Qu, C., and Nejdl, W. (2004). Integrating XQuery-enabled SCORM XML Metadata Repositories into an RDF-based E-Learning P2P Network. *Educational Technology & Society*, 7 (2), 51-60.

Integrating XQuery-enabled SCORM XML Metadata Repositories into an RDF-based E-Learning P2P Network

Changtao Qu
Learning Lab Lower Saxony
University of Hannover
Expo Plaza 1, D-30539, Hannover, Germany
qu@learninglab.de

Wolfgang Nejdl
Learning Lab Lower Saxony
University of Hannover
Expo Plaza 1, D-30539, Hannover, Germany
nejdl@learninglab.de

Abstract

Edutella is an RDF-based E-Learning P2P network that is aimed to accommodate heterogeneous learning resource metadata repositories in a P2P manner and further facilitate the exchange of metadata between these repositories based on RDF. Whereas Edutella provides RDF metadata repositories with a quite natural integration approach, XML metadata repositories have to overcome considerable incompatibility between XML's tree-like hierarchical data model and RDF's binary relational data model in order to be integrated into Edutella. In this paper we investigate a generic approach for integrating XML metadata repositories into Edutella in terms of an XQuery-enabled native XML database containing SCORM XML metadata. We first propose a triple-like XML-based common data view to cross incompatibility between arbitrary XML data model and RDF data model, then discuss the wrapper program implementation for XML metadata repositories based on the wrapper-like Edutella content provider integration architecture. At last, we propose a generic approach for querying complex XML data schemas in Edutella through QBE (Query by Example), and present the design of a QBE-based SCORM query GUI that can be used to query SCORM XML metadata in Edutella in the RDF syntax.

Keywords

Resource Description Framework, Peer-to-Peer, XQuery, Sharable Content Object Reference Model, Query by Example

Introduction

The open source project Edutella is an RDF (Resource Description Framework) based E-Learning P2P (Peer-to-Peer) network that is aimed to accommodate heterogeneous learning resource metadata repositories in a P2P manner and further facilitate the exchange of metadata between these repositories based on RDF (Nejdl et al., 2002; Nejdl et al., 2003). At present Edutella is geared towards learning resource metadata repositories that are constructed based on several popular learning resource metadata sets, e.g., DCMES (Dublin Core Metadata Element Set) (DCMI, 1999), IEEE LOM (Learning Object Metadata) (IEEE LTSC, 2002), IMS learning resource metadata specification (IMS, 2001), SCORM (Sharable Content Object Reference Model) (ADL, 2003), etc., though its architecture and design do not make any assumptions about the applied metadata sets. In Edutella we make only one essential assumption that all Edutella resources can be described in RDF and further all Edutella functionalities can be mediated through RDF statements and the queries against these statements, as we believe the modular nature of RDF metadata to be especially suitable for distributed P2P settings. This essential assumption obviously leads to RDF being the most naturally applicable metadata representation in Edutella and RDF metadata repositories being the most natural form of Edutella content providers.

In spite of that, in practice we usually have to address another important type of Edutella content providers: XML (eXtensible Markup Language) metadata repositories containing XML binding metadata of aforementioned learning resource metadata sets. As a matter of fact, nowadays large amounts of learning resource metadata still exist on the Web in the format of XML. In comparison to RDF metadata repositories, at least at present XML metadata repositories still occupy a quite dominant place in E-Learning. Apart from the reason that simple XML has a flatter learning curve and a more straightforward binding strategy to popular learning resource metadata sets than RDF, another important reason is that XML has a longer history to be applied for binding E-Learning standards and specifications. Taking the IMS learning resource metadata specification as an example, it has provided the XML binding since version 1.0 released in August 1999, whereas its RDF binding has been introduced since version 1.2 released in June 2001. In consequence, currently most of existing educational repositories are XML-based (Duval et al., 2001; Liu et al., 2001; Qu et al., 2002). They contain a large amount of learning content to be addressed by Edutella.

