An O-Telos provider peer for the RDF-based Edutella P2P-network

Martin Wolpers, Wolfgang Nejdl, Ingo Brunkhorst
Learning Lab Lower Saxony,
University of Hannover,
Expo Plaza 1, 30359 Hannover, Germany,
email: wolpers,nejdl,brunkhorst@learninglab.de

Abstract. The open source project Edutella is a Peer-to-Peer (P2P) network for learning materials. It builds upon meta data standards defined for the WWW and aims to provide a RDF-based meta data infrastructure for P2P applications. In this paper we discuss a prototype provider peer for the Edutella network, which provides advanced reasoning peers for RDF data. This prototype peer provides storage and querying services for which it uses the ConceptBase database with its powerful reasoning mechanism as storage and query facility. ConceptBase implements the meta data language O-Telos while Edutella uses RDF(S). Based on previous work in this area we present a mechanism to translate the property-centered meta modeling language RDF to the object-centered meta modeling language O-Telos. Thus we are able to provide a Datalog¬ based reasoning facility for the P2P Edutella network.

As the Edutella network also uses the query exchange language RDF-QEL we also describe in detail the peer's query service. It translates RDF-QEL queries to O-Telos queries and the respective O-Telos answers to RDF-QEL answers.

1 Introduction

Within the last year we have been working on a project towards implementing an extendable open source meta data-based peer-to-peer infrastructure called Edutella for the exchange of distributed resources. Resources initially considered are learning materials but the infrastructure is agnostic to the specific type of resources to be exchanged. In our case we have large collections of learning material and we are working on annotating this material with RDF [7] and RDFS [1]. This will enable us to use the Edutella network services like storage, exchange, query and processing services which are based on RDF(S).

In its current state the Edutella network consists of client- and provider-peers. A client-peer poses queries to the Edutella network and displays the answers returned from the network. The provider-peers receive the queries from the network and answer them if possible. Thus a peer must realize a storage and query service for the Edutella network in order to function as a provider-peer. For a complete description of Edutella see the Edutella White Paper [10] and the projects' home page [3].

In this paper we describe a provider-peer and its services for the Edutella network, which provides extended reasoning capabilities. The provider-peer uses the ConceptBase database [5] as a repository for storing meta data. ConceptBase is a deductive object-oriented database manager that implements the meta data representation language O-Telos [6] [8]. In contrast to RDF, O-Telos is an object-oriented meta-modeling language that provides facilities for unrestricted meta modeling levels. Furthermore the ConceptBase database implements a powerful query and reasoning (rules and constraints) mechanism based on Datalog¬ which we want to make accessible to the Edutella network.

The provider-peer provides two basic services which we will present in this paper. On one hand it is designed to store RDF(S) data in the ConceptBase repository; the storage service. The data represented in RDF(S) is translated to O-Telos which is used by the repository. The translation uses the RDF-O-Telos axioms as described in [9].

On the other hand the provider-peer serves as a query interface to the RDF data stored in the ConceptBase repository; the query service. Queries formulated in the RDF exchange language RDF-QEL [10] are posed to the peer that translates them into O-Telos queries. The O-Telos queries than are answered by the ConceptBase database. The peer translates the ConceptBase answers from O-Telos into RDF and returns these RDF statements to the network.

Combining the storage and query service with ConceptBases' Datalog¬ reasoning facilities enables us to reason about the meta data used in the Edutella network.

This paper is structured as follows: the next chapter describes how RDF(s) data is stored in ConceptBase. With the help of a simple example our translation algorithm is explained in more detail. The third chapter deals with the querying facility provided by this peer. We explain how we are able to use the advanced querying facilities of ConceptBase for querying RDF(S) data. The fourth chapter describes briefly the technical issues concerned with this provider peer. A short summary closes the paper.

2 Storing RDF meta data in ConceptBase

The resources exchanged in the Edutella network are basically learning materials, each learning object is described by RDF(S) data. In this paper we describe a way to provide an Edutella peer to store and retrieve RDF(S) data. The example used throughout this paper is a small database containing the RDF description of some books. This database is taken from the Edutella [3] examples library. A part of this database is represented in figure 1.

