Proc. Int. Conf. on Dublin Core and Metadata for e-Communities 2002: 81-89

Integrating Schema-specific Native XML Repositories into a 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
Computer Science Dept.
Stanford University, Stanford, CA 94305, USA
nejdl @db.stanford.edu
(On leave from University of Hannover)

Holger Schinzel
Learning Lab Lower Saxony
University of Hannover
Expo Plaza 1, D-30539
Hannover, Germany
schinzel @learninglab.de

Abstract

As its name implies, a native XML repository supports storage and management of XML in the original hierarchical form rather than in some other representations. In this paper we present our approach for integrating native XML repositories into Edutella, a RDF-based E-learning P2P network, through mapping native XML database schemas onto the Edutella Common Data Model (ECDM) and further translating ECDM's internal query language Datalog into XPath, the local query language of native XML repositories. Due to the considerable incomparability between the ECDM and the XML data model, a generic integration approach for schema-agnostic native XML repositories is found to be unrealistic. Thus our investigations are focused on three schema-specific native XML repositories respectively based on the DCMES, LOM/IMS, and SCORM XML binding data schema. Since these three metadata sets are the most popularly applied learning resource metadata specifications in E-Learning, our integration approach satisfactorily addresses the current usage of Edutella in E-Learning despite that a generic integration approach for schema-agnostic native XML repositories has not been implemented.

Keywords: *repositories, E-learning network.*

1. Introduction

The open source project Edutella[1] is a RDF (Resource Description Framework)-based E-Learning P2P (Peer-to-Peer) network that aims at accommodating distributed learning resource metadata repositories, which are generally heterogonous in applied back-end systems, applied metadata schemas, etc., in a P2P manner and further facilitating the exchange of learning resource metadata between these repositories based on RDF [16]. At present Edutella is geared towards learning resource metadata repositories that are constructed based on three popular learning resource metadata sets: DCMES (Dublin Core Metadata Element Set)[7], IEEE LOM (Learning Object Metadata)/IMS Learning Resource Metadata Specification [11][12],

[1] http://edutella.jxta.org

https://doi.org/10.23106/dcmi.952106977

and ADL (Advanced Distributed Learning) SCORM (Sharable Content Object Reference Model)[1], though its architecture and design does 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 on 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 the Edutella network and thus RDF-based repositories containing the metadata of RDF bindings to above three learning resource metadata specifications are the most natural form of Edutella content provider peers.

However, in practice we currently have to address another important form of Edutella content providers: the XML (eXtensible Markup Language)-based repositories containing the metadata of XML bindings to the three learning resource metadata sets mentioned above. As a matter of fact, at present the XML-based learning resource metadata repositories still occupy a quite dominant place in E-Learning in comparison to the RDF-based repositories, although the latter ones have recently found more and more application cases [4][8]. Besides the reason that simple XML has a flatter learning curve and also a more straightforward binding strategy to all three learning resource metadata specifications in comparison to RDF, another important reason lies in the fact that XML has a longer history to be applied for binding learning resource metadata specifications than RDF. Taking the LOM/IMS metadata specification as an example, it has provided the XML binding since version 1.0, released in August 1999, whereas its RDF binding has only been introduced since version 1.2, released in June 2001. As a direct consequence, currently most of existing learning resource metadata repositories are XML-based [9][15][17][19], containing a large number of learning resource metadata to be addressed by Edutella.

In addition, the XML-based repositories also introduce a new type of back-end system: the native XML database, which provides a very straightforward way for constructing learning resource metadata repositories in that all learning resource XML metadata profiles can be directly stored and managed in the native XML repositories without the need of any pre-processing. The native XML databases support storage and management of XML in the original hierarchical form rather than in some other representations, e.g., decomposed relational tables in RDBs (Relational Database), or decomposed objects in OODBs (Object-oriented Database). Moreover, in a native XML database, the database schema used to define how the XML is stored is virtually identical to the XML data schema defined by XML DTD (Document Type Definition) or W3C (World Wide Web Consortium) XML Schema [20]. Based on a specific XML data schema, multiple XML metadata profiles can be contained in a single collection and thus can be queried as a whole through using W3C XPath [6], the query language supported by almost all native XML databases. Also the stored XML metadata profiles can be easily updated through direct manipulation on XML fragments instead of on the whole profiles. As a matter of fact, these features of the native XML databases satisfactorily fit into the typical usage and management scenarios of learning resource metadata and thus greatly promote the application of the native XML repositories in E-Learning.

