

Spinning Interoperable Applications for Teaching & Learning using the Simple Query Interface

Frans van Assche

European Schoolnet, Rue de Trèves 61, B-1040 Brussels, Belgium
Tel: +32 2 790 7575
frans.van.assche@eun.org

Erik Duval

Katholieke Universiteit Leuven, Dept. Computerwetenschappen
Celestijnenlaan 200 A, B-3001 Leuven, Belgium
Tel: +32 16 327066
erik.duval@cs.kuleuven.ac.be

David Massart

European Schoolnet, Rue de Trèves 61, B-1040 Brussels, Belgium
Tel: +32 2 790 7575
david.massart@eun.org

Daniel Olmedilla

L3S Research Center and Hannover University, Deutscher Pavillon, Expo plaza 1, D-30539 Hannover, Germany
Tel: +49 511 762.9741
olmedilla@l3s.de

Bernd Simon and Stefan Sobernig

Vienna University of Economics and Business Administration
Institute of Information Systems & New Media, Augasse 2-6, A-1090 Vienna, Austria
Tel: +43 1 31336 4443
bsimon@wu-wien.ac.at
ssobernig@wu-wien.ac.at

Stefaan Ternier

Katholieke Universiteit Leuven, Dept. Computerwetenschappen
Celestijnenlaan 200 A, B-3001 Leuven, Belgium
Tel: +32 16 327066
stefaan.ternier@cs.kuleuven.ac.be

Fridolin Wild

Vienna University of Economics and Business Administration
Institute of Information Systems & New Media, Augasse 2-6, A-1090 Vienna, Austria
Tel: +43 1 31336 4443
fwild@wu-wien.ac.at

Abstract

The Web puts a huge number of learning resources within reach of anyone with Internet access. However, many valuable resources are difficult to find due to the lack of interoperability among learning repositories. In order to achieve interoperability, implementers require a common query framework. This paper discusses a set of methods referred to as Simple Query Interface (SQI) as a universal interoperability layer for educational networks. The methods proposed can be used by a source for configuring and submitting queries to a target system and retrieving results from it. The SQI interface can be implemented in a synchronous or an asynchronous manner and is independent of query languages and metadata schemas. In this paper SQI's universal applicability has been evaluated by more than a dozen implementations demonstrated in three different case studies. SQI has been finalized as a standard in the CEN/ISSS Learning Technologies Workshop. Latest developments of SQI can be followed at <http://www.prolearn-project.org/lori/>.

Keywords

Educational mediators, Learning object metadata, Standards, Interoperability, Brokerage, Application Program Interface, Learning Repositories, Querying, Web Services

Introduction

Performing a search constitutes a typical process for everyone involved in teaching and learning. For example, pupils complement things they have learned in school by looking for learning resources when working on a homework assignment. Teachers aim at enriching their courses with external resources in order to reduce their own workload through re-use while increasing the effectiveness of their course at the same time. Employees responsible for personnel development try to find a cost-efficient but effective learning arrangement that fill a particular competence gap.

The Web puts a huge number of resources within reach of anyone with Internet access. However, many valuable resources are difficult to find, because they are hidden in the closed and proprietary worlds of brokerage platforms, learning (content) management systems, streaming media servers and online collaboration tools. These systems - commonly referred to as learning object repositories - hold information on learning objects such as courses, online tutorials, lecture notes, electronic textbooks, tutoring sessions, quizzes, etc.

Learning object repositories lacking interoperability create three major drawbacks for its users:

1. *The number of learning objects accessible is limited* to those held in the local repository. Teachers, learners, and learning managers might miss the latest critical developments due to the fact that the existence of an external learning object is hidden from them.
2. *The way users can search for learning objects is restricted* by proprietary semantics or by document-centred interfaces offered by today's search engines. While the proprietary semantics can lead to effective search on the local repository, the lack of semantic alignment with the outside world reduces the usability. On the other hand general search engines hardly provide means for effective searches. For example, it is difficult to satisfy a search such as 'List me all courses that improve my negotiation skills and are offered in Brussels between 15th of October and 30th of November and that are frequently visited by employees of the pharmaceutical industry' using state-of-the-art technology.
3. *Context information on the learning object is limited to the local environment* hereby reducing the possibilities for deploying effective recommendation mechanisms. The evolution of web search engines has largely benefited from re-using context information (Page et al., 1998), e.g. by investigating the hyperlink structure of web documents ('who links to whom?'). This kind of information is not accessible for closed educational systems, which is especially a drawback for making a highly complex decision such as selecting the right learning object. The context of learning objects – especially information on its usage ('who has successfully used which learning object?') – constitutes highly valuable information that is hidden in the closed and proprietary worlds of today's repositories.

