# XML Perspectives on RDF Querying: Towards integrated Access to Data and Metadata on the Web

Tim Furche, François Bry, Oliver Bolzer
Institute for Informatics, University of Munich
http://www.pms.ifi.lmu.de

**Abstract**

The integral processing of data and metadata is starting to get recognized as a central challenge for the next decade (e.g. in Pat Selinger's ICDE 2005 Keynote) not only as part of realizing the Semantic Web vision, but also on a smaller scale as part of the next generation of desktop data management (cf. Apple's Spotlight and Microsoft's WinFS). In this article, we focus on metadata represented in the W3C's RDF formalism. We illustrate first steps towards integrating access to RDF metadata and access to standard Web data in XML format. For this, two XML views over RDF data are expressed in the query language Xcerpt and discussed. These views illustrate two different approaches for integrating RDF metadata processing and current data processing techniques.

## 1 Introduction

The "Semantic Web" is an endeavor widely publicized in [1], envisioning the current Web, which consists of (X)HTML and documents in other XML formats to be extended by metadata specifying the meaning of these documents in forms usable by both human beings and computers.

The integral processing of data and metadata is starting to get recognized as a central challenge for the next decade (e.g. in Pat Selinger's ICDE 2005 Keynote) not only as part of realizing the Semantic Web vision, but also on a smaller scale as part of the next generation of desktop data management (cf. Apple's Spotlight and Microsoft's WinFS that share the aim to extend current file storage and desktop search with extensive metadata facilities).

In the (Semantic) Web context, a number of formalisms have been proposed for representing metadata, in particular RDF, Topic Maps, and OWL. We concentrate on RDF as the most widely used. This article illustrates first steps towards integrating access to standard Web data in XML format and RDF metadata: First, as argued above, integrated access to standard Web data in XML and metadata in RDF is essential. A framework to access RDF data through XML views is proposed. Second, we argue that the currently predominant treatment of RDF data as flat triples is, although easy to comprehend, not the only and often not the best way of considering RDF data. Rather, a view of the RDF data directly as a graph is not only natural and closer to the RDF data model and allows for easy expression of graph patterns using much the same constructs as for navigating in XML data. This is particularly evident in face of incomplete information about the precise graph structure.

The proposed framework is realized by rules in the XML query language Xcerpt that allow (a) the easy conversion between the two views on RDF and (b) the "serialization transparent" querying of RDF, i.e., the querying of RDF in many of the over a dozen serialization formats for RDF proposed in recent years.

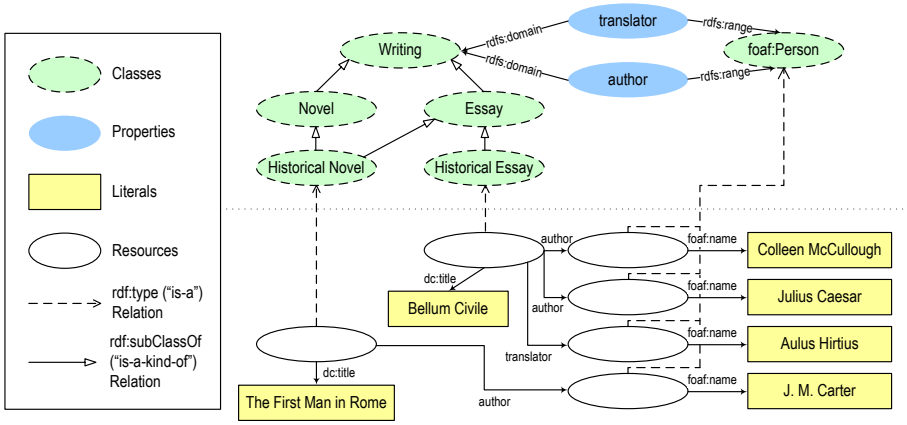


Figure 1: Sample Data: representation as a (simplified) RDF graph.

# 2 Preliminaries

## 2.1 RDF and RDF Schema: Metadata Representation in the Semantic Web

*RDF* [6] data is sets of "triples" or "statements" of the form (*Subject*, *Property*, *Object*). RDF data is commonly seen as a directed graph, whose nodes correspond to a statement's subject and object and whose arcs correspond to a statement's property (thus relating a subject with an object). Nodes (i.e. subjects and objects) are labeled by either (1) URIs describing (Web) resources, or (2) literals (i.e. scalar data such as strings or numbers), or (3) are unlabeled, being so-called anonymous or "blank nodes". Blank nodes are commonly used to group or "aggregate" properties. Edges (i.e. Properties) are always labeled by URIs indicating the type of relation between its subject and object.

*RDFS* allows one to define so-called "RDF Schemas" or "ontologies", similar to object-oriented data models. Based on an RDFS, "inference rules" can be specified, for instance the transitivity of the class hierarchy, or the type of an untyped resource that has a property associated with a known domain.

RDF can be *serialized* in various formats, the most frequent being XML. Early approaches to RDF serialisation have raised considerable criticism due to their complexity. As a consequence, a surprisingly large number of RDF serialisation have been proposed, cf. [3].