However, despite of the fact that the XML-based learning resource metadata repositories have been popularly applied in E-Learning, there exists a big obstacle to integrate them into the RDF-based Edutella network. This obstacle comes from the considerable incomparability between RDF's binary relational data model and XML's hierarchical data model, which makes it difficult to establish the mapping from an arbitrary XML data schema to the RDF data model, although the reverse mapping is definitely feasible [14]. Therefore, in this paper we will mainly concentrate on three schema-specific native XML repositories, which accommodate learning resource metadata respectively based on the DCMES, LOM/IMS, and SCORM XML binding schema, and present our approach for integrating them into the RDF-based Edutella network. Since these three metadata sets are the most popularly applied learning resource metadata specifications in E-Learning, our integration approach satisfactorily addresses the current usage of Edutella in E-Learning despite that a generic integration approach for schema-agnostic native XML repositories has not been implemented.

2. Edutella provider integration architecture

Edutella employs a wrapper-like architecture for integrating heterogeneous content provider peers. In figure 1 we illustrate the Edutella provider integration architecture.

The wrapper-like architecture has been popularly applied for integrating heterogeneous information sources for many years [10][18]. The key to such sort of integration architecture is a common data model that is shared by all information sources and provides the common data view of the underlying heterogeneous repositories. For each wrapper program, it is on the one hand responsible for generating the common data view of the individual repository based on the pre-defined common data model, on the other hand, it is also responsible for translating the common query language for the common data view into the local query language of the individual repository, and vice versa, transforming the local query results into the results represented by the common result

https://doi.org/10.23106/dcmi.952106977

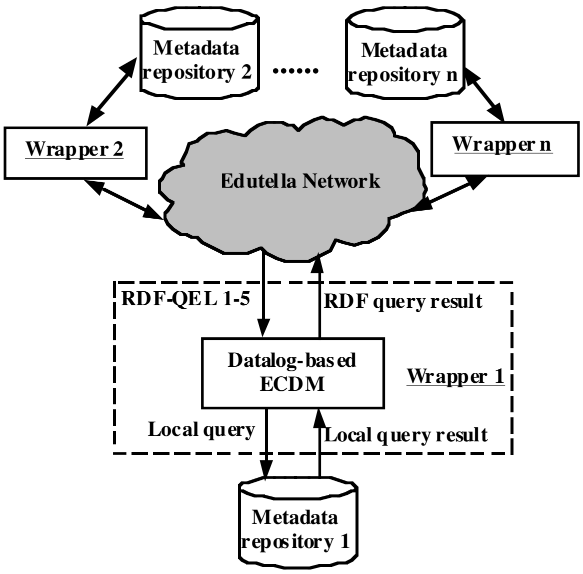

Figure 1. The Edutella provider integration architecture

exchange format after the query against the individual repository is completed.

Following this design and usage scenario, in Edutella we first proposed the Edutella Common Data Model (ECDM), which is defined in full compliance with the RDF data model and uses Datalog [10] as its internal query language. Externally, we defined a common query language: RDF Query Exchange Language (RDF-QEL) for the whole Edutella network using RDF syntax. As illustrated in Figure 1, the wrapper program of each Edutella provider is responsible for translating RDF-QEL into ECDM's internal query language Datalog. Because Datalog and RDF share 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 generating the ECDM-based common data view of the RDF-based repositories, as well as a natural approach for translating RDF-QEL into Datalog, which is internally used to manipulate the ECDM-based common data view.

Based on the wrapper-like Edutella provider integration architecture, we have successfully integrated several heterogeneous content provider peers into the Edutella network [16]. However, when we tried to handle the native XML repositories, several issues had to be addressed.

First, the XML data model is in some sense quite incomparable to the ECDM, which makes it difficult to integrate schema-agnostic XML-based repositories into Edutella. The ECDM, which is compliant with the RDF data model as well as the Datalog data model, is at its basis a binary relational data model consisting of a set of ground assertions represented either as binary predicates: *Predicate(Subject, Object)* or as ternary statements *s(Subject, Predicate, Object)*, if the predicate is taken as an additional argument. In contrast to the ECDM, the XML data model, which possesses a tree-like hierarchical data structure, cannot be easily mapped onto a binary relational data model, especially when the XML data schemas become complex enough, e.g., containing recursive elements, as it occurs in the LOM/IMS XML binding [12]. Moreover, in comparison to some powerful query languages supported by RDBs and OODBs, which can be used to generate the ECDM-based common data view of the underlying repositories, the XPath query language, which is currently the most used tool for manipulating the native XML repositories, is much weaker and thus incapable of manipulating some complex XML data models to generate their ECDM-based common data view. This incomparability between the XML data model and the ECDM influenced our decision to apply our integration approach only to several schema-specific XML repositories at the current time.

