SHARE & the Semantic Web - This Time it's Personal!

Benjamin Vandervalk[1] , Luke McCarthy[1], and Mark Wilkinson[1]

[1] Heart + Lung Institute at St. Paul's Hospital, UBC, Vancouver, BC, Canada.
markw@illuminae.com

Abstract. Currently, in the Semantic Web in Healthcare and Life Sciences large ontologies representing a "consensus" world-view are selected, then data relevant to those ontologies is manually located and aggregated prior to reasoning. This approach presents challenges in addition to the real-world limitations of reasoning over such large-scale data: data critical to discovery in the life sciences must often be generated dynamically by analytical algorithms. Here we describe a prototype system – SHARE – that consumes ontologies and queries and automatically discovers, retrieves, and reasons over information relevant to that ontological world view by locating and executing Web Services. The SHARE system exhibits what we believe are crucial characteristics of the Semantic Web vision: relevant information is dynamically discovered and/or generated from distributed resources, and is interpreted, updated, or re-interpreted as the over-laid local ontology changes.

Keywords: RDF, OWL, Semantic Web, Semantic Web Services, SPARQL, Workflow, Workflow Orchestration, SADI, SHARE

1 Introduction

As Semantic Web technologies become more pervasive in bioinformatics, we must urgently define scalable ways to discover, analyse, and organize biological data in ways that encourage scientific exploration, discourse and disagreement, and in a manner compliant with the proposed Semantic Web "stack"[1]. The vision of the "stack" is that distributed, linked data is interpreted by logical reasoning, guided by ontological axioms. Unfortunately, today's Semantic Web in bioinformatics lacks many features that make this vision scalable to the size of the Web. For example, it is common to select an ontology representing a shared "consensus" world-view, then manually locate and aggregate relevant data into a local triple store, and then reason over that data and classify it into the ontology. Unfortunately, the size of typical bioinformatics datasets, combined with the fact that commonly-available OWL reasoners must load the entire dataset into memory before reasoning, render this approach infeasible without extensive manual pre-filtering. Perhaps an even bigger problem is that many important bioinformatics data exist only as the transient output from invocations of analytical tools (often accessed through Web Services) and this data must be pre-computed and converted before it is semantically accessible.

SHARE (the Semantic Health and Research Environment) is a mediator system that enables simultaneous querying of databases and analytical programs distributed across the Web, where resources are exposed as Services using the SADI (Semantic Automated Discovery and Integration) Semantic Web Service framework[2]. Together SADI+SHARE differ from other mediator systems in that every aspect of the system design utilizes Semantic Web standards. On the client-side, queries to SHARE are expressed in the SPARQL[3] query language, while on the server-side, SADI Services natively consume and generate RDF data. SADI Service input and output data-types are described using OWL[4], and Services are discovered and matched by the SHARE mediator using OWL reasoning. The properties added to a dataset by a SADI Service can be automatically determined by examining its input and output OWL Class definitions, and SADI provides a simple Web Service registry in which Services are indexed and discoverable based on these properties.

In contrast to typical SPARQL query engines, SHARE does not search an existing RDF dataset for a sub-graph with a specified triple-pattern structure. Instead, SHARE utilizes SADI by matching individual triple-patterns in the SPARQL query against SADI Service interface descriptions (in OWL) to discover services capable of generating those triple patterns; thus the RDF data required to answer any given SPARQL query is dynamically generated in response to the query being posed. Note that while distributed SPARQL endpoints may provide some of the query-relevant data, data may also be dynamically generated through execution of an application or algorithm since all resources are exposed as Web Services, thus making the nature of the underlying data resource opaque to the client.

We first introduce the methodology by which SHARE converts SPARQL queries into Web Service workflows. We then show how formal OWL Class definitions can similarly be converted into workflows, thus making it possible to dynamically discover OWL Individuals, compiled from globally distributed data resources, that are compliant with the OWL Classes in a given local or remote ontology.

2 Methodology

2.1 Conversion of SPARQL

SHARE utilizes properties in a SPARQL query (the predicates in the triple-patterns) as keys in a query against the SADI registry. Thus the process of discovery for any given query is relatively straightforward.: for each triple pattern (*s,p,o*) SHARE searches for services that have *p*, or the OWL inverse of *p*, as one of their generated properties. Consequently, at each step there are two cases:

Case 1: *s* is a concrete node (URI or literal), or a variable that is already bound to a set of candidate nodes. In this case, the triple pattern is resolved in the forward direction. First, the SADI registry is queried to find Services that are capable of generating the predicate in the triple pattern. For each Service discovered, the candidate nodes of *s* are compared, using an OWL reasoner, to the axioms in the input OWL Class definition for the discovered service. Matching nodes become rdf:typed according to the Service input class and are submitted for processing. (Optionally, SHARE itself can be used to dynamically resolve OWL Class membership as defined in §2.2 below.) If *o* has not been bound to any candidate nodes, the set of retrieved values for property *p* become the candidates of *o*.