Besides above reasons, the popularity of XML metadata repositories in E-Learning can also be attributed to a new type of XML back-end system: the native XML databases (Chaudhri et al., 2003). Unlike some other XML back-end systems such as RDBs (Relational Databases) and OODBs (Object-oriented Databases), in which XML metadata usually need to be pre-processed and stored in some transformed representations, e.g., decomposed relational tables in RDBs or decomposed objects in OODBs, the native XML databases provide a more straightforward way for constructing XML metadata repositories in that all XML metadata profiles can directly be stored and managed in their original hierarchical forms without the need of any pre-processing. In a native XML database, the database schema used to define how XML metadata are stored is virtually identical to the XML data schema defined by XML DTD (Document Type Definition) or XML Schema. Therefore, based on a specific XML data schema, multiple XML metadata profiles can be contained in a single collection thus be queried as a whole through using W3C XPath (Clark et al., 1999) or W3C XQuery (Boag et al., 2003). Also XML metadata profiles can easily be updated through direct manipulation on XML fragments instead of on the whole metadata profiles. In overall, all these features of native XML databases satisfactorily fit into the typical usage and management scenarios of learning resource metadata thus greatly promote the application of XML metadata repositories in E-Learning. Taking into account the current application status of XML metadata repositories, in this paper we focus on native XML metadata repositories to investigate the approach for integrating XML metadata repositories into Edutella.

Edutella Content Provider Integration Architecture

The P2P infrastructure design of Edutella is based on Sun's open source P2P platform JXTA (Sun Microsystems Inc., 2003). In order to integrate heterogeneous content provider peers, Edutella uses a wrapper-like (Papakonstantinou et al., 1995) content provider integration architecture as illustrated in Figure 1.

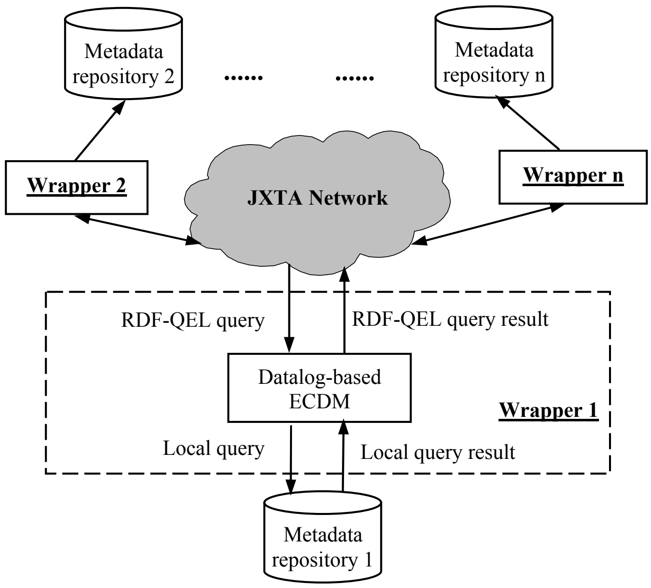

Figure 1. Wrapper-like Edutella content provider integration architecture

The key to the wrapper-like Edutella content provider integration architecture is ECDM (Edutella Common Data Model), which is shared by all metadata repositories and provides the common data view of the underlying metadata. At its basis, ECDM is a binary relational data model, which is defined in full compliance with the RDF data model and uses Datalog (Garcia-Molina et al., 2001) as its internal query language. Externally, Edutella defines a common query language: RDF-QEL (RDF Query Exchange Language) (Nilsson et al., 2003) to represent the query and query result. Taking ECDM as the unique interface, heterogeneous metadata repositories can be integrated into Edutella through different wrapper programs.

For each wrapper program, it has to accomplish two wrapper functionalities. First, it has to generate the ECDM-based common data view of underlying metadata. Second, it has to translate the RDF-QEL query into the local query languages, and vice versa, transform the local query result into the RDF-QEL query result. As RDF shares with ECDM/Datalog the central feature that their relational data models are based on sets of ground assertions conceptually grouped around properties, there exists a natural approach for the wrapper programs of RDF metadata repositories to generate the ECDM-based common data view, as well as to translate between RDF-QEL and local query languages based on Datalog (Nejdl et al., 2002). In contrast, for the wrapper programs of XML metadata repositories, it becomes difficult to achieve two wrapper functionalities.