In order to store the RDF meta data in the ConceptBase database the peer has to translate RDF to O-Telos. O-Telos itself is an object-

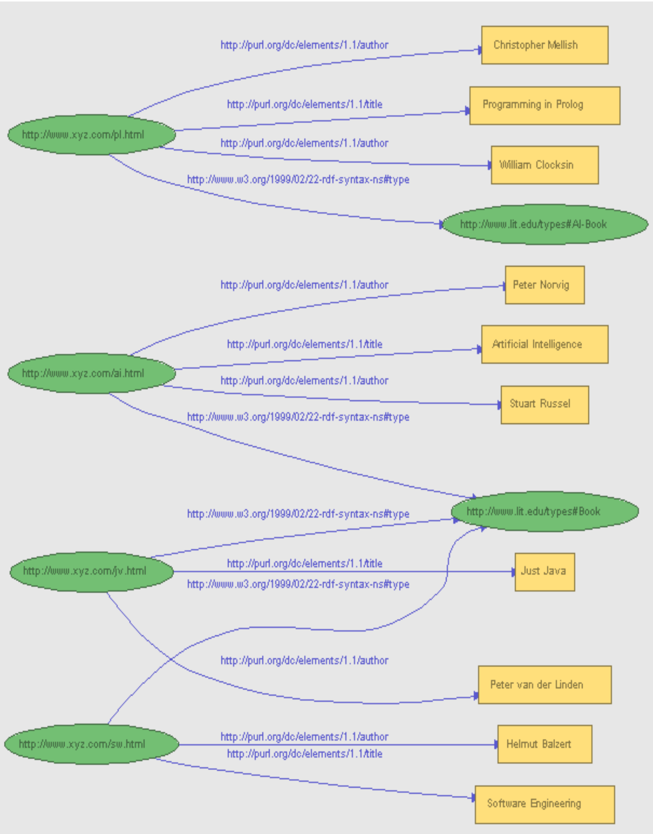

Figure 1. RDF graph of the example book database

oriented modeling language defined by 32 axioms. In order to translate RDF to O-Telos it is necessary to modify and eventually extend RDF(S). Therefore we have developed a translation based on the axioms from O-Telos-RDF. For a description of O-Telos-RDF and its defining axioms please see [9].

The algorithm ensures that RDF(S) data is translated without any loss of information. At first we are changing the representation from XML serialization to its representation as triple statements. Each triple is provided with an unique id. Furthermore the quadruples built are eventually supplemented with additional quadruples capturing further facts of the actual namespace to fulfill the axioms of O-Telos and O-Telos-RDF. The sum of compound and generated quadruples form the O-Telos database where each quadruple represents an O-Telos statement.

An O-Telos quadruple statement has the form (sid,subject,literal,object). Sid, subject and object are unique statement identifiers. Sid identifies the statement that connects the subject statement with the object statement using the predicate literal. Sid and subject can hold the same identifier only if sid, subject and object are the same identifier, so that the statement represents an individual with the literal as name. O-Telos declares several predefined literals like the In literal stating instantiation, the IsA literal stating inheritance, etc.

As an example for the translation process we will focus on just one RDF resource description. The translation of the other resource descriptions of the example book database follows basically the same procedure varying in small details only. The RDF statements for the example resource `http://www.xyz.com/jv.html` describe a book with its properties “Just Java” of type dc:title and “Peter van der Linden” of type dc:author. The resource itself is of the rdf:type `http://www.lit.edu/ types#Book`.

The RDF-triple representation of this resource description is given in table 1.

Nr.	triple
1	http://www.xyz.com/jv.html, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.lit.edu/types#Book
2	http://www.xyz.com/jv.html, http://purl.org/dc/elements/1.1/author, ”Peter van der Linden”
3	http://www.xyz.com/jv.html, http://purl.org/dc/elements/1.1/title, ”Just Java”

Table 1. RDF triples of the book ”Just Java”

Nr.	Quadruple
1	sid1, sid1, http___www_xyz_com_jv_html, sid1
2	sid2, sid1, In, #Book
3	sid3, sid1, In, #DCElements11
4	sid4, sid1, title1, ”Just Java”
5	sid5, sid4, In, #DCElements11.title
6	sid6, sid1, author1, ”Peter van der Linden”
7	sid7, sid6, In, #DCElements11.author
8	sid8, sid1, namespace1, ”http://www.xyz.com/”
9	sid9, sid8, In, #Book.namespace

Table 2. O-Telos quadruples of the book ”Just Java”

The O-Telos quadruples that describe the example resource `http://www.xyz.com/jv.html` are given in table 2. Statement #1 represents a individual name `http___www_xyz_com_jv_html`. This individual is instance of a class #Book and a class #DCElements11 (statements #2 and #3). The individual has an attribute named title1 connecting it to the object "Just Java". The attribute is instance of #DCElements11.title. The individual is also connected to the object "Peter van der Linden" with the author1 attribute. This attribute is instance of #DCElements11.author. Also the individual is connected via the namespace1 attribute to "http://www.xyz.com/" which is an instance of #Book.namespace.