Second, in comparison to ECDM's internal query language Datalog, XPath is also far from comparable and thus cannot express all Datalog queries. Whereas Datalog is a relationally complete query language that is able to express relational algebra such as "selection", "union", "join", and "projection", etc., and also possesses some additional features such as transitive closure and recursive definitions, XPath can only express part of relational algebra, such as "union", limited "selection", and "negation" in terms of the XML tree-like data model, but lacks the support for expressing "join" and "projection". As introduced in our previous publication [16], at present we have defined five sets of RDF-QELs in the Edutella network according to their different expressivity, namely, RDF-QEL1 (can express conjunctive query), RDF-QEL2 (RDF-QEL1 plus disjunctive query), RDF-QEL3 (RDF-QEL2 plus query negation), RDF-QEL4 (RDF-QEL3 plus linear recursive query), and RDF-QEL5 (RDF-QEL4 plus arbitrary recursive query), all of which can be transparently translated into the corresponding Datalog queries. While all sets of RDF-QEL queries can be fully handled by some high-performance RDF-based repositories such as RDBs supporting SQL3, the native XML repositories can only handle part of the RDF-QEL sets, namely, RDF-QEL1 to RDF-QEL3. In fact, the weak expressivity of XPath determines that the native XML repositories in the Edutella network are unable to achieve the same functionalities as other high-performance repositories with the support of some powerful local query languages.

Finally, the incomparability between the XML data model and the ECDM as well as the incomparability between Datalog and XPath also have a negative influence on the query result representation of the native XML repositories. Whereas the RDF-based repositories can naturally adapt the query results into Edutella's RDF-based common result exchange format with the support of some powerful local

https://doi.org/10.23106/dcmi.952106977

query languages, the native XML repositories can only return XML fragments selected by the XPath expressions rather than sets of tuples that can be naturally brought into the RDF model due to XPath's limited capability of expressing "selection", as well as its incapability of expressing "join" and "projection". Therefore, the query results generated by the native XML repositories need some additional processing in order to be adapted into the Edutella common result exchange format.

In the following we will present our approach addressing above issues. The native XML repository introduced here is implemented using the open source project Apache Xindice 1.0[2], but the presented approach is also applicable to some other native XML repository implementations, e.g., Tamino XML database 3.1.1.1[3], Ipedo XML database 3.0.1[4], etc. In addition, although our approach will address three schema-specific native XML repositories that accommodate learning resource metadata respectively based on the DCMES, LOM/IMS, and SCORM XML binding schema, we will use the DCMES, which constitutes the minimal interoperable basis of some more complicated metadata sets, as the "standard" schema throughout the discussion. In section 6 we will describe the integration approach, which is based on the DCMES XML binding data schema, for integrating the LOM/IMS and SCORM XML binding schema based native XML repositories into the Edutella network.

3. Generating the ECDM-based common data view of the native XML repositories

The DCMES XML binding [2] is the guideline proposed by DCMI (Dublin Core Metadata Initiative) for the XML encoding of DCMES. The primary goal of this guideline is to provide a simple DCMES encoding, where there are no extra elements, qualifiers, operational or varying parts allowed. The secondary goal is to make the encoding also be valid RDF, which allows the XML binding to be manipulated using the RDF model. For the DCMES XML binding schema based native XML repositories contained in the Edutella network, the second design goal of the DCMES XML binding to a certain degree facilitates the adaptation of their local query results into the Edutella common result exchange format[5].

In Figure 2 we show the XML schema of the DCMES XML binding in the format of XML DTD [2].

The above XML schema can be also viewed in a schematic way, represented in the hedgehog model, as depicted in Figure 3 [13].