Figure 1 shows the running example for this article, a (simplified) representation of an RDF graph as used, e.g., in a book recommender system.

## 2.2 Xcerpt, a versatile Web Query Language

Xcerpt [9] is a query language designed after principles given in [4] for querying both data on the "standard Web" (e.g., XML and HTML data) and data on the Semantic Web (e.g., RDF, Topic Maps, etc. data).

Xcerpt is "data versatile", i.e. a same Xcerpt query can access and generate, as answers, data in different Web formats. Xcerpt is "strongly answer-closed", i.e. it not only gives rise to construct answers in the same data formats as the data queries, but also to include in a query program data generated by this same query program. Xcerpt's queries are pattern-based and give rise to incompletely specify the data to retrieve by (1) not explicitly specifying all children of an element, (2) specifying descendant elements at indefinite depths (restrictions in the form of regular path expressions being possible), and (3) specifying optional query parts. Xcerpt's evaluation of incomplete queries is based on a novel form algorithm called "simulation unification". Xcerpt's processing of XML documents is graph-oriented, i.e., aware of the reference

mechanisms (e.g., ID/IDREF attributes and links) of XML. Xcerpt is rule-based: An Xcerpt rule expresses how data queried can be re-assembled into new data items.

# 3 Two Perspectives on RDF

This section presents two different perspectives on RDF: (1) a flat, almost relational view and (2) a graph view reminiscent of semi-structured data. These perspectives are compared briefly with some existing approaches for RDF querying.

To illustrate these two perspectives, the selection query "Select all *Essays* together with their *authors* (i.e. author URIs and corresponding names)" is used against the data of Figure 1.

## 3.1 RDF Triples: A Flat, Relational View

The following Xcerpt program expresses the above query on a triple view of the RDF data:

```
1  DECLARE ns-prefix rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   DECLARE ns-prefix books = "http://example.org/books#"
3  GOAL
     result [
5      all essay [
         id [ var Essay ],
7        all author [
           id [ var Author ],
9          all name [ var AuthorName ]
     ]  ]  ]
11 FROM
     and{ RDFS-TRIPLE [
13       var Essay, rdf:type{{}}, books:Essay{{}}            ],
       RDF-TRIPLE [
15       var Essay, books:author{{}}, var Author             ],
       RDF-TRIPLE [
17       var Author, books:authorName{{}}, var AuthorName    ]  }
   END
```

The query pattern (between `FROM` and `END`) is a conjunction of queries against the RDF triples represented in the predicate `RDF-TRIPLE` using the prefixes declared in line 1 and 2. Notice that the first conjunct actually uses `RDFS-TRIPLE`. This view of the RDF data contains all basic triples plus the ones entailed by the RDFS semantics (cf. [2] for a detailed description). Using `RDFS-TRIPLE` instead of `RDF-TRIPLE` ensures that also resources actually classified in a sub-class of `books:Essay` are returned.

In the construct pattern (between `GOAL` and `FROM`), one of the strengths of combining XML and RDF querying in Xcerpt is shown: Following the W3C's requirements for an RDF data access language, yet in contrast to most other RDF query languages, it is possible to construct arbitrary XML: E.g., here, a list of all essays with their authors grouped inside is constructed.

Except for the construction of arbitrary XML, a similar (triple) view of RDF is taken in most of the current RDF query languages, most notably in RDQL and the W3C's SPARQL [7], and also in [8], an approach for querying RDF with XQuery: A query is composed of conjunctions (and in some languages including our proposal disjunctions) of "triple patterns", i.e., triples with variables indicating queried data. Using multiple occurrences of same variables more complex conditions can be expressed, e.g., for traversing paths in the RDF data or even for restricting a resource using several of its properties. While familiar from SQL, this style leads for RDF data to hard-to-read and lengthy queries that also pose problems for evaluation (cf., e.g., [5]).

The previous observations lead us to an alternative view of RDF that is both closer to its actual data model and can make better use of the advanced features of an XML query language such as the traversal of arbitrary length paths in tree or graph data.

### 3.2 RDF Graph: A Semi-structured View

For this view of RDF, Xcerpt's treatment of XML as graph data is an advantage over XML query languages such as XPath or XQuery, which consider XML as strictly tree shaped, providing no direct support for (ID/IDREF or similar) links in the data model. Although there have been proposals for slicing an (acyclic) RDF graph into trees for processing them with XSLT or XQuery (e.g., [11]), these approaches invariantly suffer (a) from choosing an appropriate slicing and (b) from the (in general) exponential blow-up of the tree view of an acyclic RDF graph.

In Xcerpt, a graph view of RDF is rather natural as the following Xcerpt program expressing the same query as above, but on the graph instead of the triple view, demonstrates:

```
  ... % prefixes and construction identical to above query
FROM
  RDFS-GRAPH {{
    var Essay {{
      rdf:type {{ books:Essay {{ }} }},
      books:author {{
        var Author {{ books:name {{ var AuthorName }} }}
      }}
    }} }}
END
```

The RDF graph view is represented in the `RDF-GRAPH` predicate. Here, the `RDFS-GRAPH` view is used that extends `RDF-GRAPH` as `RDFS-TRIPLE` extends `RDF-TRIPLE`. Triples are represented similar to striped RDF/XML: each resource is a direct child element in `RDF-GRAPH` with a sub-element for each statement with that resource as object. The sub-element is labeled with the URI of the predicate and contains the object of the statement. As Xcerpt's data model is a rooted *graph* this can be represented without duplication of resources.