Beyond the individual level, institutions are increasingly demanding interoperability in the educational domain in order to realize scenarios, such as

- exchanging learner performance records between applications,
- integrating course descriptions from heterogeneous providers or sharing them among multiple applications (e.g., learning management systems and performance measuring tools),
- delivering learning using a multi-application learning environment (content management systems and video conferencing tools), or
- comparing course evaluation data between different sites using different evaluation tools.

On the institution level, interoperability is a driving force for integrating organisations, improving decision making processes, or expanding electronic distribution channels.

In this paper we propose a common query interface as one part of the solution for making educational systems interoperable and exploring the hidden educational web (Raghavan & Garcia-Molina, 2001). The remainder of this paper starts with an introduction to the various aspects of interoperability. The sub-subsequent section is devoted to the specification of a query service, but also reviews related work in this area. We continue by presenting three case studies established in different educational sub-domains that demonstrate interoperability. The paper concludes with a discussion of the status quo and outlines future work in the area.

Within its focus on interoperability for querying, this paper targets architects of educational networks, managers of learning object repositories, stakeholders in learning object re-use, as well as researches in web services and system interoperability.

Aspects of Interoperability

Interoperability can be defined as ‘the ability of two or more systems or components to exchange information and to use the information that has been exchanged’ (IEEE, 1990). This definition comprises three primary aspects that constitute interoperability, namely, *exchange* (i.e., communication) and *use* (i.e., correct interpretation) of *information* (i.e., data plus meaning).

Information emerges from data through processing and utilisation. To ensure the correct interpretation in communication, as denoted in the definition, a common semantic model is required. This semantic model – also referred to as ontology – should specify the properties of the learning object accessible within the repository (Wiederhold et al., 1992). Each declaration of a learning object property constitutes an ontological commitment to use the defined term in interactions with the repository.

The sum of all ontological commitments is reflected in a *common schema*. In the last couple of years institutions and associations such as ISO, IEEE, IMS, ADL, CEN/ISSS, or the HR-XML consortium have given rise to an inflationary number of standards and specifications that can be used as the basis for a common schema. In the case of learning object descriptions the IEEE LOM standard is the predominant reference for developing schemas. While in some cases LOM (and its standard XML binding) can be used as such, in other cases application specific adaptations are necessary (Quemada & Simon, 2003; Najjar et al., 2004; Rivera et al., 2004). These specific instantiations of the LOM model are referred to as LOM Application Profiles.

On a lower layer, *data model & format* choices need to be specified, as different options apply for data representation format (i.e., the data model) and character set encodings. Typically, XML and relational databases with UTF-8 support are to be found here.

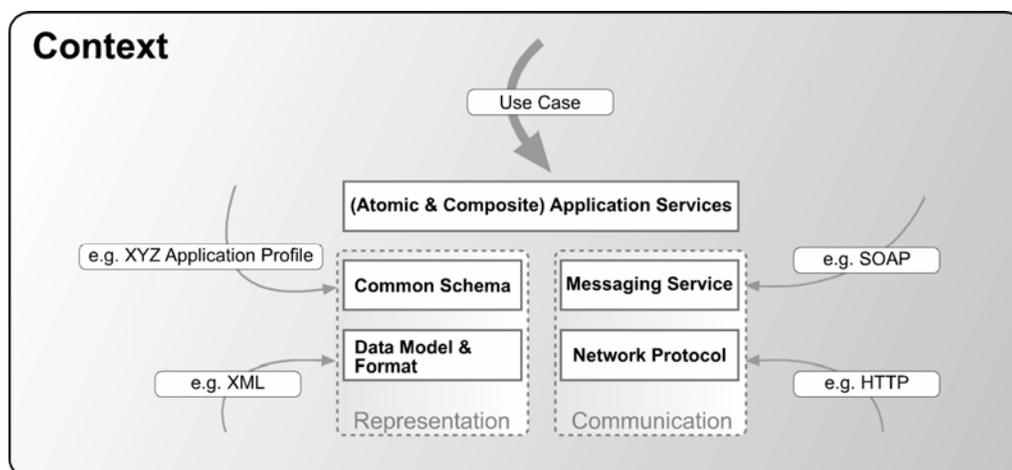


Figure 1. Interoperability Framework

In order to realize system to system communication a common *messaging service*, enabling repositories to interact, is required. XML RPC, Java RMI, and WSDL/SOAP are examples for such a messaging infrastructure. The messaging service is based on lower level *network protocols* such as TCP/IP, or HTTP.