From the hedgehog model of the DCMES XML binding, in which all assertions are made about a

```
<!ENTITY rdfns 'http://www.w3.org/1999/02/22-rdf-syntax-ns#' >
<!ENTITY dcns 'http://purl.org/dc/elements/1.1/' >
<!ENTITY % rdfnsdecl 'xmlns:rdf CDATA #FIXED "&rdfns;"' >
<!ENTITY % dcnsdecl 'xmlns:dc CDATA #FIXED "&dcns;"'>
<!ELEMENT rdf:RDF (rdf:Description)* >
<!ATTLIST rdf:RDF %rdfnsdecl; %dcnsdecl; >
<!ENTITY % dcmes "dc:title | dc:creator | dc:subject | dc:description|
dc:publisher | dc:contributor | dc:date | dc:type | dc:format |
dc:identifier | dc:source | dc:language | dc:relation | dc:coverage |
dc:rights" >
<!ELEMENT rdf:Description (%dcmes;)* >
<!ATTLIST rdf:Description rdf:about CDATA #IMPLIED>
<!ELEMENT dc:title (#PCDATA)>
<!ATTLIST dc:title xml:lang CDATA #IMPLIED>
<!ATTLIST dc:title rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:creator (#PCDATA)>
<!ATTLIST dc:creator xml:lang CDATA #IMPLIED>
<!ATTLIST dc:creator rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:subject (#PCDATA)>
<!ATTLIST dc:subject xml:lang CDATA #IMPLIED>
<!ATTLIST dc:subject rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:description (#PCDATA)>
<!ATTLIST dc:description xml:lang CDATA #IMPLIED>
<!ATTLIST dc:description rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:publisher (#PCDATA)>
<!ATTLIST dc:publisher xml:lang CDATA #IMPLIED>
<!ATTLIST dc:publisher rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:contributor (#PCDATA)>
<!ATTLIST dc:contributor xml:lang CDATA #IMPLIED>
<!ATTLIST dc:contributor rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:date (#PCDATA)>
<!ATTLIST dc:date xml:lang CDATA #IMPLIED>
<!ATTLIST dc:date rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:type (#PCDATA)>
<!ATTLIST dc:type xml:lang CDATA #IMPLIED>
<!ATTLIST dc:type rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:format (#PCDATA)>
<!ATTLIST dc:format xml:lang CDATA #IMPLIED>
<!ATTLIST dc:format rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:identifier (#PCDATA)>
<!ATTLIST dc:identifier xml:lang CDATA #IMPLIED>
<!ATTLIST dc:identifier rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:source (#PCDATA)>
<!ATTLIST dc:source xml:lang CDATA #IMPLIED>
<!ATTLIST dc:source rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:language (#PCDATA)>
<!ATTLIST dc:language xml:lang CDATA #IMPLIED>
<!ATTLIST dc:language rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:relation (#PCDATA)>
<!ATTLIST dc:relation xml:lang CDATA #IMPLIED>
<!ATTLIST dc:relation rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:coverage (#PCDATA)>
<!ATTLIST dc:coverage xml:lang CDATA #IMPLIED>
<!ATTLIST dc:coverage rdf:resource CDATA #IMPLIED>
<!ELEMENT dc:rights (#PCDATA)>
<!ATTLIST dc:rights xml:lang CDATA #IMPLIED>
<!ATTLIST dc:rights rdf:resource CDATA #IMPLIED>
```

Figure 2. The XML DTD of the DCMES XML binding

[2] http://xml.apache.org/xindice
[3] http://www.softwareag.com/tamino
[4] http://www.ipedo.com
[5] see also section 5.

https://doi.org/10.23106/dcmi.952106977

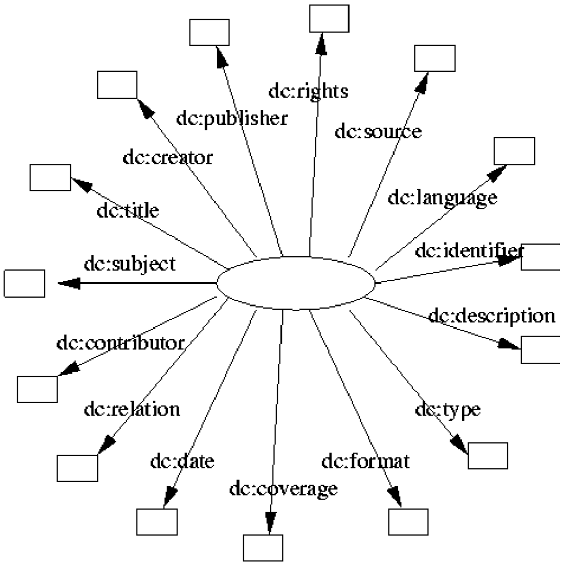

Figure 3. The hedgehog model of the DCMES XML binding

R1: //[@rdf:about] as u1*	⇒	Subject
R2: u1/ as u2*	⇒	Predicate
R3: u2[@rdf:resource] or u2[text()]	⇒	Object