In contrast to the previous query against the RDF triple view, no conjunction is used but rather a nested pattern that naturally reflects the structure of the RDF graph. The more complex a query gets, the more evident the advantage of the graph view becomes: instead of having to use multiple occurrences of same variables for relating parts of the query, that relation is represented in the structure of the query itself (represented in the textual version of the query shown above by nesting and indentation).

Path traversals of arbitrary length can be expressed using traversal operators such as descendant. E.g., to find all subclasses of a given class one can use Xcerpt's qualified descendant `desc(rdfs:subClassOf<rdfs:Class)*` that is similar to regular path expressions or conditional XPath. Similarly, other constructs for querying XML data with incomplete information about the structure of the queried data can be used for RDF as well.

Considering the efficient evaluation of queries against such a graph view of RDF data, there are results on the efficient evaluation of queries against graph-shaped semi-structured data, cf. [10]. Ongoing work by the authors targets efficient evaluation methods for implementing Xcerpt queries against graph-shaped data. We believe it likely that at least for some interesting subsets of Xcerpt queries efficient evaluation methods against graph-shaped data can be found.

#### 3.2.1 A "Retrospective" View: From Triples To Graphs

A final observation on the graph view is that it can not only be implemented directly on the different RDF serializations (just as the triple view) but also on top of the triple view, as shown in the following rule:

```
CONSTRUCT
  RDF-GRAPH {
    all var Subject @ var Subject:var SubjectType {
      all optional var Predicate {  ^var Object   },
      all optional var Predicate {  var Literal   }
    } }
FROM
  or{
```

```
    RDF-TRIPLE[
10    var Subject,  var Predicate:uri{},
      optional var Literal as literal{{}},
12    optional var Object {{}} where { var Object != 'literal'}
    ],
14  RDF-TRIPLE[
      /.*/:/.*/{{}},  /.*/:/.*/{{}},  var Subject{{}}
16  ]  }
END
```

Notice the use of the `optional` keyword in lines 11 and 12. This indicates that the contained part of the pattern does not have to occur in the data, but if it does occur the contained variables are bound appropriately. In lines 3 and 4 the actual graph structure is constructed: by using the operators `@` and `^` a (possibly cyclic) link can be constructed.

### 3.2.2 Serialisation Transparency

Aside of providing the above discussed two views on RDF, Xcerpt's rules are also convenient for making the language "serialisation transparent". For each RDF serialisation, a set of rules expresses a translation from or into that serialisation. They can be found in [2], similar functions for parsing RDF/XML in XQuery are described in [8].

## 4 Conclusion and Outlook

In this article, a brief overview of a framework for RDF querying in the XML query language Xcerpt is presented highlighting in particular the need for reconsideration of the triple view as the only perspective on RDF available in the established RDF query languages. We believe that a richer view of RDF more akin to XML data with graph-shape not only makes the integration of data and metadata easier but also leads in many cases to more succinct queries without sacrificing efficiency.

**Acknowledgments.** This research has been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. `http://rewerse.net`).

## References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.

[2] O. Bolzer. Towards Data-Integration on the Semantic Web: Querying RDF with Xcerpt. Diplomarbeit/Master thesis, University of Munich, 2005.

[3] F. Bry, T. Furche, L. Badea, C. Koch, S. Schaffert, and S. Berger. Identification of Design Principles for a (Semantic) Web Query Language. Deliverable I4-D1, REWERSE, 2004.

[4] F. Bry, T. Furche, L. Badea, C. Koch, S. Schaffert, and S. Berger. Querying the Web Reconsidered: Design Principles for Versatile Web Query Languages. *J. of Semantic Web and Inf. Sys.*, 1(2), 2005.

[5] E. Hung, Y. Deng, and V. S. Subrahmanian. RDF Aggregate Queries and Views. In *ICDE*, 2005.

[6] G. Klyne, J. Carroll, and B. McBride. *Resource Description Framework (RDF).* W3C, 2004.

[7] E. Prud'hommeaux and A. Seaborne. *SPARQL Query Language for RDF.* W3C, 2005.

[8] J. Robie. The Syntactic Web: Syntax and Semantics on the Web. In *XML*, 2001.

[9] S. Schaffert and F. Bry. Querying the Web Reconsidered: A Practical Introduction to Xcerpt. In *Extreme Markup Languages*, 2004.

[10] R. Schenkel, A. Theobald, and G. Weikum. HOPI: An Efficient Connection Index for Complex XML Document Collections. In *EDBT*, 2004.

[11] N. Walsh. RDF Twig: Accessing RDF Graphs in XSLT. In *Extreme Markup Languages*, 2003.