Choosing between Axioms, Rules and Queries: Experiments in Semantic Integration Techniques

Aidan Boran[1], Ivan Bedini[1], Christopher J. Matheus[1]
Peter F. Patel-Schneider[2] and John Keeney[3]

[1] Bell Labs Ireland, Alcatel-Lucent, Dublin, Ireland,
{aidan.boran, ivan.bedini, chris.matheus}@alcatel-lucent.com
[2] Bell Labs, Alcatel-Lucent, Murray Hill, NJ, USA, pfps@research.bell-labs.com
[3]Trinity College Dublin, Ireland, john.keeney@cs.tcd.ie

Abstract. When using semantic technologies developers are frequently confused about which specific inferencing technique is best to use for a given problem. As an initial step towards identifying "best practices" for users of semantic technologies we are conducting a series of experiments to contrast the benefits and limitations of three approaches to inferring cross-data relationships: RDFS/OWL axioms, user-defined rules and SPARQL queries. At the highest level our aim is to identify which approaches provide the best (or acceptable) solutions in terms of memory use, cpu cycles and developer effort, given a variety of specific problem characteristics. In this paper we describe the three semantic approaches we are investigating, identify three broad problem areas in our initial focus and summarize some preliminary results.

Keywords: semantic integration, empirical experiments, axioms, rules, queries.

1 Introduction

The work described in this paper* is part of an on-going research effort to develop a methodological approach to Semantic Data Access (SDA) that will enable application programmers to more easily use semantic techniques to access, transform, query and reason about data residing in distributed systems. One aspect of this effort focuses on specific techniques for inferring relationships needed to integrate information across heterogeneous data sources. This paper discusses three semantic integration approaches - OWL axioms, user-defined rules and SPARQL queries – and outlines a series of empirical experiments being developed to evaluate their general applicability across a range of problem domains. The objective is to develop an understanding of the strengths and limitations of these approaches that can be turned into general design patterns for guidance on their use by application developers in realistic scenarios, *i.e.*, with a substantive number, but often not billions, of triples. Some initial results have been produced for experiments involving a "Smart Conference" scenario, which is used in this paper to provide an overview of our experimental approach and the objectives of our longer-term research involving empirical evaluation of semantic techniques.

* This work was partly funded by the Industrial Development Authority (IDA) Ireland and the Science Foundation Ireland via grant 08/SRC/I1403 (FAME).

2 Three Integration Techniques

The simplest of the data integration techniques we cover involves using RDF and OWL axioms by themselves. As will be demonstrated below, this can be achieved using ontological constructs such as rdfs:subPropertyOf, owl:sameAs, *etc*. without resorting to the use of user-defined rules or SPARQL queries. One advantage of this approach is that at runtime the integration process can be accomplished using commonly available OWL reasoners alone. A major drawback here is that the representational limits of OWL may not permit the formulation of the relationships needed for the intended data integration problem.

Our second integration technique is to use a rule language (*e.g.*, SWRL, RIF) to define specific rules to perform the desired integration. This approach permits more complex data integration than OWL alone can afford, such as is often required for joining property-values across multiple objects (*i.e.*, relating an object's property value to a property value of another object). This approach can be used in isolation or along side OWL reasoning, but for the work described here we focus on the use of rules without an explicit OWL reasoning system.

The final technique investigated involves the use of SPARQL queries to extract and construct the desired data integration from an RDF datastore. Using a combination of query match clauses (*i.e.*, the SPARQL query WHERE statement) and output construction clauses (*i.e.*, the CONSTRUCT statement) can be shown to be equivalent to a rule-based approach. This technique may also be used in conjunction with an OWL/RDF reasoning system (whereby a full or partial materialization of the inferable triples in the knowledge base is pre-stored in the RDF datastore), but for this work it is assumed that the datastore only contains the original (un-materialized) data.

3 Three Problem Domains

We have identified three broad problem domains in which to carry out specific experiments to empirically explore the use of the three integration techniques identified above on problems with different data characteristics. The first domain involves social network information in the context of a location-based service for a "Smart Conference" scenario; image a conference service based around onsite registration using smartphones that automatically identifies and locates your acquaintances who are also in attendance at the conference. The data sources for this problem include a large Friend-of-a-Friend (FOAF) dataset and a semantically enhanced version of the DBLP Computer Science Bibliography (DBLP++). The integration problem is to identify "acquaintances" of a conference attendee, which are defined as either people the attendee foaf:knows or co-authors of papers the attendee has written as indicated by DBLP++. For this domain a series of experiments have been conducted and some of the high-level results are summarized below.

The second problem domain is that of wireless telecommunication networks, with an initial focus on Femto Cell networks owing to the recent establishment of a fully functional Femto Testbed at Bell Labs Ireland. The scenario under development for this domain involves the use of network information (*e.g.*, configuration, performance, customer data) in conjunction with location (*e.g.*, FourSquare, Google Latitude, ALU Presence Server) and context (*e.g.*, smartphone sensors, office

equipment proximity, local services) information to provide "Smart Communication" services to assist in efficiently connecting with people, information and networked objects. The integration problem will involve relating dynamic information retrieved from the network with social network and personal smartphone based information. In this domain, most of the information (in particular all of the network data) will need to be automatically "lifted" into a semantic layer using, for example, DERI's XSPARQL[†] engine. We hope to have some preliminary results to share by the time of the OWLED Workshop.