Based on the two pillars ‘communication’ and ‘representation’, interoperable applications can be built (see Figure 1). These applications might take advantage of interoperable atomic services that are aggregated in composite services. Atomic services might be needed, for example, to agree on a common procedure for uniquely identifying learning objects. Another example for an atomic service is related with authenticating users and repositories, or with creating and managing access control rights. Higher level services that create interoperable applications are, for example, an indexing service, a harvesting service or a query service. An indexing service is a kind of replication service that allows repository A to *push* learning object metadata to repository B. It supports distributed maintenance of metadata through insert, delete or update operations. A harvesting service is a service, where repository A *pulls* metadata from a repository B. The query service allows repository A to search repository B for suitable learning objects, so the metadata transferred matches a specific query. A contracting service assigns access rights to a learning object stored at a remote repository. A delivery

service interacts with the repository where the learning object is stored and delivers an electronic learning object to the end user.

The design choices made when implementing the applications are driven by the use cases the applications are designed for. Examples of use cases are ‘Course selection for the purpose of providing certain qualifications’ or ‘Provision of a federated search in order to satisfy teachers in their need for new course material’. Use cases take place in a particular context. The context can be abstracted via the domain the application is designed for. Educational sub-domains such as higher education, vocational training or schools influence interoperability along all aspects. Technological choices made in the context also influence the concrete realization of interoperable application services.

The Simple Query Interface – An Interoperable Application Service for Querying

While for many aspects of the Interoperability Framework wide-spread solutions do exist, the educational domain is still lacking interoperable application services that take advantage of standards such as IEEE LOM, WSDL/SOAP, and XML. In this section we propose an application service for querying that can be used for federating heterogeneous learning object repositories.

An application service for querying needs to specify a number of methods a repository can make available in order to receive and answer queries from other applications. To distinguish the requestor from the answering system in our scenarios, the term ‘source’ is introduced in order to label a system which issues a search (the source of the query). The term ‘target’ labels the system which is queried (the target of the query). Alternatively, the ‘source’ can also be referred to as ‘requestor’ and ‘target’ as ‘provider’.

Metadata can be stored using different means, such as file-based repositories, (possibly distributed) relational databases, XML databases, or RDF repositories. In order to make learning repositories interoperable, not only a common interface needs to be defined, but also a common query language together with a common results format for learning object descriptions needs to be agreed on.

The query service is used to send a query in the common query language to the target. Next, the query results, represented in the common results format, are transported to the source. On the implementation level, wrappers may need to be created to convert a query from a common query language X to a local query language Y and transform the query and the query results from a proprietary format to a common one and vice-versa.

Figure 2 illustrates an exchange process, where Learning Repository A (the source) submits a query to Learning Repository B (the target). It is assumed that both systems have agreed upon a common query language beforehand. The concepts used in the query statement are part of a common (query) schema. At Repository B, the interface component might need to transfer the query from the common query language to the local one. Also some mappings from the common to the proprietary schema might be required before submitting the search. This task is performed by a wrapper component. Once the search has yielded results, the results set is forwarded to the source, formatted according to a common results format.

The collaborative effort of combining highly heterogeneous repositories has led to the following requirements:

- The application service needs to be neutral in terms of results format, query schema and query language. The repositories connecting can be of highly heterogeneous nature: therefore, no assumptions about these components of the interoperability framework can be made.
- The application service needs to support synchronous and asynchronous queries in order to allow the application to be deployed in various use cases and heterogeneous network architectures.
- The application service needs to support, both, a stateful and a stateless implementation.
- The application service shall be based on a session management concept in order to separate authentication issues from query management, but also for providing an anchor point for implementing simple business models for access control.