Figure 4. The rules used to map the DCMES XML binding data model onto ECDM's binary relational data model

fixed resource, we can see that there exists an obvious mapping approach from the DCMES XML binding data schema to ECDM's binary relational data model. Moreover, since the DCMES XML binding only uses limited sets of RDF constructs (e.g., rdf:Bag, rdf:Seq, and rdf:Alt are excluded), the mapping becomes more straightforward. In Figure 4 we list three rules used to map the DCMES XML binding data model onto ECDM's binary relational data model. The XML data model is expressed here through XPath location paths using XPath's abbreviated syntax.

Note that in the DCMES XML binding data schema, the value of an element can be either plain text or another resource with a URI. This definition complies with the RDF data model and can be also appropriately expressed using XPath. In fact, based on the above mapping rules, the wrapper program can easily generate the ECDM-based common data view of the native XML repositories containing the DCMES XML binding metadata.

4. Translating Datalog into XPath

According to the usage scenario of the Edutella provider integration architecture, a common behaviour of the provider wrapper programs is to translate RDF-QEL queries into ECDM's internal query language Datalog. In addition, each wrapper program also has a specific behaviour: translating Datalog into the local query languages of the underlying repositories. Since the common behaviour of the wrapper programs has already been discussed in our previous publication [16], here we mainly concentrate on the specific behaviour of the wrapper programs of the native XML repositories, namely, translating ECDM's internal non-recursive Datalog queries, which correspond to the RDF-QEL sets from RDF-QEL1 to RDF-QEL3, into XPath.

Datalog is a non-procedural query language based on Horn clauses without function symbols. The basic construct of Datalog is the Atom, 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. In Datalog, an Atom can be negated and represented as: *NOT P(arg1, arg2)*.

A Datalog program can be expressed as a set of Datalog rules. Each Datalog rule has a general representation as *head :- atom1, atom2,..., atomn*, where *head* is a single positive Atom, and *atom1* to *atomn* are a set of Atoms conjunctively called the body of the Datalog rule. Note that a Datalog rule may include negated Atoms in its body, but generally in some restricted forms [10]. Additionally, the disjunction in Datalog is expressed as a set of rules with the identical head. As an example, in Figure 5 we show a

```
H(X) :- P1(X,U), NOT P2(X, V)
H(X) :- P3 (X,W)
```

H is head; P1, P2, P3 are predicates; X is variable; U, V, W are constants.

Figure 5. A Datalog example query covering conjunctive query, disjunctive query, and query negation

```
//*[@rdf:about and
    (P1 [@rdf:resource=U] or P1 [text()=U]) and
   not (P2 [@rdf:resource=V] or P2 [text()=V])
  ]
  |
//*[@rdf:about and
    (P3[@rdf:resource=W] or P3 [text()=W])
  ]
```

Figure 6. A translated XPath query covering conjunctive query, disjunctive query, and query negation

https://doi.org/10.23106/dcmi.952106977

Datalog example query against a binary relational data model, covering conjunctive query, disjunctive query, and query negation. It corresponds to a typical RDF-QEL3 query defined in Edutella.

In Figure 6 we show the XPath query that is translated from the Datalog query illustrated in figure 5. As we have mentioned, the XPath expressions are based on the DCMES XML binding schema illustrated in Figure 2.

XPath can be seen as a general purpose query notation for addressing and filtering the elements and text of XML documents. A notation indicates the hierarchical relationship between the nodes and is used by a pattern to describe the types of nodes to match. All XPath queries occur within a particular context, which is the single node against which the pattern matching operates. The collections of all elements selected from the current context by XPath queries preserve document order, hierarchy, and identity, to the extent that these are defined. In addition, constraints and branching can be applied to any collection by adding a filter clause to the collection. The filter in XPath is analogous to the SQL WHERE clause, expressed in the form of *[filter pattern]*. The filter pattern evaluates to a Boolean value and is tested for each element in the collection. Any elements in the collection failing the filter pattern test are omitted from the result collection.

In general, each Datalog rule is mapped onto an XPath pattern, based on which a set of elements are selected under a certain context. The conjunctive queries, represented in Datalog by a number of Datalog Atoms contained in a single rule, are translated into sets of filter patterns that are combined together using the XPath Boolean operator *"and"* and are applied to the collection selected by the XPath pattern. The negation of a Datalog Atom can be represented using the XPath Boolean operator *"not"*.

