Proceedings

The Ninth IEEE International Conference on Advanced Learning Technologies

Riga, Latvia
15-17 July 2009

Sponsored by
IEEE Computer Society
IEEE Technical Committee on Learning Technology

Organized by
Faculty of Computer Science and Information Technology, Riga Technical University
Informatics and Telematics Institute, Centre for Research and Technology Hellas

Editors
Ignacio Aedo
Nian-Shing Chen
Kinshuk
Demetrios Sampson
Larissa Zaitseva

CPS Conference Publishing Services

Los Alamitos, California

Washington • Tokyo

IEEE computer society

IEEE Computer Society Order Number E3711
BMS Part Number CFP09261-CDR
ISBN 978-0-7695-3711-5

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: <reprints@computer.org>

Editorial production by Lisa O'Conner

IEEE Computer Society
***Conference Publishing Services* (CPS)**
http://www.computer.org/cps

The Ninth IEEE International Conference on Advanced Learning Technologies

ICALT 2009

Table of Contents

Doctoral Consortium

First European Workshop on ICOPER Best Practice

Fourth International M3C Workshop on Interaction with Content around Curriculum Lifecycle

First International Workshop on Visual Instructional Design Languages and Applications in Technology Enhanced Learning (VIDLATEL)

2009 Ninth IEEE International Conference on Advanced Learning Technologies

Evaluating simple query interface compliance in public repositories

José R. Hilera, Salvador Otón, Antonio Ortiz, Luis De Marcos, José J. Martínez,
José A. Gutiérrez, José M. Gutiérrez, Roberto Barchino
Department of Computer Science
University of Alcalá
Alcalá de Henares, Spain
jose.hilera@uah.es

Abstract

Standards and specifications widely accepted and used lay the foundations to enable and facilitate the interoperability among systems, and the software maintenance and reuse, especially within the scope of learning objects' search systems. One of these standards is the SQI (Standard Query Interface) specification by the European Committee for Standardization in which many search systems, including public ones, are based. This paper analyzes the degree of compliance with this specification by a significant number of learning objects repositories.

1. Introduction

A digital repository of learning objects is a system used to store and distribute learning objects through the Internet, to share them with other users and to facilitate their reuse in a variety of educational activities. Usually, a repository works as a Web portal that can be accessed through a Web based interface, providing a search mechanism. The clients of a repository can be users (teachers or learners); e-learning platforms, such as Content Management Systems (LCMS) or Learning Management System (LMS); or other repositories, in the case of repositories that launch a query to several federated repositories (Fig. 1).

A repository must present to its clients an interface to receive queries and return to the user their results. These results comprise the learning objects (or their metadata records) that satisfy the search criteria. The European Committee for Standardization published the Simple Query Interface (SQI) specification [1] in 2005, in order to facilitate interoperability among public repositories and the applications using them. Since then, the main public repositories have been adapted in order to comply with this specification.

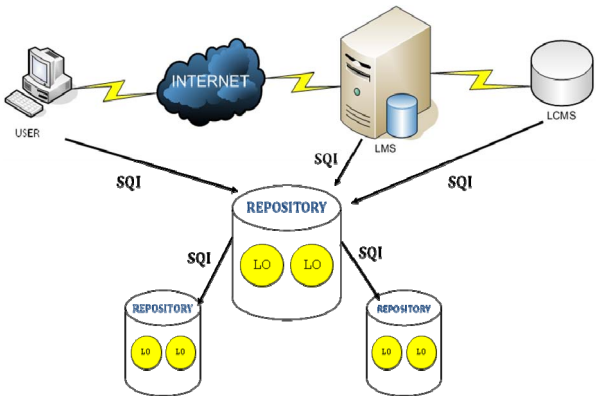

Figure 1. Distributed Repositories of Learning Objects

This paper presents the results of a survey carried out to evaluate the level of compliance with the SQI specification in twelve public available repositories. Section two briefly describes the SQI specification. Section three presents the repositories selected for the assessment and the rationale underlying that selection. Section four presents the results of the research. Finally in section 5 conclusions are drawn.

2. Standard Query Interface

SQI specification consists in a definition of a methods' set that a repository should provide, so that remote systems (clients) can perform queries of learning objects stored within the repository.