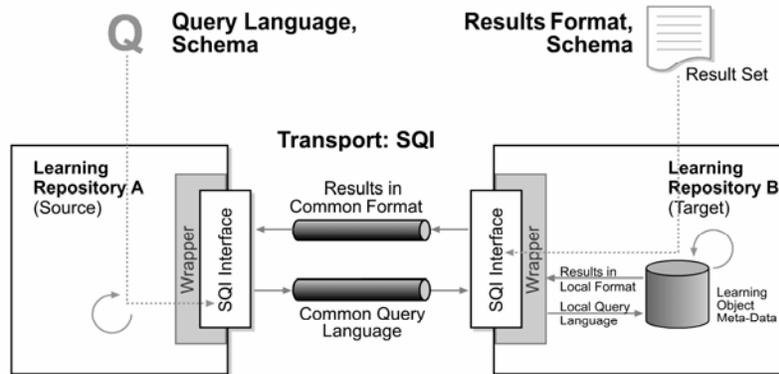


Figure 2. Communication between two Repositories

In addition, the design of the application service itself is based on the following design principles:

- Command-Query Separation Principle,
- Simple Command Set and Extensibility.

The following sub-sections describe each of the above mentioned requirements and design principles of the query service in more detail. Since one design objective of the query service is to keep the specification simple and easy to implement, we decided to call the application service 'Simple Query Interface (SQI)'.

Query Language and Results Format

In order to make use of SQI to implement full query functionality, the application service needs to be complemented with agreements about:

- the set of attributes and vocabularies that can be used in the query,
- the query language and its representation, and
- the representation of (a list of) learning object metadata that satisfy the query.

Driven by the Interoperability Framework of Section 2, SQI is agnostic on these issues: Any agreement between two or more repositories is valid for SQI. Such agreements can, for example, be expressed via XML schemas or RDF schemas. Although SQI does not directly contribute to overcome the differences of the various paradigms in metadata management (Z39.50, XML-based approaches, RDF community), it aims to become an independent specification for all open educational repositories.

Synchronous and Asynchronous Queries

SQI can be deployed in two different scenarios: In the synchronous scenario, the source, after having sent a query, keeps waiting for an answer from the target or a time-out. Results retrieval is therefore initiated by the source through the submission of the query and through other methods allowing the source to access the query results.

In the asynchronous scenario, results transmission is target-initiated. Whenever a significant amount of matching results is found, these results are forwarded to the source by the target. To support this communication the source must implement a results listener. The source must be able to uniquely identify a query sent to a particular target (even if the same query is sent to multiple targets). Otherwise the source is not able to distinguish the search results retrieved from various targets and/or queries previously submitted to a target.

Please note that the asynchronous query mode does not require an asynchronous handling on the messaging layer. It can also be implemented by two synchronous functions at the source and the target, respectively.

A query interface operated in synchronous mode can perform multiple queries per session (even simultaneously). In case of an asynchronously operated query interface, the source provides a query ID that allows it to link

incoming results to a submitted query (the source might query many targets and each target might answer to a query by returning more than one result to the source). Multiple queries can also be active within a session in asynchronous query mode.

Session Management

The application interfaces make abstraction from authentication and access control issues. However, there is a need to authenticate the source in order to allow a target, for example, to link query policies to a source repository. For instance:

- Repository A is allowed to query Repository B without any limitations,
- Repository C is only allowed to retrieve 1000 query results per day from Repository D at a maximum.

Ideally, authentication is performed only once for a series of interactions. To accomplish this, a session token needs to be returned after successful authentication that can be used to identify the system in the subsequent communication.

Session management needs to be understood as a higher-layer management of configuration settings and authentication. The session ID serves as a mandatory element in the application interfaces in order to identify the requestor/source in all query commands.

Therefore, the SQI is based on a simple session management concept. A session has to be established before any further communication can take place. This specification separates query management and processing from authentication and authorization.

The source establishes a session at the target and uses the Session ID, which is obtained from the target, to identify itself during communication. Authentication does not need to be based on username/password declarations or credentials, since also anonymous sessions can be created. The specification introduces an incomplete list of possible means for establishing a session for the communication between two systems. Once a session has been established, the source has the right to communicate with the target. In order to establish a session, a user name and password or any other credential may be required. The identification of a source repository can prevent candidate target repositories from opening up their systems to unknown partners, and enables query policies.

A session is valid until it is destroyed. Hence, it continues to be active after a query has been executed. Alternatively, a session times out when no communication takes place during e.g. 30 minutes. However, a session might be valid much longer than 30 minutes and sometimes might even require manual destruction.

