Provenance and Search Issues in RDF Data Warehouse

Li Ding

Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180
ding@ksl.stanford.edu

Effective data management infrastructures are needed by applications to utilize distributed Semantic Web data. RDF databases are available today as large triple stores that manage RDF triples and support certain Semantic Web inference; however, triple stores are not suitable for information integration applications that require intensive knowledge provenance support for assuring data quality, confidentiality and explaining workflow. Unlike an RDF database, an *RDF data warehouse* tracks knowledge provenance and helps users to locate data. In what follows, we investigate two design issues.

Tracking knowledge provenance. Figure 1 shows three types of typical knowledge provenance events in an RDF data warehouse instance *DR*. Given the description of a source, DR *retrieves* a snapshot (cached document) from the source in its original format (usually free-text). DR then *extracts or parses* an RDF graph from the cached version and names it with an IRI. Later, users may *derive* new named graphs from existing named graphs in DR using OWL inference or user-defined functions.

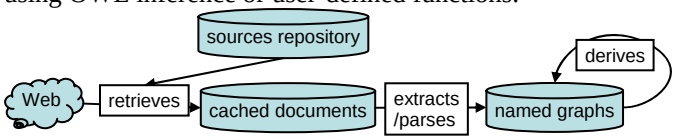

Figure 1 knowledge provenance in RDF data repository

Named graphs [1] are the building blocks of an RDF data warehouse. Some graphs are directly parsed or extracted from cached revisions of web pages, and they are uniquely identifiable by their source URL and creation date time. By maintaining graphs for every revision, a data warehouse keeps a full revision history and eliminates *delete* and *update* operations. The other graphs in a RDF data warehouse are derived from existing graphs.

The three types of knowledge provenance events share common structure: each event derives the resulting data by conducting certain computations on a set of input data. We differentiate them using a small taxonomy: a *SourceUsage* logs an event when a piece of information is obtained from a source; an *InformationUsage* logs an event when a piece of information is derived from the other information; and a *GraphUsage* is a special case of *InformationUsage* and focuses on named graphs. Besides *GraphUsage* relations, named graphs may also be linked via versioning relations.

Although many computations involved in knowledge provenance events are hard to declaratively represent, the advance of SPARQL[1] enables convenient descriptions for many 'derives' activities. SPARQL queries can annotate simple activities such as graph-copy and graph-import, and they also help annotate complex graph transformation/ derivation such semantic mapping. For example,

- Schema mappings are essentially SPARQL queries that create new graphs using mapping information.

```
CONSTRUCT   { ?x vcard:FN ?name }
WHERE       { ?x foaf:name ?name }
```

- Instance-reference mappings can be derived by SPARQL queries equipped with customized value test functions and graph pattern specifications.

```
CONSTRUCT  { ?x rel:same_by_rule1 ?y }
 WHERE { ?x foaf:name ?n_x.   ?y foaf:name ?n_y.
                FILTER (fn:name_rule1 (?n_x, ?n_y))  }
```

Data Access Interface. When a RDF data warehouse has stored huge amount of RDF graphs, users may need effective data access interfaces.

- Word-occurrence search provided by conventional full-text search engines is simple and intuitive to many users. Such search can be enhanced by semantic query expansion (e.g. using WordNet synonyms).
- Index-based browsing techniques (e.g. alphabetical, chronological, categorical, and geographical index) partitions the data space from various perspectives and help organize and present all indexed data.
- SPARQL can be used in RDF data warehouses, but scalability and efficiency issues remain.
- Faceted query is natural to RDFS/OWL instances, and versioning, semantic-mapping, and content-duplication relations should be addressed in interface design.
- Besides hyperlinks, users may surf to information using provenance knowledge [2] and social network [3].
- Statistical summary and analysis of stored data are also important to enhance users' data access experience.

We have shown preliminary results concerning two design issues for RDF data warehouses. Future work will focus on implementation details using our past experiences on Inference Web[4], Swoogle[2], and social networks[3].

References

[1] Carrol, J., et al., Named Graphs, Provenance and Trust, WWW, 2005
[2] Ding, L., et al., *Finding and Ranking Knowledge on the Semantic Web*, ISWC, 2005
[3] Ding, L., et al., *Trust Based Knowledge Outsourcing for Semantic Web Agents*, International Conference on Web Intelligence, 2003.
[4] McGuinness, D. et al., *Explanation Interfaces for the Semantic Web: Issues and Models*. Semantic Web User Interaction Workshop, 2006

Bio: Li Ding is a postdoctoral researcher in the department of computer science, Rensselaer Polytechnic Institute. Prior to that, he was a postdoc fellow in Knowledge System, AI Lab (KSL) at Stanford University, and he received his PhD in Computer Science from UMBC in 2006. He is interest in building practical and accountable Semantic Web with emphasis on knowledge provenance, data access interface, and information integration.

[1] http://www.w3.org/TR/rdf-sparql-query/