Searching SCORM Metadata in a RDF-based E-Learning P2P Network Using XQuery and Query by Example

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

The SCORM (Sharable Content Object Reference Model) Metadata Information Model is a reference to the IMS Learning Resource Metadata Information Model, which itself is based on the IEEE 1484.12.1 LOM (Learning Object Metadata) standard. Since the LOM standard possesses a rather complex data model consisting of over 60 metadata entries, searching SCORM metadata has to overcome several challenges to both queries’ run-time performance and efficient, user-friendly query GUI (Graphical User Interface) design. Furthermore, because we start from Edutella, a RDF-based E-Learning P2P (Peer-to-Peer) network, we also have to address the incompatibility between the SCORM Metadata data model and the RDF data model, as well as the syntax incompatibility between XML and RDF. In this paper we propose an approach for searching SCORM metadata in Edutella, which addresses aforementioned challenges through using an XQuery-enabled triple-like SCORM metadata data view and a QBE (Query by Example) based SCORM query GUI. This triple-like data view efficiently bridges the SCORM Metadata data model and the RDF data model, and at the same time greatly improves queries’ run-time performance. The QBE-based SCORM query GUI facilitates the query construction process.

1. Introduction
2. Searching SCORM metadata in Edutella

Edutella employs a wrapper-like architecture to integrate heterogeneous metadata repositories. In figure 1 we illustrate the Edutella integration architecture for SCORM metadata repositories.

The key to such an integration architecture is the Edutella Common Data Model (ECDM), which is shared by all metadata repositories and provides the common data view of the underlying metadata. At its basis, the ECDM is a binary relational data model, which is defined in full compliance with the RDF data model and uses Datalog as its internal query language. Externally, Edutella defines a common query language: RDF Query Exchange Language (RDF-QEL) and a common result exchange format for the whole Edutella network using RDF syntax. The two provide a uniform way to represent queries and query results in Edutella.

According to the Edutella integration architecture for SCORM metadata repositories, searching SCORM metadata in Edutella consists of three phases. First of all, we have to manipulate SCORM metadata stored in native XML databases to generate a SCORM metadata data view, whose underlying data model is compatible with the Edutella Network Datalog-based ECDM. RDF-QEL RDF-based Query Result Exchange Format Wrapper 1 Wrapper n SCORM repository n SCORM repository 1 XQuery function library QBE-based SCORM query GUI RDF-QEL

Figure 1. Edutella integration architecture for SCORM metadata repositories
2.1. Phase 1: generate the triple-like SCORM metadata data view

In the first phase, we propose a triple-like SCORM metadata data view, which can be generated through using XQuery without the loss of any original SCORM metadata information. The triple-like data view has two features. First, its underlying data model is 100% compatible with the ECDM/RDF data model thus can be queried via ECDM’s internal query language Datalog without any problem. Second, it adopts a very simple XML syntax, which can be easily manipulated through using XQuery thus can ensure queries’ run-time performance. In figure 2 and figure 3 we take a SCORM metadata entry: lom.general.catalogentry as an example to demonstrate the generating process of the triple-like data view. Figure 2 shows the graphical data model of this metadata entry in the form of XML Schema. Figure 3 shows an example metadata instance.

In order to generate the triple-like data view, the example metadata instance is represented through a RDF graph, which is then serialized using a simple XML syntax, as illustrated in figure 4 and figure 5. The serialization is realized through using a self-developed XQuery function library.

From figure 5 we can see that taking advantage of a very simple XML syntax, the triple-like data view can be easily manipulated through using XQuery. Moreover, since currently most of native XML databases support the indexing on specific XML elements, the queries’ run-time performance against the triple-like data view could be further improved through the indexing on “//subject”, “//predicate”, and “//object”.

As the underlying data model of the triple-like view is compatible with the ECDM/RDF data model, any SCORM Metadata instance could be represented as a RDF graph.
2.2. Phase 2: develop the wrapper program for SCORM metadata repositories

As previously covered in our discussions, we have focused on the first task of the two sub-tasks in our previous work. The second sub-task is to translate Datalog into XQuery, more precisely, into sets of calls to the self-developed XQuery function library, as illustrated in figure 7. Developing the wrapper program for SCORM metadata repositories consists of two tasks: (1) translating RDF-QEL and various local query languages. In Edutella, we should be also aware that it is not necessarily necessary to achieve 100% compatibility between the SCORM XML binding and the potential SCORM RDF binding, which is expected to become the metadata data view and the potential SCORM RDF binding with the compatibility between the triple-like SCORM metadata data view is not 100% compatible with the IMS Learning Resource Metadata (LOM) data model as: $P(\text{arg}_1, \text{arg}_2)$

\[
\text{Head} : \text{arg}_1, \text{arg}_2, \ldots, \text{lit}\text{eral}_n
\]

\[
\text{Head} : \text{lit}\text{eral}_1, \ldots, \text{lit}\text{eral}_n
\]

Therefore, while handling SCORM XML binding into its potential RDF binding, we should not expect 100% compatibility with the IMS Learning Resource Metadata (LOM) data model, and also because metadata repositories consist of two tasks: (1) translating RDF-QEL into Datalog, which, as ECDM's internal query language, serves as the unique query interface bridging RDF and goes beyond a simple syntactic level. For example, we can translate an example Datalog query into two sub-tasks. The first sub-task is to translate RDF-QEL into XQuery; and (2) transforming XML-based metadata repositories into a uniform way in Edutella. Some efforts might increase the complexity and the run-time cost without the loss of original metadata information. While handling self-contained SCORM XML binding into its potential RDF binding, it is unnecessary for us to bear such sort of additional query overhead.

On the other hand, we should be also aware that it is unnecessary to achieve 100% compatibility between the SCORM XML and RDF binding, remaining the purpose of handling SCORM XML and RDF binding data view and the potential SCORM RDF binding with the compatibility with Dublin Core, Dublin Core Qualifiers, and vCard in the triple-like data view, etc. These efforts include, e.g., using mostly the same namespaces proposed to handle SCORM XML and RDF binding metadata, it is unnecessary for us to include such sort of additional query overhead. Hence, while handling SCORM XML binding into its potential RDF binding, we should not expect 100% compatibility with the IMS Learning Resource Metadata (LOM) data model as: $P(\text{arg}_1, \text{arg}_2)$

\[
\text{Head} : \text{arg}_1, \text{arg}_2, \ldots, \text{lit}\text{eral}_n
\]

\[
\text{Head} : \text{lit}\text{eral}_1, \ldots, \text{lit}\text{eral}_n
\]

While handling self-contained SCORM XML binding into its potential RDF binding, it is unnecessary for us to include such sort of additional query overhead. Hence, while handling SCORM XML binding into its potential RDF binding, we should not expect 100% compatibility with the IMS Learning Resource Metadata (LOM) data model as: $P(\text{arg}_1, \text{arg}_2)$

\[
\text{Head} : \text{arg}_1, \text{arg}_2, \ldots, \text{lit}\text{eral}_n
\]

\[
\text{Head} : \text{lit}\text{eral}_1, \ldots, \text{lit}\text{eral}_n
\]
2.3. Phase 3: search SCORM metadata using QBE

3. Conclusions

4. References

---

Figure 8. QBE-based SCORM query GUI