Nowadays XML metadata repositories are principally supported by two XML query languages: XPath and XQuery. As XPath is relationally incomplete and incomparable to Datalog, for XPath-enabled XML metadata repositories, there exists no generic approach to integrating arbitrary XML schema based XML metadata repositories into Edutella (Qu et al., 2002a). However, for XQuery-enabled XML metadata repositories, as XQuery is much more powerful than XPath with regard to the transformation and expression capability, it becomes possible to integrate arbitrarily complex XML data schema based XML metadata repositories into Edutella through a generic integration approach. With the focus on E-Learning, in this paper we take a rather complex XML data schema: the SCORM XML binding as an example to present a generic approach for integrating XQuery-enabled XML metadata repositories into Edutella.

Integrating XQuery-enabled SCORM XML Metadata Repositories into Edutella: Design Issues

The SCORM is a model that references a set of interrelated technical specifications and guidelines with the purpose of achieving reusability, accessibility, durability, and interoperability of learning resources. The cornerstone of the SCORM is a Web-based learning "content aggregation model" defining how learning resources can be identified and described, aggregated into a course or portion of a course, and moved between different learning management systems or content repositories. In the latest SCORM 1.3, the SCORM content aggregation model is composed of four basic components (ADL, 2003):

- *SCORM content model*: Nomenclature defining the content components of a learning experience. Four types of defined components are Assets, SCA (Sharable Content Assets), SCO (Sharable Content Object), and Content Aggregation.
- *SCORM metadata*: A mechanism for describing specific instances of the components of the content model.
- *SCORM content packaging*: A mechanism defining how to represent the intended behavior of a learning experience (content structure) and how to package learning resources for the movement between different learning management systems (content packaging).
- *SCORM sequencing definition model*: A rule-based model defining how to sequence and deliver learning activities.

Among above four components, the key to the learning content discovery is the SCORM metadata, which provides descriptive information for each of the SCORM content model components to enable the discovery of learning resources at different aggregation levels. From the basis, the SCORM metadata information model is a reference to the IMS learning resource metadata information model, which itself is based on the IEEE LOM standard. In addition, the SCORM metadata also references the IMS learning resource metadata XML binding specification (IMS, 2001) and provides an XML representation for the SCORM metadata information data model. Throughout this paper we refer to the "SCORM XML metadata" as the XML representation of the SCORM metadata information model.

The SCORM XML metadata data model is rather complex, consisting of over 60 metadata entries and even recursive data structure, e.g., in the category of "classification". While integrating XQuery-enabled SCORM XML metadata repositories into Edutella, we have to address several critical design issues.

First, there exists considerable incompatibility between the XML's tree-like hierarchical data model and ECDM's binary relational data model. In order to integrate XML metadata repositories into Edutella, we first need to design the XML representation of the ECDM-based common data view, whose underlying XML data model is compatible with the ECDM's binary data model.

In practice, ECDM can be represented in the XML syntax in various ways. As the syntactic form of the XML data may strongly affect complexity and run-time performance of XQuery, we need to reasonably design the XML representation of the ECDM-based common data view. On the one hand, this XML-based common data view should be interoperable with ECDM. On the other hand, it should also be able to easily be transformed from any arbitrarily complex XML data schemas by means of XQuery.

Second, XQuery and Datalog use different mechanisms to express the query. This makes it quite difficult to translate Datalog into XQuery as well as to transform local XML query results into RDF-QEL results while implementing the wrapper program for XQuery-enabled XML metadata repositories.

Whereas Datalog is a relationally complete query language that can express relational algebra in terms of the relational data model, XQuery is a functional query language in which a query is generally represented as an expression in terms of the XML's tree-like data model. In most cases, Datalog queries cannot simply be represented through simple XQuery expressions, instead they usually have to be translated into sets of XQuery function calls. As the development of XQuery functions depends on individual XML data schema to be queried, in order to express the relational algebra and represent corresponding Datalog queries, we need to develop sets of XQuery functions based on the XML representation of the common data view. In addition, although the local XML query results returned by XQuery-enabled XML metadata repositories have already possessed an underlying data model that is compatible with ECDM, they still need some additional processing in order to be transformed into RDF-QEL results.