The translation from RDF triple to O-Telos quadruple representation employ the axioms from O-Telos-RDF. Especially the axioms describing instantiation, property and type translation are used. Note that each RDF statement obtains a unique id to form the O-Telos statement. The ids are the statement identifiers really and as such are unique globally (except when exactly the same statements are made in two different places. In our example the ids are represented by a short form `sidx` to enhance readability which would expand to `namespace:resourcename`, e.g. here sid1 is the abbreviated form for `http://www.xyz.com:jv.html`.

If a RDF statement includes elements from other than the current schema, identified by a different than the current namespace, these elements are grouped in specially created O-Telos classes. For example the properties of the "Just Java" books' RDF description dc:title and dc:author, originating in the Dublin Core schema, are grouped in an O-Telos class called DCElements11 (see table 2). Note that the sids prefixed with '#' are stated for simplicity reasons and denote the respective class names. In reality these are again statements with own ids which would enlarge the example without adding any new insight. We therefore use the shortcut with the '#' prefix and the class name instead.

Also the O-Telos attribute `namespace` is introduced to hold the namespace of each RDF resource and property. Usually the namespace is part of the unique id of each statement as hinted in the example 2.1 but unfortunately the O-Telos frame syntax and parser forbids the possible special characters like dot, slash, dash, backslash, etc. of URLs in the element name. Therefore we introduce the namespace attribute that is assigned to each element and attribute. This attribute holds the namespace or resource URL respectively as a work around.

```
Individual http___www_xyz_com_jv_html
  in Book, DCElements
  with
    title
        title1 : "Just Java"
    author
        author1 : "Peter van der Linden"
    attribute
        namespace : "http://www.xyz.com/"
end
```

Example 2.1: O-Telos frame representation of the book "Just Java"

The frame representation of the O-Telos quadruples of table 2 is stated in example 2.1.

From the small example of the book "Just Java" in its RDF and O-Telos descriptions we have gained an insight on the translation of RDF to O-Telos. When applied to the whole book database the respective complete O-Telos representation is generated and stored

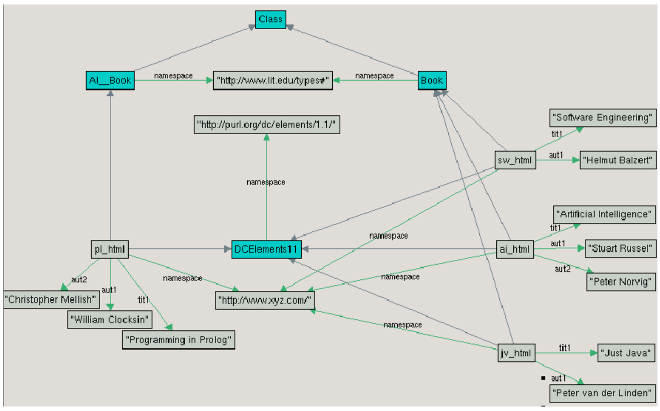

Figure 2. O-Telos graph of the example book database

in the ConceptBase database. Thus figure 2 shows the O-Telos graph of the same part of the book database which is shown as RDF graph in figure 1

The comparison of both graphs of the book database (figure 1 for RDF and figure 2 for O-Telos) shows their equivalence. They are different representations of the same book database using various different notations and serializations (triple/quadruple and XML/frame syntax). Thus by employing the translation described above it is possible to store RDF(S) data in the O-Telos meta data language. The next chapter shows how the lossless transformation of the RDF(S) data from its O-Telos representation is realized.

3 The peer query service

The chapter will provide some insight in how the data stored in an O-Telos provider-peer is regained using RDF queries. The RDF queries are posed to the peer by using the RDF-QEL exchange language as described in [10]. Based on Datalog [11] RDF-QEL abstracts the various RDF storage layer query languages (e.g. SQL) and user level query languages (e.g. RDQL) thus providing the syntax and semantics for an overall standard query interface across heterogeneous peer repositories for any kind of RDF meta data.

RDF-QEL does not distinguish between data and schema levels thus enabling the querying of different modeling levels. In doing so RDF-QEL conforms with the RDF(S) schema definitions and the more recent RDF model theory [4]. In order to describe and handle different query capabilities of this particular peer, we defined several RDF-QEL-i exchange language levels with increasing expressiveness: RDF-QEL-1 (conjunctive queries) is expressed as unreified RDF graphs while the higher levels use reified RDF statements for increased expressiveness (e.g. RDF-QEL-3 covers relational algebra, RDF-QEL-4 incorporates full Datalog). In this paper we will use examples in RDF-QEL-1 only for simplicity reasons. More advanced working examples can be found at `http://cip1-s.cip1.uni-hannover.de:3120/rdf2cb` where advanced features like negation are demonstrated.