The final problem domain we intend to begin working with later this year is that of sensor networks. This domain is the least well defined as we are waiting for initial developments to materialize in a sister project focused on wide-area Wireless Sensor Networks (WSN). Our main interests will be in semantically describing the sensors and the networks that they comprise as well as providing the means to semantically access and integrate the data that a federation of WSNs would collectively gather.

4 Smart Conference Experiments

A series of initial experiments carried out in the Smart Conference domain are very briefly described here; additional details can be found in a paper submitted to the 2011 IEEE/WIC/ACM International Conference on Web Intelligence.[‡]

Data and Tools. For this work FOAF datasets where combined from a number of sources resulting in a dataset with 19,054,802 triples, 340,430 foaf:Person individuals identified by name, and 511,745 occurrences of the foaf:knows relationship. The DBLP dataset used in this work contains 36,439,753 triples, 850,149 person individuals identified by name, and zero occurrences of the foaf:knows relationship. The semantic tools used in these experiments included Pellet for OWL reasoning, Jena for rule processing and Jena TDB/ARQ for SPARQL querying.

Integration Problem. To identify "acquaintances" there is a key piece of *implicit* information within the DBLP data that we would like to make explicit and integrate with the FOAF data. Specifically, in the case of publications having more than one author (foaf:maker), we can infer that the co-authors are "acquaintances". As neither dataset explicitly defines a mechanism to capture a symmetric "acquaintance" relationship between two conference attendees, we introduce an additional ontology where we define a new symmetric OWL object property, "sda:acquaintance", for our scenario. The challenge then becomes that of mapping the various relationships defined in the FOAF and DBLP datasets onto this sda:acquaintance relationship.

OWL Axioms. Mapping the foaf:knows to sda:acquaintance in the FOAF dataset using OWL axioms was done by simply asserting sda:acquaintance as an rdfs:subPropertyOf foaf:knows. In the DBLP dataset mapping co-authorship to a "knows" relationship involved two subparts: transforming the multiple foaf:maker relationships between a document and its authors into a co-maker (co-author) relationship, and deriving the sda:acquaintance relationship from the co-maker/co-authorship relationship. The co-maker/co-authorship relationship was achieved by defining an OWL property chain for sda:co-maker using foaf:maker and the inverse

[†] http://xsparql.deri.org/

[‡] http://liris.cnrs.fr/~wi-iat11/WI_2011/

of foaf:maker: SubPropertyOf(ObjectPropertyChain(foaf:maker ObjectInverseOf(foaf:maker)) sda:co-maker). Deriving sda:acquaintance from co-authorship was achieved by marking the sda:co-maker property chain as a sub-property of the symmetric sda:acquaintance property.

User-defined Rules. For the FOAF dataset a simple Jena rule was defined to effectively make sda:acquaintance into a subproperty of foaf:knows:

```
[FoafRule: (?Person2 foaf:knows ?Person1) ->
            (?Person1 sda:acquaintance ?Person2)
            (?Person2 sda:acquaintance ?Person1) ]
```

For the DBLP dataset we infer the colleague relationship using this rule:

```
[AuthorRule: (?Document foaf:maker ?Person1)
             (?Document foaf:maker ?Person2)  ->
             (?Person1 sda:acquaintance ?Person2)
             (?Person2 sda:acquaintance ?Person1)]
```

Caveat: with FOAF it is common for a single person to be represented by a number of different foaf:Person instances inter-related by the owl:sameAs property. Without OWL inference, properties for each individual will not be materialized for all other relevant individuals. For this reason we extended the FOAF ruleset as follows:

```
[SameAs1: (?x owl:sameAs ?y)  (?x ?p ?o ) -> (?y ?p ?o)]
[SameAs2: (?x owl:sameAs ?y)  (?s ?p ?x )  -> (?s ?p ?y)]
```

SPARQL Queries. For the FOAF dataset SPARQL queries were constructed at runtime to work for a specific individual (in this code example, *John Doe*):

```
WHERE     {?Person1 foaf:name "John Doe".
           ?Friend foaf:knows ?Person1.
           ?Friend foaf:name ?Friendname}
CONSTRUCT {?Person1 sda:acquaintance ?Friend.
           ?Friend sda:acquaintance ?Person1.
           ?Friend foaf:name ?Friendname.
           ?Person1 foaf:name "John Doe".}
```

Unfortunately, this query needs to be extended to return not just the foaf:Person instances but all of their owl:sameAs instances. It is also necessary for the CONSTRUCT clause to fully materialize all sda:acquaintance relationships, and their inverses, for all returned foaf:Person instances and their optional owl:sameAs instances. This makes the FOAF query much more complicated than represented here.

A slightly less complex SPARQL query was required for the DBLP dataset but its details are not presented here due to space concerns.

Results. Performance wise, SPARQL queries proved most efficient in terms of compute time and memory requirements (*i.e.*, triples generated) while being able to handle our largest datasets; but they were also difficult to construct, particularly given the need to implement within the queries inferences built into OWL reasoners (*e.g.*, owl:sameAs). User-defined rules suffer from the same problems and were arguably even more challenging to develop than queries; this was particularly true in our initial use of SWRL where its verbose XML syntax added to rule complexity. OWL axioms were by far the easiest to construct and performed reasonably well except on the largest datasets, which could not be handled due to memory constraints.

For experienced OWL practitioners these results may not appear surprising, but for those newly exposed to semantic technologies this type of direct comparison of semantic techniques using readily available tools will hopefully provide instructive insight. Collective results from our planned future experiments should help further reveal how these semantic techniques perform relative to each other under a wider range of problem characteristics.