Third, As the SCORM XML metadata data model is rather complex, it is quite difficult to construct RDF-QEL queries for querying SCORM XML metadata repositories in Edutella.

Whereas for RDF metadata repositories the queries are always against a binary relational data model, for SCORM XML metadata repositories, the queries are against a tree-like hierarchical data model that possesses more than 60 metadata entries. For these metadata entries, as the user's query interest is unforeseeable, we cannot expect which metadata entry would be queried and how the Boolean logics between these queries would seem. Including all metadata entries and all possible query Boolean logics in a form-like query GUI (Graphical User Interface) is a straightforward first idea, but can usually lead to some cumbersome and inefficient query experiences. Moreover, as the queries should uniquely be in the RDF-QEL format in Edutella, it becomes more complex for inexperienced users to construct RDF-QEL queries to query SCORM XML metadata repositories.

In order to address above design issues to integrate XQuery-enabled SCORM XML metadata repositories into Edutella, in this paper we propose a triple-like XML-based common data view to represent the ECDM data model, and a QBE (Query by Example) (Zloof, 1977) based SCORM query GUI to simplify the query construction process. As the key to the system design, the triple-like common data view can overcome incompatibility between the SCORM XML metadata data model and ECDM, and at the same time ensure XQuery's run-time performance.

Integrating XQuery-enabled SCORM XML Metadata Repositories into Edutella: Technical Implementation

According to the wrapper-like Edutella content provider integration architecture, integrating XQuery-enabled SCORM XML metadata repositories into Edutella includes two major tasks. The first task is to manipulate SCORM XML metadata instances through XQuery to generate the triple-like XML-based common data view. This is actually the prerequisite for SCORM XML metadata repositories to be integrated into Edutella and sequentially be queried via ECDM's internal query language Datalog. The second task is to develop the wrapper program for SCORM XML metadata repositories to accomplish two wrapper functionalities: (1) translating RDF-QEL into XQuery; and (2) transforming the local XML query result into the RDF-QEL result. As we have mentioned, in order to simplify the query construction process, we also have the third task: implementing the QBE-based SCORM query GUI.

Generating the triple-like XML-based Common Data View

The XML-based common data view is the XML representation of the ECDM data model. It should have two features. First, its underlying data model should be in full compliance with ECDM/RDF's binary relation data model. Second, it should use a very simple XML syntax, which can easily be manipulated through XQuery to ensure the query's run-time performance. In this paper we propose a triple-like XML-based common data view, which can be described in the XML DTD as:

```
<!ELEMENT statements (statement+)>
<!ELEMENT statement (subject, predicate, object)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT predicate (#PCDATA)>
<!ELEMENT object (#PCDATA)>
```

For SCORM XML metadata repositories, the triple-like XML-based common data view can directly be generated through XQuery without the loss of any original SCORM XML metadata information. In order to present the generating process, we take a SCORM XML binding metadata entry: lom.general.catalogentry as an example. In Figure 2 we show the graphical data model of this metadata entry in the form of the XML Schema.

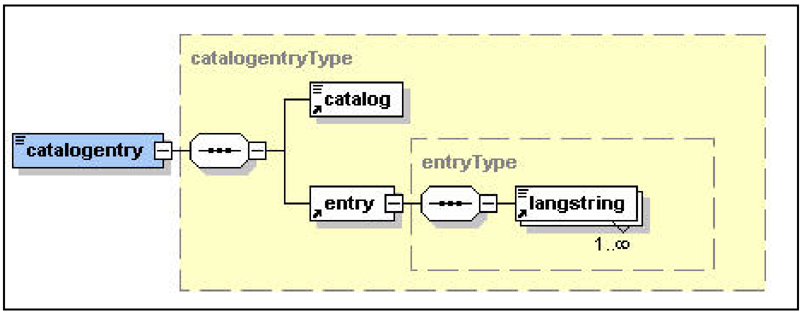

Figure 2. Graphical data model of a SCORM XML metadata entry

An example metadata instance of this metadata entry might seem as:

```
<?xml version="1.0" encoding="UTF-8"?>
<lom xmlns="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1
                        http://www.imsglobal.org/xsd/imsmd_rootv1p2p1.xsd">
   <general>
       <catalogentry>
          <catalog>Blah</catalog>
             <entry>
                <langstring xml:lang="en"> Blah EN</langstring>
                <langstring xml:lang="de"> Blah DE</langstring>
             </entry>
        </catalogentry>
   </general>
</lom>
```