The disjunctive queries, represented in Datalog by a number of Datalog rules with the identical head, are expressed in XPath by a number of patterns combined together using the XPath union operator "|". Multiple union operators can union together sets of collections selected by multiple XPath patterns, also being able to exclude duplicates. In the XPath query listed in Figure 6 we have also used several XPath operators for grouping operation, filtering operation,

Table 1. XPath operators and their precedence orders

1	( )	Grouping
2	[ ]	Filter
3	/ //	Path operations
4	|	Union
5	not ( )	Boolean not
6	and	Boolean and
7	or	Boolean or

Boolean operation, and path operation. These operators are used according to certain precedence orders. In Table 1 we list these operators according to their precedence orders, from the highest to the lowest.

Note that against a binary relational data model, the example query listed in Figure 5 can be seen as a query for Subjects. In fact, in Datalog it is easy to express the queries for Predicates and Objects. Referring to the XPath expressions listed in figure 6, we can easily translate these Datalog queries into the corresponding XPath queries.

5. Adapting local query results into the Edutella common result exchange format

In Edutella, we have defined a RDF-based common result exchange format that represents query results as a set of tuples of variables with their bindings [16]. Whereas the RDF-based repositories can naturally adapt the local query results into Edutella's RDF-based common result exchange format with the support of some powerful local query languages, the native XML repositories can only return XML fragments selected by the XPath expressions. Regarding the DCMES XML binding schema based native XML repositories, the XPath queries can only return sets of whole XML metadata profiles that describe learning resources, since any XPath query must take the entire XML metadata profile as a whole in order to get a virtual binary relational data model against which the XPath query can be operated. Although most of native XML database implementations also provide means for further identifying the underlying elements/attributes of any XML fragments, we decided to use the whole XML metadata profile as the direct output and leave the further processing work on query results to a RDF parser, the Jena RDF Toolkit[6]. An important reason for this choice lies in the fact that the DCMES XML binding metadata profiles themselves are in valid RDF syntax and can be easily handled by RDF parsers. Through using the Jena RDF Toolkit, the query results generated by the native XML repositories can be easily transformed into the RDF model and then naturally adapted into the Edutella common result exchange format. However, it should be noted that in comparison to the query results returned from the RDF-based repositories, the query results from the native XML repositories are a bit redundant.

6. Integrating the LOM/IMS and SCORM XML binding schema based native XML repositories into Edutella

LOM is a learning resource metadata specification proposed by IEEE LTSC (Learning Technology

[6] http://www.hpl.hp.com/semweb/jena-top.html

https://doi.org/10.23106/dcmi.952106977

Standards Committee), which specifies a conceptual data schema that defines the structure of a metadata instance for a learning object [11]. The LOM data schema is actually the basis of some other popular learning resource metadata specifications. For example, the IMS Learning Resource Metadata Specification directly employs the LOM data model and further provides an XML binding for it[7], the SCORM metadata specification extends LOM a little bit and provides a 100% downwards compatibility with it. In the following, our discussion will be based on the native XML repositories containing the LOM/IMS XML binding metadata. The SCORM XML binding schema based native XML repositories can use the same approach to be integrated into the Edutella network.

In comparison to the DCMES XML binding, the LOM/IMS XML binding data schema is much more complex, consisting of nine categories, over 50 metadata entries, and possibly recursive hierarchies (e.g., in the category "Classification"). In general, for such a complex XML schema, it is difficult to generate the ECDM-based common data view using XPath and further apply the same integration approach that is applicable to the DCMES based native XML repositories, as described in section 3, 4, and 5. At present some native XML database implementations begin to support a more powerful query language W3C XQuery [3], which provides a new possibility to generate the ECDM-based common data view of the LOM/IMS based native XML repositories and further apply the same integration approach. However, we argue that the XQuery-enabled new integration approach is more expensive than directly constructing the RDF-based metadata repositories using the LOM/IMS RDF binding [12] and further integrating these repositories into the Edutella network. In fact, for some complex learning resource metadata sets such as LOM/IMS and SCORM, using RDF is a more efficient and more extendible way for representing learning resources. Obviously, such types of repositories can be also more easily and naturally integrated into Edutella.