SQI identifies thirteen methods to be offered by such systems. They are classified in four categories (Table 1). Four methods are concerned with configuration operations that are previous to the query

978-0-7695-3711-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICALT.2009.89

launch. These include methods for setting up the query language (*setQueryLanguage*), the results' format (*setResultsFormat*), the maximum number of results returned by the query (*setMaxQueryResults*), and the maximum time allowed for a query to return a result (*setMaxDuration*). The last method is especially relevant for asynchronous queries.

Table 1. Classification of SQI methods

CONFIGURATION METHODS	SESSION MANAGEMENT METHODS
setQueryLanguage *(sessionID, queryLanguageID)*	createSession *(userID, password): sessionID*
setResultsFormat *(sessionID, resultsFormat)*	createAnonymousSession *(): sessionID*
setMaxQueryResults *(sessionID, maxQueryResults)*	destroySession *(sessionID)*
setMaxDuration *(sessionID, maxDuration)*	
QUERY METHODS	
SYNCHRONOUS QUERY	**ASYNCHRONOUS QUERY**
setResultsSetSize *(sessionID, resultsSetSize)*	asynchronousQuery *(sessionID, queryStatement, queryID)*
sychronousQuery *(sessionID, queryStatement, startResult): queryResults*	setSourceLocation *(sessionID, sourceLocation)*
getTotalResultsCount *(sessionID, queryStatement): resultsCount*	queryResultsListener *(queryID, queryResults)*

Concerning to session control and management, SQI defines methods for establishing a new session with its credentials (*createSession*), creating anonymous sessions (*createAnonymousSession*), and closing sessions (*destroySession*).

SQI takes into account synchronous and asynchronous queries in repositories. For the former, SQI includes methods for launching queries that return metadata records of learning objects (*sychronousQuery*); to set up in advance the number of results that will be obtained (*setResultsSetSize*), which should be less than that established by the *setMaxQueryResults* method; and to return the number of learning objects that meet the search criteria (*getTotalResultsCount*), even if the returned number is limited by the previous method.

As for asynchronous queries, and besides the query method (*asynchronousQuery*), SQI includes a method that should be previously executed, in order to specify the client system that will be waiting for the query results (*setSourceLocation*), and a method that should be implemented in the system client that will be called by the repository when the search has been executed to return the query results (*queryResultsListener*).

A set of seventeen fault codes is also included in the SQI specification. These faults may occur during methods' invocation, and the repositories should return them for every case (Table 2).

Table 2. SQI Fault codes

CODE	DESCRIPTION
SQI-00000	Undefined
SQI-00001	Method failure
SQI-00002	No source location
SQI-00003	Invalid start result
SQI-00004	Invalid query statement
SQI-00005	Invalid results set size
SQI-00006	Invalid max duration
SQI-00007	Invalid max query results
SQI-00008	Invalid query results
SQI-00009	Query mode not supported
SQI-00010	Results format not supported
SQI-00011	Query language not supported
SQI-00012	Method not supported
SQI-00013	No such session
SQI-00014	No such query
SQI-00015	Wrong credentials
SQI-00016	No more results

3. Public SQI-based repositories

Since the SQI specification emergence many existing and newly created repositories have included in their systems the methods established in this standard. Table 3 shows the repositories considered in our survey. In the web page *http://ariadne.cs.kuleuven.be/SqiInterop/free/SQIImplementationsRegistry.jsp* a record of the SQI compliant repositories is kept.

Table 3. Public SQI-based repositories

REPOSITORY	URL
Acknowledge	http://www.acknowledge.be
Agrega	http://contenidos.proyectoagrega.es
Ariadne	http://www.ariadne-eu.org
EdNA Online	http://www.edna.edu.au
EducaNext	http://www.educanext.org
LACLO-FLOR	http://www.laclo.espol.edu.ec
LORNET	http://www.lornet.org
LORS-SQI	http://www.cc.uah.es
MACE	http://www.mace-project.eu
Merlot	http://www.merlot.org/
Nime	http://nime-glad.nime.ac.jp/en/
OER Commons	http://www.oercommons.org

Although other repositories, such as *CGIAR, Fire, Flickr, Learn eXact, Lion, PlanetDR-URV, Prolearn deliverables, Pubelo, Visualizing cultures* or *Youtube*, are included in this list; they have not been considered in our survey because they are no longer operative or they do not offer information about their utilization.