In order to generate the triple-like XML-based common data view, we re-describe the example SCORM XML metadata instance into an RDF graph through using RDF properties to represent XML elements. In Figure 3 we illustrate the RDF graph representing the example SCORM XML metadata instance.

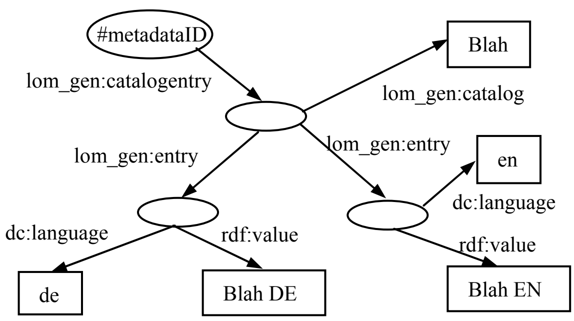

Figure 3. RDF graph representing the example SCORM XML metadata instance

In terms of the XML DTD of the triple-like common data view, the RDF graph can be serialized into the XML representation as illustrated in Figure 4. The serialization is realized through using a self-developed XQuery function library.

```xml
<?xml version='1.0' encoding='ISO-8859-1'?>
<statements xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:lom_gen="http://www.imsproject.org/rdf/imsmd_generalv1p2#">

<statement>
  <subject>#metadataID</subject>
  <predicate>lom_gen:catalogentry</predicate>
  <object>#anonymous_0</object>
</statement>

<statement>
  <subject># anonymous_0</subject>
  <predicate>lom_gen:catalog</predicate>
  <object>Blah</object>
</statement>

<statement>
  <subject># anonymous_0</subject>
  <predicate>lom_gen:entry</predicate>
  <object>#anonymous_1</object>
</statement>

<statement>
  <subject>#anonymous_0</subject>
  <predicate>lom_gen:entry</predicate>
  <object>#anonymous_2</object>
</statement>

<statement>
  <subject># anonymous_1</subject>
  <predicate>dc:language</predicate>
  <object>en</object>
</statement>

<statement>
  <subject># anonymous_1</subject>
  <predicate>rdf:value</predicate>
  <object>Blah EN</object>
</statement>

<statement>
  <subject># anonymous_2</subject>
  <predicate>dc:language</predicate>
  <object>de</object>
</statement>

<statement>
  <subject># anonymous_2</subject>
  <predicate>rdf:value</predicate>
  <object>Blah DE</object>
</statement>
</statements>
```

Figure 4. The XML serialization of the example SCORM XML metadata instance

While representing the SCORM XML metadata instance as the RDF graph, we try to follow the RDF model and syntax specification (Lassila et al., 1999) as much as possible. On the one hand, we make heavy use of anonymous resources as well as some RDF built-in properties such as *rdf:type* and *rdf:value*. On the other hand, we also try to remain compatibility with some popular metadata sets such as the DCMES, DCMI Metadata Terms (DCMI, 2003) and vCard, etc., in the representation. However, in spite of that, the triple-like common data view of SCORM XML metadata is not fully compatible with the IMS learning resource metadata RDF binding specification (IMS, 2001), which is usually viewed as the potential SCORM metadata RDF binding. In fact, the potential SCORM RDF binding is proposed without taking into account compatibility with the SCORM XML binding. It takes advantage of the semantic richness of RDF and goes beyond a simple syntactic level representation of the SCORM metadata information model, which makes it quite hard to transform SCORM XML binding metadata into potential RDF binding metadata without the loss of original metadata information. Therefore, while handling SCORM XML binding metadata, we should not expect 100% compatibility with the potential SCORM RDF binding.