Case 2: *s* has no candidate nodes and *o* is a concrete node (URI or literal) or a variable bound to a set of candidate nodes. In this case, the triple pattern is resolved as above, but in the reverse direction: instead of *p*, the OWL inverse of *p* is used to find Services. The retrieved values for inverse(*p*) become the bindings of *s*.

Between these two cases, it is frequently possible to automatically synthesize a Web Service workflow that collects all query-relevant data prior to SPARQL query resolution. It is currently not possible with SHARE to resolve triple patterns where both *s* and *o* are unbound, but SHARE attempts to order the query to avoid this case. We are experimenting to determine the optimal behaviour when *p*, the predicate itself, is unbound, but *s* and/or *o* are bound.

2.2 Conversion of OWL

When OWL Class definitions extend beyond mere declaration of a Class and include axioms that must be fulfilled in order to achieve class membership, those axioms can be used to drive Web Service discovery in a manner similar to that described above for SPARQL queries. SHARE decomposes the definition of the Class to determine what properties are required for Class membership and these properties (*p*) are used as keys in a SADI service look-up. For each discovered Service, candidate individuals matching the Service's input OWL Class are submitted for processing. Further, if a candidate individual does not conform to a Service's Input Class definition, the OWL axioms of that input Class are similarly decomposed and used for further Web Service discovery until all required data elements are present.

Clearly, this iterative process must halt, and these stop-points are either non-axiomatic Class definitions or axioms with strict value requirements. In bioinformatics both cases often involve database identifiers, so SHARE requires all database record URIs to be explicitly rdf:typed according to their source database. At present, SHARE is coded to recognize identifiers from the LSRN and bio2rdf database record naming schemes; URIs that adhere to either scheme will be automatically rdf:typed by SHARE and can be used for Web Service matchmaking, allowing most simple queries to be performed with no initial data whatsoever.

3 Demonstration

The example query below demonstrates SADI+SHARE functionality. This query can be executed at the CardioSHARE client website[5]. The query simply asks for instances of the *AtRiskPatient* OWL Class that appear in the RDF document referenced in the FROM clause. *AtRiskPatient* is defined in OWL as equivalent to a restriction on the *BMI* property where the value is greater than or equal to 25. SHARE examines this definition and queries SADI for a Service that can attach the *BMI* property, finding the Service at *http://sadiframework.org/examples/calculateBMI*. Instances of that Service's input OWL class in the patient data are identified and the

Service is invoked on those resources.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX patients: <http://biordf.net/cardioSHARE/patients.owl#>
PREFIX bmi: <http://sadiframework.org/examples/bmi.owl#>
SELECT ?patient ?bmi
FROM <http://biordf.net/cardioSHARE/patients.rdf>
WHERE {
    ?patient rdf:type patients:AtRiskPatient .
    ?patient bmi:BMI ?bmi
}
```

The Service performs the BMI calculation based on the patient's height and weight, the output RDF is stored in a transient triple store and a reasoner classifies the appropriate URIs as instances of *AtRiskPatient*. Finally, the SPARQL query is executed over the transient triple store. A more detailed walkthrough of the example query[6] and additional examples[5] can be found at the CardioSHARE website.

3 Conclusion

We believe that this is the first time relevant, distributed data has been dynamically discovered, generated and assembled in response to the assertion of an ontological class. Formally classifying data in terms of its properties allows the consequences of a particular definition to be easily explored. Further, the ontological classes thus created provide a means to share domain knowledge among researchers. The SADI+SHARE approach, we believe, gives significant impetus to begin defining "personal ontologies" – ontologies that reflect an *individual* rather than a shared world view, which can then be resolved over global data and analytical resources. The first step towards a personalized Semantic Web!

Acknowledgments. Heart + Stroke Foundation of BC and Yukon, Microsoft Research, The Canadian Institutes for Health Research, The Natural Sciences and Engineering Research Council of Canada, and CANARIE.

References

1. Description of W3C Technology Stack Illustration, http://www.w3.org/Consortium/techstack-desc.html
2. Wilkinson M.D., Vandervalk, B, McCarthy, L.: SADI Semantic Web Services – 'cause you can't always GET what you want! Proceedings of SWSIP 2009, Singapore. http://sadiframework.org/documentation/SADI_SWSIP09_personal.pdf
3. SPARQL query language for RDF, http://www.w3.org/TR/rdf-sparql-query/
4. OWL 2 Web Ontology Language Document Overview, http://www.w3.org/TR/owl2-overview/
5. CardioSHARE Demo, http://biordf.net/cardioSHARE/query
6. CardioSHARE Walkthrough, http://dev.biordf.net/cardioSHARE/bmi-walkthrough