In order to address the immediate need of integrating the LOM/IMS based native XML repositories into the Edutella network, we employ the approach that relies on the DCMES XML binding as a lingua franca and scale-down maps the LOM/IMS XML binding into the DCMES XML binding through using W3C XSLT (XML Stylesheet Language Transformations) [5]. After the transformation, the integration approach for the DCMES-based native XML repositories can be directly applied to the LOM/IMS based native XML repositories.

As one can imagine, such a transformation from LOM/IMS to DCMES unavoidably loses some information of the original LOM/IMS metadata set. However, we argue that most of lost metadata information are useful only for detailed description of learning resources rather than for the simple discovery of these resources. Thus our integration approach for the LOM/IMS based native XML repositories can still ensure the essential discoverability of the learning resource metadata contained in these repositories. Moreover, the validity of this integration approach is also guaranteed by the common efforts from IEEE LTSC and DCMI (especially the Dublin Core Education Working Group[8]), which have been continuously focused on providing enough interoperability between LOM/IMS and DCMES, as outlined in the MoU[9] (Memorandum of Understanding) between IEEE LTSC and DCMI.

In Table 2 we list the 15 rules used to map LOM/IMS to DCMES [11]. Based on these rules, the transformation from the LOM/IMS XML binding to

Table 2. The rules used to map LOM/IMS to DCMES

LOM/IMS	DCMES
1.1.2:General.Identifier.Entry	DC.Identifier
1.2:General.Title	DC.Title
1.3:General.Language	DC.Language
1.4:General.Description	DC.Description
1.5:General.Keyword or 9:Classification with 9.1: Classification.Purpose equals "Discipline" or "Idea".	DC.Subject
1.6:General.Coverage	DC.Coverage
5.2:Educational.Learning ResourceType	DC.Type
2.3.3:LifeCycle.Contribute.Date when 2.3.1: LifeCycle.Contribute. Role has a value of "Publisher".	DC.Date
2.3.2:LifeCycle.Contribute. Entity when 2.3.1: LifeCycle. Contribute.Role has a value of "Author".	DC.Creator
2.3.2:LifeCycle.Contribute.Entity with the type of contribution specified in 2.3.1: LifeCycle. Contribute.Role.	DC.Other-Contributor
2.3.2:LifeCycle.Contribute. Entity when 2.3.1: LifeCycle. Contribute.Role has a value of "Publisher".	DC.Publisher
4.1:Technical.Format	DC.Format
6.3:Rights.Description	DC.Rights
7.2.2:Relation.Resource.Description	DC.Relation
7.2:Relation.Resource when the value of 7.1:Relation.Kind is "IsBasedOn".	DC.Source

[7] until now IEEE LTSC itself has not yet provided the XML binding for LOM.
[8] http://dublincore.org/groups/education/
[9] http://dublincore.org/documents/2000/12/06/dcmi-ieee-mou/

https://doi.org/10.23106/dcmi.952106977

the DCMES XML binding can be easily accomplished by an XSLT program.

In the native XML repositories, all XML metadata profiles are stored in the separate XML collections according to certain XML schemas. Utilizing an XSLT program, we can easily create a specific collection to store the transformed LOM/IMS metadata profiles, just like creating a database view in RDBs. Moreover, since each XML metadata profile stored in the native XML repositories possesses a unique key to identify itself, we can also retrieve the original metadata profile and get all metadata information.

7. Conclusions

Due to the considerable incomparability between the XML data model and the RDF data model, a generic approach for integrating schema-agnostic native XML repositories into the RDF-based Edutella network was deemed to be unrealistic for our application. This is also attributable to the fact that XPath, the local query language of the native XML repositories, is less powerful and thus incapable of manipulating some complex XML data models to generate their ECDM-based common data view. Moreover, XPath is also incomparable to ECDM's internal query language Datalog and thus incapable of supporting full relational algebra queries. At present, some native XML database implementations begin to support a more powerful query language W3C XQuery, which provides a new possibility to manipulate the native XML repositories and is also more comparable to Datalog (besides providing additional features for handling and creating hierarchical data structures). However, we argue that for schema-agnostic native XML repositories, integrating them into Edutella through using XQuery is more expensive than the integration approach of constructing the RDF-based repositories and then directly integrating them into Edutella. As a matter of fact, for some complex metadata sets such as LOM/IMS and SCORM, using RDF and some high-performance back-end systems is a more efficient and more extendable way for building learning resource metadata repositories. Therefore, although we have found a feasible approach for integrating schema-specific native XML repositories into the Edutella network, which has satisfactorily addressed the current usage and immediate integration need of Edutella by covering most of popular learning resource metadata sets such as DCMES, LOM/IMS, and SCORM, we encourage the application of more RDF-based learning resource metadata repositories in the Edutella network, given the inherent advantages of RDF in distributed P2P settings, such as the easy composability of schemas, as well as the extendability and modularity of distributed RDF metadata.