Furthermore, we should also be aware that it is unnecessary to achieve full compatibility between the triple-like common data view of SCORM XML metadata and potential SCORM RDF binding metadata. From the syntactic view, the potential SCORM RDF binding uses several RDF built-in properties such as *rdf:type* in a rather ambiguous way, which might increase complexity and run-time cost of XQuery functions. While handling self-contained SCORM XML metadata, it is unnecessary for us to bear such sort of additional query overhead.

Nevertheless, we still tried to achieve the maximal compatibility between the triple-like common data view of SCORM XML metadata and potential SCORM RDF binding metadata with the purpose of handling SCORM XML and RDF metadata in a uniform way in Edutella. Some efforts include, e.g., using the same namespaces proposed in the potential SCORM RDF binding, remaining compatibility with DCMES, DCMI Metadata Terms and vCard in the triple-like common data view, etc. These efforts have actually covered most of principal design criteria of the potential SCORM RDF binding.

Developing the Wrapper Program for SCORM XML Metadata Repositories

The wrapper program of XQuery-enabled SCORM XML metadata repositories has to accomplish two wrapper functionalities: (1) translating RDF-QEL into XQuery; and (2) transforming local XML query results into RDF-QEL results. Since we adopt a triple-like common data view whose underlying data model is quite close to the ECDM/RDF's binary relational data model, and also because XQuery is capable of returning query results in any desirable XML formats, the accomplishment of the second wrapper functionality is relatively straightforward. In this section our discussion will be focused on the implementation of the first wrapper functionality.

According to the Edutella content provider integration architecture, translating RDF-QEL into XQuery consists of two sub-tasks. First, RDF-QEL queries have to be translated into Datalog queries. This is a common task for all types of Edutella wrapper programs and can be completed through a common parser program in Edutella (Nejdl et al., 2002). Second, Datalog queries have to be translated into XQuery queries, more precisely, into sets of calls to the self-developed XQuery function library.

Datalog is a non-procedural, relationally complete query language based on Horn clauses without function symbols (Garcia-Molina, et al. 2001). As its counterpart, XQuery is also relationally complete and can perform all operations of relational algebra on XML's tree-like data model including some complicated query operations such as the recursive query. In order to present the translation process from Datalog to XQuery, we illustrate an example Datalog query in Figure 5, which can be read in plain English as: *find a SCORM XML metadata record, whose lom.general.title entry contains English value "computer" and lom.general.keyword entry contains English value "TCP", or a SCORM XML metadata record, whose lom.general.description entry contains English value "network".*

```
scorm(X) :- lom_gen:title(X,U), dc:language(U,"en"), rdf:value(U,"computer"),
            lom_gen:keyword(X,V), dc:language(V,"en"), rdf:value(V,"TCP")
scorm(X) :- lom_gen:description (X,W), dc:language(W,"en"), rdf:value(W, "Network")
? - scorm(X)
Here X,U,V,W are Datalog variables
```

Figure 5. Example Datalog query

One basic construct of Datalog is the Literal, which describes ground assertion and can be represented in a simplified form corresponding to the binary relational data model as: *P(arg1, arg2)*, where *P* is Predicate that might be a relation name or arithmetic predicates (e.g., “<”, “>”, etc.), and *arg1, arg2* are Arguments that might be variables or constants. A Datalog query can be expressed as a set of Datalog rules. Each Datalog rule has a general representation as *head :- literal1, literal2,..., literaln*, where head is a single positive Literal, and *literal1* to *literaln* are a set of Literals conjunctively called the body of the Datalog rule. The disjunction in Datalog is expressed as a set of rules with the identical head. As illustrated in figure 5, the example Datalog query consists of two rules covering conjunctive and disjunctive query. It can be translated into sets of calls to the self-developed XQuery function library, as illustrated in Figure 6.

```
let $p := query_on_element_with_langstring("lom_gen_title","","en","computer")
let $q := query_on_element_with_langstring("lom_gen_keyword ","","en","TCP")
let $r := query_on_element_with_langstring("lom_gen_description","","en", "Network")
return handle_Boolean_OR( handle_Boolean_AND($p union $q) union $r)
```