The RDF-QEL queries are translated into O-Telos queries which than are posed to and answered by the ConceptBase database. The O-Telos answers are translated back to RDF in XML serialization. In order to clarify the translation further we will stick with the example from above. There the RDF description of the resource with the URL `http://www.xyz.com/jv.html` and the title "Just Java" is stored in the database.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
  <!DOCTYPE rdf:RDF [
    <!ENTITY rdf 'http://www.w3.org/1999/02
                        /22-rdf-syntax-ns#'>
    <!ENTITY rdfs 'http://www.w3.org/2000/0
                             1/rdf-schema#'>
    <!ENTITY dc 'http://purl.org/dc/element
                                   s/1.1/'>
    <!ENTITY edu 'http://www.edutella.org/e
                                dutella#'>
  ]>

  <rdf:RDF xmlns:rdf="&rdf;"
   xmlns:rdfs="&rdfs;"
   xmlns:dc="&dc;"
   xmlns:edu="&edu;">

  <edu:QEL1Query rdf:about="#genQuery">
    <edu:hasVariable rdf:resource="#X"/>
    <edu:hasVariable rdf:resource="#Y"/>
    <edu:hasResultType rdf:resource="&edu;G
                             raphResult"/>
  </edu:QEL1Query>

  <edu:Variable rdf:about="#X"
   dc:title="Just Java"
   rdfs:label="X">
    <dc:title rdf:resource="#Y"/>
  </edu:Variable>

  <edu:Variable rdf:about="#Y"
   rdfs:label="Y"/>
</rdf:RDF>
```

Example 3.1: The RDF-QEL XML representation of the example query

Now we want to know from the book database which books have the title “Just Java”. The respective query expressed in RDF-QEL-1 and XML is stated in example 3.1.

The query asks for at least one resource in the database which has a property called dc:title with a value “Just Java”. The resulting resource(s) are returned in XML notation. The peer constructs from this RDF representation the respective O-Telos representation. Thus the query is reformulated in O-Telos so that it now states that there is at least one resource in the database which has a property called dc:title with a value “Just Java”. If this is true the respective resources are retrieved.

As the example in 3.2 shows is each query an instance of ConceptBase’s build-in class QueryClass. It is a subclass of the O-Telos class Individual thus has as answers other instances of the class Individual. The answers hold attributes called title and namespace of type String. The constraint varX declares that all instances of the answer set must have an attribute named title and that this attribute holds the value ”Just Java”.

The namespace of the title attribute is derived from ConceptBase with a second query. For simplicity reasons we omit this query here. The query answers are than translated to RDF as shown in example 3.4.

```
QueryClass EduQuery isA Individual with
  retrieved_attribute
    title : String;
    namespace : String
  constraint
    varX : $ exists Y/Individual
             A(this,title,Y)
             and A(this,title,"Just Java") $
end
```

Example 3.2: The O-Telos frame representation of the query example

```
jv_html in EduQuery with
    title
        tit1 : "Just Java"
    namespace
        nam1 : "http://www.xyz.com/"
end
```

Example 3.3: Frame syntax of ConceptBases’ answer for the query shown in 3.2

The answer consists of RDF statements declaring that there is a resource with URL `http://www.xyz.com/jv.html`. This resource is the domain of a property named title which has the range “Just Java”. The title property is defined in the Dublin Core namespace.

The above example completes our presentations on how RDF(S) can be translated to O-Telos and back. The previous chapter introduced the ability of translating RDF to O-Telos thus enabling the storing capabilities of the meta database ConceptBase. This chapter now shows how the RDF data stored in ConceptBase can easily be retrieved and presented in RDF.

4 The peer’s implementation

The Edutella peer described is fully implemented in Java as is the whole EDUTELLA-Project. The actual implementation consists of three parts: the servlet enables a simple user interface, the EdutellaPeer implementation realizes the Edutella peer service and the EdutellaProvider provides the actual query service.

The demo and test servlet running at `http://www.cip1.uni-hannover.de :3120/rdf2cb/` runs within a SUN JavaWebServer 2.0. The servlet itself realizes the user interface for testing purposes only thus enabling users to pose RDF-QEL queries in the same way the Edutella peer service will do

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF
 [ ENTITY rdf 'http://www.w3.org/1999/02/22
                         -rdf-syntax-ns#'
   ENTITY a 'http://purl.org/dc/elements/1.
                                    1/' ]>
<rdf:RDF xmlns:rdf="&rdf;"
         xmlns:a="&a;">
<rdf:Description rdf:about="http://www.xyz.
                              com/jv.html"
   a:title="Just Java"/>
</rdf:RDF>
```

Example 3.4: The XML serialization of the query answer expressed in RDF and inspecting the respective answers.