References

[1] ADL Technical Team, SCORM Specification V1.2, http://www.adlnet.org/index.cfm?fuseaction=scormat

[2] Beckett, D., E. Miller, and D. Brickley, Expressing Simple Dublin Core in RDF/XML, http://dublincore.org/documents/2001/11/28/dcmes-xml/

[3] Boag, S., D. Chamberlin, M. F. Fernandez, D. Florescu, J. Robie, J. Siméon, and M. Stefanescu, XQuery 1.0: An XML Query Language, http://www.w3.org/TR/2002/WD-xquery-20020430/

[4] Brantner, S., T. Enzi, S. Guth, G. Neumann, and B. Simon, UNIVERSAL - Design and Implementation of a Highly Flexible E-Market Place of Learning Resources, in Proc. of the 1st IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2001), Madison, WI, USA, Aug. 2001.

[5] Clark, J., XSL Transformations (XSLT) Version 1.0, http://www.w3.org/TR/xslt

[6] Clark, J., and S. DeRose, XML Path Language (XPath), http://www.w3.org/TR/xpath

[7] DCMI, Dublin Core Metadata Element Set, Version 1.1, http://dublincore.org/documents/1999/07/02/dces

[8] Dhraief, H., W. Nejdl, B. Wolf, and M. Wolpers, Open Learning Repositories and Metadata Modelling, in Proc. of the International Semantic Web Working Symposium, Stanford, CA, USA, Aug. 2001.

[9] Duval, E., E. Forte, K. Cardinaels, B. Verhoeven, R. Van Durm, K. Hendrikx, M. Wentland-Forte, N. Ebel, M. Macowicz, K. Warkentyne, and F. Haenni, The ARIADNE Knowledge Pool System, Communications of the ACM, 44 (5), 2001.

[10] Garcia-Molina, H., J. D. Ullman, and J. Widom, Database Systems: The Complete Book, Prentice Hall, USA, 2001.

[11] IEEE LTSC, IEEE LOM working draft 6.4, http://ltsc.ieee.org/wg12/index.html

[12] IMS Global Learning Consortium, IMS Learning Resource Metadata Specification V1.2.1, http://www.imsproject.org/metadata/index.html

[13] Kokkelink, S., and R. Schwänzl, Expressing Qualified Dublin Core in RDF/XML, http://dublincore.org/doc uments/2002/04/14/dcq-rdf-xml

[14] Lassila, O., and R. R. Swick, Resource Description Framework (RDF) Model and Syntax Specifica-

https://doi.org/10.23106/dcmi.952106977

tion, http://www.w3.org/TR/REC-rdf-syntax

[15] Liu, X., K. Maly, M. Zubair, and M. L. Nelson, Arc - An OAI Service Provider for Cross Archiving Searching, in Proc. of the ACM/IEEE Joint Conference on Digital Libraries, Roanoke, VA, USA, June 2001.

[16] Nejdl, W., B. Wolf, C. Qu, S. Decker, M. Stintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, Edutella: A P2P Networking Infrastructure Based on RDF, in Proc. of the 11th International World Wide Web Conference (WWW 2002), Hawaii, USA, May 2002.

[17] Qu, C., and W. Nejdl, Towards Interoperability and Reusability of Learning Resources: a SCORM-conformant Courseware for Computer Science Education, in Proc. of the 2nd IEEE International Conference on Advanced Learning Technologies (IEEE ICALT 2002), Kazan, Tatarstan, Russia, Sept. 2002.

[18] Papakonstantinou, Y., H. Garcia-Molina, and J. Widom, Object Exchange Across Heterogeneous Information Sources, in Proc. of the 11th International Conference on Data Engineering, Taipei, Taiwan, March 1995.

[19] Teachware on Demand, http://www.teachwareondemand.de

[20] Thompson, H. S., D. Beech, M. Maloney, and N. Mendelsohn, XML Schema Part 1: Structures, http://www.w3.org/TR/xmlschema-1

https://doi.org/10.23106/dcmi.952106977