Figure 6. XQuery query translated from the example Datalog query

The XQuery query is generally represented through the FLOWR (For-Let-Where-Order by-Return) expression, which generalizes select-from-having-where expression from SQL (Structure Query Language) and supports iteration and binding of variables to intermediate results. By means of the FLOWR expression, we can compute joins between different SCORM XML metadata records and further re-structure the query results. In our implementation, the self-developed XQuery function library contains sets of functions used to query different SCORM XML metadata entries. The returned query results are further handled by two specific XQuery functions: "handle_Boolean_OR" and "handle_Boolean_AND", which are responsible for managing the Boolean logics between the queries against multiple metadata entries. These two XQuery functions can also eliminate duplicate local XML query result sets.

Implementing the QBE-based SCORM Query GUI

QBE is a graphical language originally designed for querying RDBs. The idea behind QBE is that the user provides an example of outputs that he expects from the query and constructs the query by filling example tables (Zloof, 1977). Whereas QBE fits well with RDBs in that QBE's tabular query interface is quite analogous to the internal tabular structure of RDBs, it cannot directly be used to query native XML metadata repositories, which, as the document databases by nature, adopt tree-like hierarchical data models to store XML metadata. In order to query SCORM XML metadata repositories in Edutella, we propose an improved QBE, which uses a visual template to represent the query against individual XML metadata entry, and further adopts a single table to represent the Boolean logics between multiple visual templates. While the visual template provides a quite analogous representation of the internal structure of individual XML metadata entry, the single tabular structure inherits QBE's original forte for representing the Boolean logics between queries.

According to above design idea, we implement the QBE-based SCORM query GUI that can be used to query SCORM XML metadata repositories in Edutella in the RDF-QEL syntax. The QBE-based SCORM query GUI has four features:

- Arbitrary SCORM XML metadata entry could be taken as the "query example".
- User-friendly drag & drop manipulation based on the SCORM XML binding DOM (Document Object Model) tree.
- Automatic RDF-QEL output.
- Integration of the graphical RDF-QEL result presentation.

In Figure 7 we show a screen shot of the QBE-based SCORM query GUI, which demonstrates the construction process of the example query illustrated in figure 5/figure 6.

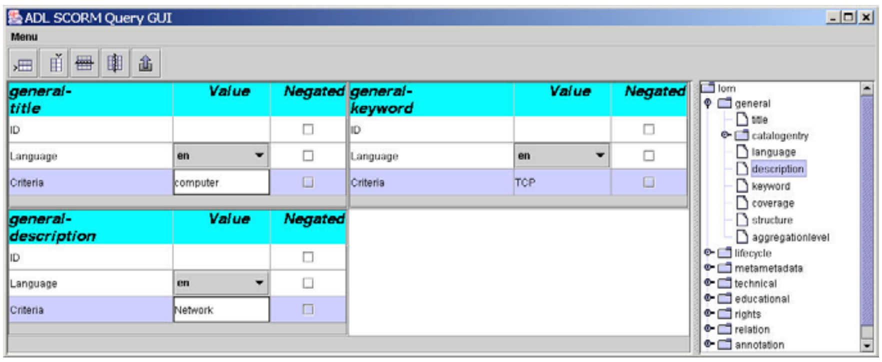

Figure 7. QBE-based SCORM query GUI

In the QBE-based SCORM query GUI, the user can first choose SCORM XML metadata entries to be queried from the SCORM XML binding DOM tree through drag & drop manipulation. The user can then compose sets of visual templates corresponding to different SCORM XML metadata entries in a table according to the expected Boolean logics between the queries. The output of the SCORM query GUI is RDF-QEL queries. These

RDF-QEL queries can then be sent to the Edutella network and are expected to get query results from all SCORM XML metadata repositories.