Figure 3 shows that the Servlet as well as the Edutella peer pose EduQuery-objects to the Edutella provider implementation. The EduQuery-objects contain the queries. They are built using the classes of the Edutella software package, especially employing the parser that is based on an adaptation of the Jena RDF parser API [2].

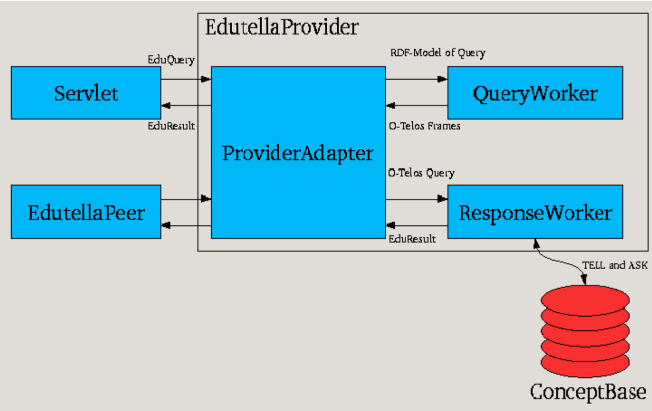

Figure 3. Schematic representation of the architecture

As shown in figure 3 the ProviderAdapter receives EduQuery objects. In a first step these are translated to O-Telos query objects by the QueryWorker. In a second step the ResponseWorker establishes a connection to the ConceptBase database, poses the query, receives the answer and returns the EduResult objects constructed from ConceptBases' answer. The ProviderAdapter itself returns the EduResult objects to either the servlet or the Edutella peer which in turn process them as according to their respective needs.

5 Summary

The evolving Edutella P2P network combines provider and client peers of various kinds. In this paper we have described a prototypical implementation of a provider peer which provides extended reasoning capabilities for RDF data. The provider peer uses the ConceptBase database which implements the meta modeling language O-Telos. Based on this database the basic Edutella services storage and querying are realized: a repository for storing RDF data and query facilities for the stored data.

The storage service uses a translation from RDF(S) to O-Telos and vice versa where property-centered RDF(S) data is translated to and from object-centered O-Telos data. Based on this translation the query service translates RDF-QEL queries to O-Telos query classes and translates the answers to these queries, instances of O-Telos query classes, back into RDF(S) graphs.

REFERENCES

[1] D. Brickley and R. V. Guha. Resource Description Framework (RDF) Schema Specification 1.0, 2002. http://www.w3.org/TR/rdf-schema.

[2] Hewlett-Packard Company. The Jena Semantic Web Toolkit, 2001. http://www.hpl.hp.com/semweb/jena-tob.html.

[3] Edutella Project Homepage. http://edutella.jxta.org.

[4] P. Hayes. RDF model theory. W3C working draft, February 2002. http://www.w3.org/TR/rdf-mt/.

[5] M. A. Jeusfeld, M. Jarke, H. W. Nissen, and M. Staudt. ConceptBase - Managing Conceptual Models about Information Systems. In P. Bernus, K. Mertins, and G. Schmidt, editors, *Handbook on Architectures of Informations Systems*. Springer Verlag, 1998.

[6] B. M. Kramer, V. K. Chandhri, M. Koubarakis, T. Topaloglon, H. Wang, and J. Mylopoulos. Implementing Telos. *SIGART Bulletin*, 2(3), June 1991.

[7] O. Lassila and R. Swick. W3C Resource Description framework (RDF) Model and Syntax Specification, 2001. http://www.w3.org/TR/REC-rdf-syntax/.

[8] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. Telos: A language for representing knowledge about information systems. *ACM Transactions on Information Systems*, 8(4), 1990.

[9] W. Nejdl, H. Dhraief, and M. Wolpers. O-Telos-RDF: a resource description format with enhanced metamodeling functionalities based on O-Telos. In *Workshop on Knowledge Markup and Semantic Annotation at the First International Concerence on Knowledge Capture (K-CAP'2001)*, October 2001. http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2001/kcap01_final.pdf.

[10] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *WWW 11 Conference Proceedings*, Hawaii, USA, May 2002. http://edutella.jxta.org/reports/edutella-whitepaper.pdf.

[11] J. D. Ullmann. *Principles of database and knowledgebase systems*. Principles of computer science series. Computer Science Press, Inc., Maryland, USA, 1988.