantic web use case for integrating data in the life sciences domain. *Bioinformatics* 2005, 21:i85-i96.
10. Lam HYK, Marenco L, Shepherd GM, Miller PL, Cheung K-H: Using Web Ontology Language to Integrate Heterogeneous Databases in the Neurosciences. *AMIA Annu Symp Proc* 2006, 2006:464-468.
11. Rahayu JW, Chang E, al.: A methodology for transforming inheritance relationships in an object-oriented conceptual model to relational tables. *Information and Software Technology* 2000, 42:571-592.
12. Tirmizi S, Sequeda J, al.: Translating SQL Applications to the Semantic Web. In *Database and Expert Systems Applications*. 2008:450-464.