Conclusions

Acknowledging the inherent advantages of RDF such as easy composability of schemas, extendability and modularity of distributed RDF metadata, etc., Edutella adopts a fully RDF-based design to manage learning resource metadata, aiming to become part of the future Semantic Web. Still, we do want to use large amounts of XML-based learning resource metadata, which have been produced in E-Learning in the last years and are still populating today. In this paper we present our exploration to integrate XQuery-enabled SCORM XML metadata repositories into Edutella. Actually, the integration approach is not merely applicable to XML-based learning resource metadata in Edutella. It can be extended as a generic approach for managing arbitrary XML metadata in any RDF-based Semantic Web settings.

References

ADL (2003). SCORM Specification V1.3 Application Profile. Retrieved February 24, 2004 from http://www.adlnet.org.

Boag, S., Chamberlin, D., Fernandez, M. F., Florescu, D., J. Robie, & Siméon, J. (2003). XQuery 1.0: An XML Query Language. Retrieved February 24, 2004 from http://www.w3.org/TR/xquery/.

Chaudhri, A., Zicari, R., & Rashid, A. (2003). *XML Data Management: Native XML and XML Enabled DataBase Systems*, USA: Addison-Wesley.

Clark, J., & DeRose, S. (1999). XML Path Language (XPath), Version 1.0. Retrieved February 24, 2004 from http://www.w3.org/TR/xpath.

DCMI (1999). Dublin Core Metadata Element Set, Version 1.1. Retrieved February 24, 2004 from http://dublincore.org/documents/1999/07/02/dces.

DCMI (2003). DCMI Metadata Terms. Retrieved February 24, 2004 from http://dublincore.org/documents/2003/03/04/dcmi-terms/.

Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., & Hendrikx, K. (2001). The ARIADNE Knowledge Pool System. *Communications of the ACM, 44* (5), 73-78.

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2001). *Database Systems: The Complete Book*, USA: Prentice Hall.

IEEE LTSC (2002). IEEE 1484.12.1: Draft Standard for Learning Object Metadata. Retrieved February 24, 2004 from http://ltsc.ieee.org.

IMS (2001). IMS Learning Resource Metadata Specification V1.2.1. Retrieved February 24, 2004 from http://www.imsproject.org.

Lassila, O., & Swick, R. R. (1999). Resource Description Framework (RDF) Model and Syntax Specification. Retrieved February 24, 2004 from http://www.w3.org/TR/REC-rdf-syntax.

Liu, X., Maly, K., Zubair, M., & Nelson, M. L. (2001). Arc - An OAI Service Provider for Cross Archiving Searching. *Paper presented at the ACM/IEEE Joint Conference on Digital Libraries*, June 24-28, 2001, Roanoke, VA, USA.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Stintek, M., & Naeve, A. (2002). Edutella: A P2P Networking Infrastructure Based on RDF. *Paper presented at the 11th International World Wide Web Conference*, May 7-11, 2002, Hawaii, USA.

Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., & Brunkhorst, I. (2003). Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. *Paper presented at the 12th International World Wide Web Conference*, May 20-24, 2003, Budapest, Hungary.

Nilsson, M, & Siberski, W. (2003). RDF Query Exchange Language (QEL): Concepts, Semantics, and RDF Syntax. Retrieved February 24, 2004 from http://edutella.jxta.org/.

Qu, C., & Nejdl, W. (2002). Towards Interoperability and Reusability of Learning Resource: a SCORM-conformant Courseware for Computer Science Education. *Paper presented at the the 2nd IEEE International Conference on Advanced Learning Technologies*, September 9-12, 2002, Kazan, Russia.

Qu, C., Nejdl, W., & Schinzel, H. (2002a). Integrating Schema-specific Native XML Repositories into a RDF-based E-Learning P2P Network. *Paper presented at the 2nd International Conference on Dublin Core and Metadata Applications (DC 2002)*, October 13-17, 2002, Florence, Italy.

Papakonstantinou, Y., Garcia-Molina, H., & Widom, J. (1995). Object Exchange Across Heterogeneous Information Sources. *Paper presented at the 11th International Conference on Data Engineering*, March 6-10, 1995, Taipei, Taiwan.

Sun Microsystems Inc. (2003). *Project JXTA*. Retrieved February 24, 2004 from http://www.jxta.org/.

Zloof, M. M. (1977). Query by Example: a database language, *IBM Systems Journal*, *16* (4), 324-343.