All this repositories are characterized by the fact that they offer SQI methods as web services, and therefore their use is defined in WSDL files [2]. The access URL to each web service, as well as the location of the WSDL file, is included in the aforementioned web page. Repositories mostly encapsulate methods for session's management in one service and all other methods in a second one, as illustrated in table 4.

Table 4. Web Services containing SQI methods

REPOSITORY	Session management methods	Other methods
Acknowledge	--	SqiTarget
Agrega	SrvSesionesService	SrvSQIService
Ariadne	SessionManagementService	TargetService
EdNA Online	--	EdnaTargetService
EducaNext	SessionManagementService	TargetService
LACLO-FLOR	FSSessionManagementService	FSTargetService
LORNET	SessionManagementService	TargetService
LORS-SQI	SQISessionManagementService	SQITargetService
MACE	--	SqiTarget
Merlot	--	MerlotTargetService
Nime	--	target.php
OER Commons	--	SqiTarget

Listing 1 shows a fragment of code from the Service *TargetService* offered by Ariadne. It can be easily identified the method synchronousQuery.

Listing 1. Extract of *TargetService* service description (WSDL) from Ariadne repository

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
 xmlns="http://schemas.xmlsoap.org/wsdl/"
 xmlns:types="urn:www.cenorm.be/isss/ltws/wsdl/
  SQIv1p0"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 targetNamespace="urn:www.cenorm.be/isss/ltws/wsdl/
  SQIv1p0">

<types>
 <xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:www.cenorm.be/isss/ltws/wsdl/
   SQIv1p0">
  <xsd:element name="synchronousQuery">
   <xsd:complexType>
    <xsd:sequence>
     <xsd:element name="targetSessionID"
      type="xsd:string"/>
     <xsd:element name="queryStatement"
      type="xsd:string"/>
     <xsd:element name="startResult"
      type="xsd:int"/>
    </xsd:sequence>
   </xsd:complexType>
  </xsd:element>
  <xsd:element name="synchronousQueryResponse">
   <xsd:complexType>
    <xsd:sequence>
     <xsd:element name="synchronousQueryReturn"
      type="xsd:string"/>
     </xsd:sequence>
   </xsd:complexType>
  </xsd:element>
 </xsd:schema>
</types>

<message name="synchronousQueryRequest">
 <part name="synchronousQuery"
  element="types:synchronousQuery"/>
</message>
<message name="synchronousQueryResponse">
 <part name="queryResults"
  element="types:synchronousQueryResponse"/>
</message>

<operation name="synchronousQuery">
  <input message="types:synchronousQueryRequest"/>
  <output message="types:synchronousQueryResponse"/>
  <fault name="SQIFault" message="types:SQIFault"/>
 </operation>

<service name="TargetServiceBinding">
 <soap:address
  location="http://ariadne.cs.kuleuven.be/AWS/
  services/TargetService"/>
</service>

</definitions>
```

4. SQI compliance analysis

In order to do a comparative study about repositories compliance of the SQI specification, the web services offered by the different repositories must be analyzed, so as to confirm whether the SQI methods have been incorporated and work as expected. This task can be done easily by developing and application able to access to every web service and call every method. A few freeware programs, such as *SQITest* (http://ariadne.cs.kuleuven.ac.be/SQI/SQITest.jnlp), may help to perform this task. Learning objects federated search systems that work over distributed repositories with SQI interface may also be used. These systems let the user select the remote repositories in which the search will be extended.

An even more flexible system is *LORS-SQI Federated Search* [3], since it also allows users to federate or add links to new repositories, just by indicating the URL that connects to the web services containing the SQI methods (figure 2 shows how *Ariadne* system is added to that system).

Tables 5 to 7 show the results obtained from the analysis of the twelve repositories. As it can be noticed in table 5, all the studied repositories implement the four SQI configuration methods, except for the last repository (*Nime*) which only implements two. However, session methods are implemented by only one half of the repositories; the rest offer a fixed value session identifier (*SessionID*).

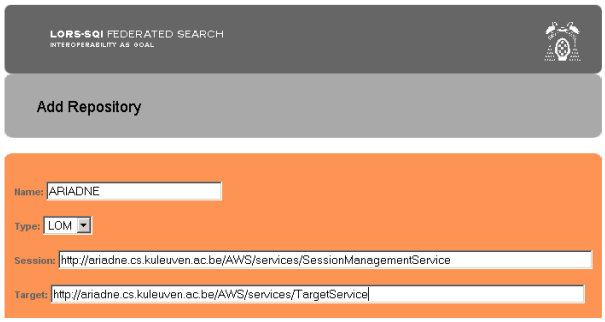

Figure 2. Federating a SQI-based repository in LORS-SQI

In relation to search methods (table 6), all the repositories offer the three synchronous search methods (except for two of them that only offer two). However, only four repositories provide asynchronous searches.

Table 5. Configuration and Session Management methods implemented

REPOSITORY	Configuration methods	Session Management methods
Agrega	Yes	Yes
Ariadne	Yes	Yes
EducaNext	Yes	Yes
LACLO-FLOR	Yes	Yes
LORNET	Yes	Yes
LORS-SQI	Yes	Yes
Acknowledge	Yes	No (SessionID = *vsql*)
EdNA Online	Yes	No (SessionID = *ariadnekey*)
MACE	Yes	No (SessionID = *vsql*)
Merlot	Yes	No (SessionID = *vsql*)
OER Commons	Yes	No (SessionID = *vsql*)
Nime	Yes (*)	No (ID=*Static_Session_Ariadne*)

(*) Methods *setMaxQueryResults* and *setMaxDuration* are not implemented

Table 7 summarizes the results. It is important to underline that only one of the repositories (*Ariadne*) implements all the methods established in the SQI specification, that is, thirteen methods. Most other repositories offer the mean values of 10 methods, so it can be stated that they make up 77% of the compliance. *Nime* is the repository with a lower compliance level, since it only incorporates five out of thirteen methods.

We have not included comparisons related to other issues of SQI, such as the supported query language which is set with the *setQueryLanguage* method, or the metadata records returned by the query and set by the *SetResultsFormat* method, or the SQIFault codes generated; as for all cases, the evaluated repositories use the same query language (VSQL [4]) and the same metadata format (IEEE LOM [5]), and all of them generates the seventeen fault codes established in SQI. Listing 2 shows a fragment of the obtained result in LOM format when sending the following VSQL query to the Ariadne repository:

```
"<simpleQuery>\n\n<term>JAVA</term></simpleQuery>"
```

Table 6. Query methods implemented.

REPOSITORY	Synchronous Query methods	Asynchronous Query methods
Ariadne	Yes	Yes
EdNA Online	Yes	Yes
Merlot	Yes	Yes
LACLO-FLOR	Yes (*)	Yes
Acknowledge	Yes	No
Agrega	Yes	No
EducaNext	Yes	No
LORNET	Yes	No
LORS-SQI	Yes	No
MACE	Yes	No
Nime	Yes	No
OER Commons	Yes (*)	No

(*) Method *setResultsSetSize* is not implemented

Table 7. Summary of SQI methods implemented in repositories

REPOSITORY	CONF.	SES.	AS.QRY	S.QRY	TOTAL
Ariadne	4	3	3	3	13 (100%)
LACLO-FLOR	4	3	3	2	12 (92%)
Acknowledge	4	0	0	3	10 (77%)
Agrega	4	3	0	3	10 (77%)
EducaNext	4	3	0	3	10 (77%)
LORNET	4	3	0	3	10 (77%)
LORS-SQI	4	3	0	3	10 (77%)
Merlot	4	0	3	3	10 (77%)
EdNA Online	4	0	3	3	10 (70%)
MACE	4	0	0	3	7 (54%)
OER Commons	4	0	0	2	6 (46%)
Nime	2	0	0	3	5 (38%)

(*) Types of methods: CONF.= Configuraton, SES.=Session, AS.QRY=Asynchronous query, S.QRY= Synchronous query

5. Conclusions

Broadly speaking, compliance with standards and specifications widely accepted assure interoperability among people, products and people and products. As for learning objects' repositories, the SQI specification has facilitated considerably the interoperability among repositories, and among them and the search engines, since it is no longer necessary to implement different search mechanisms for different repositories; the only requirement is to register the location of the web services including the SQI methods of every repository, and from that point, queries will be

executed in the same way over all of them; so search systems development is lightened to a great degree.

Listing 2. Extract of a query result about "JAVA" returned from Ariadne repository

```
<results>
 <lom>
  <general>
   <identifier>
    <catalog>ARIADNE</catalog>
   </identifier>
   <title>
    <string language="nl">
     BVP: BASISBEGRIPPEN PROGRAMMEREN MET JAVA
    </string>
   </title>
  </general>
  <lifeCycle>
   <contribute>
    <role>
     <source>LOMv1.0</source><value>author</value>
    </role>
    <entity><![CDATA[BEGIN:VCARD FN:HENK OLIVIE
     N:OLIVIE;HENK VERSION:3.0 END:VCARD]]>
    </entity>
   </contribute>
  </lifeCycle>
  <metaMetadata>
   <language>nl</language>
  </metaMetadata>
  <technical>
   <format>application/pdf</format>
   <size>392192</size>
   <location>http://ariadne.cs.kuleuven.be/silo2006/
    ShowDescription.do?ID=BLKLKP1088</location>
   <requirement>
    <orComposite>
     <type>
      <source>ARIADNE</source>
      <value>operating system</value>
     </type>
     <name>
      <source>ARIADNE</source>
      <value>Multi-OS</value>
     </name>
    </orComposite>
   </requirement>
  </technical>
  <educational>
   <intendedEndUserRole>
    <source>LOMv1.0</source><value>learner</value>
   </intendedEndUserRole>
  </educational>
  <educational>
   <interactivityType>
    <source>LOMv1.0</source>
    <value>expositive</value>
   </interactivityType>
  </educational>
  <rights>
   <cost>
    <source>LOMv1.0</source><value>no</value>
   </cost>
  </rights>
  <classification>
   <taxonPath>
    <source>
     <string language="en">ARIADNE</string>
    </source>
    <taxon>
     <entry>
      <string language="en">Informatics/Information
       Processing</string>
     </entry>
    </taxon>
    <taxon>
     <entry>
      <string language="nl">programming</string>
     </entry>
    </taxon>
   </taxonPath>
  </classification>
 </lom>
</results>
```

Taking into account the data obtained in the survey, we can conclude that most of the repositories are not fully compliant (100%) with SQI specification, but the degree of compliance is quite acceptable, with a 70% implementation level, over all the SQI methods, in most cases. It can be also inferred that main lacks in compliance are concerned with asynchronous search mechanisms that SQI defines, due to the inherent complexity that these methods imply.

6. References

[1] *SQI: Simple Query Interface*. European Committe for Standarization, 2005. ftp://ftp.cenorm.be/PUBLIC/ CWAs/e-Europe/WS-LT/cwa15454-00-2005-Nov.pdf.

[2] *Web Services Description Language (WSDL)*, World Wide Web Consortium, 2001. http://www.w3.org/TR/wsdl.

[3] Otón, S., Ortiz, A., Hilera, J.R., Martínez, J.J., Barchino, R., Gutiérrez, J.M., Gutiérrez, J.A., De Marcos, L., "The Integration of SQI in a Reusable Learning Objects System: Advantages and Disadvantages", *Proceedings of IIWAS2008*, ACM, Austria, 2008, pp. 600-603.

[4] *Query Languages stable*, ARIADNE Foundation for the European Knowledge Pool, 2006. http://ariadne.cs.kuleuven.be/lomi/index.php/QueryLanguages_stable.

[5] *IEEE 1484.12.1 Learning Objects Metadata (LOM)*, Institute of Electrical and Electronics Engineers, 2002.

Acknowledgments

This research is co-funded by the Spanish Ministry of Industry PROFIT program (grant TSI-020501-2008-157) and Castilla-La Mancha autonomous community educational innovation cooperation program (grant EM2007-004). Authors also want to acknowledge support from the TIFyC research group.