The specification assumes the use of secure authentication, and encryption mechanisms such as those provided by state-of-the-art technology (e.g., SSL).

Stateful and Stateless Communication

Stateful and stateless are attributes that describe whether repositories are designed to keep track of one or more preceding events in a given sequence of interactions. Stateful means that the target repository keeps track of the state of interaction, for example, by storing the results of a previously submitted query in a cache. Stateless means that there is no record of previous interactions and that each interaction request can only be handled on the basis of the information that comes with it. The SQI specification allows implementers to opt for a stateful or a stateless approach.

Command-Query Separation Principle

SQI design adheres to the principle of Command/Query Separation, a design convention originally devised for the Eiffel programming language. It states that every method should either be a command that performs an action (also referred to as ‘procedure’), or a query that returns data to the caller (a ‘function’), but not both. More formally, methods should return a value only if they are referentially transparent, i.e. they can be expected to leave an object’s state unchanged and hence cause no side-effects. This convention is valuable for assertion-

based programming, a building block of the design-by-contract approach, the latter being essential with respect to service-oriented architectures in general (Henney, 2000).

Simple Command Set and Extensibility

In order to make the interface easily extensible an approach, minimizing the number of parameters of the various methods rather than the number of methods is adopted. Variations of the interface (e.g., a separation between common query schema and common results format), can easily be introduced by adding a new function (e.g., `setSupportedQuerySchema`) while no change in the already implemented methods is needed. Hereby, backwards compatibility can be more easily maintained.

As a result, additional methods for setting query parameters like maximum duration and maximum number of returned search results were introduced. This design choice leads to simpler methods, but the number of interdependent methods is higher. However, default values can be used for many of these query parameter configuration methods.

Overview of SQI Methods

Table 1 provides an overview of the various methods that are described below from a workflow perspective. A detailed description of the methods is provided in the specification (CEN/ISSS, 2005).

First, the source needs to create a connection with the target, for example by using `createAnonymousSession`. Once a session has been established, the query interface at the target awaits the submission of a search request. In addition, a number of methods allow for the configuration of the interface at the target. Query parameters such as:

- the query language (`setQueryLanguage`),
- the number of results returned within one results set (`setResultsSetSize`),
- the maximum number of query results (`setMaxQueryResults`),
- the maximum duration of query execution (`setMaxDuration`),
- and the results format (`setResultsFormat`)

can be set with the respective methods. The parameters set via these methods remain valid throughout the whole session or until they are set otherwise. If none of the methods is used before the first query is submitted, default values are assumed. The specification provides default values for `MaxQueryResults`, `MaxDuration`, and `ResultsSetSize`.

Table 1. SQI Methods

| |
|---|
| <i>Session Management</i> |
| <code>createSession</code> |
| <code>createAnonymousSession</code> |
| <code>destroySession</code> |
| <i>Query Parameter Configuration</i> |
| <code>setResultsFormat</code> |
| <code>setMaxQueryResults</code> |
| <code>setMaxDuration</code> |
| <i>Synchronous Query Interface</i> |
| <code>setResultsSetSize</code> |
| <code>synchronousQuery</code> |
| <code>getTotalResultsCount</code> |
| <i>Asynchronous Query Interface</i> |
| <code>asynchronousQuery</code> |
| <code>setSourceLocation</code> |
| <code>queryResultsListener</code> |

Next, the source submits a query, using either the `asynchronousQuery` or the `synchronousQuery` method. The query is then processed by the target and produces a set of records, referred to as results set. The query is expressed in a query language identified through a query parameter. In the query, reference to a common schema

might be made. In synchronous mode the query results are directly returned by the synchronousQuery method. The getTotalResultsCount method returns the total number for matching metadata records found by the target operating. In case of an asynchronously operated query interface the queryResultsListener method is called by the target to forward the query results to the source.

In order to report abnormal situations (e.g., erroneous parameters or inability to carry out an operation), an SQIFault is provided, which can be thrown by all the SQI methods. A system of fault codes permits to document those abnormal situations.

Related Work

In recent years, researchers from various fields have become increasingly interested in interoperability standards for information systems in the (wider) context of Technology Enhanced Learning. In order to dissociate the efforts leading to the shaping and the adoption of SQI from comparable work, we provide a comparison on several dimensions: motivation and constitutive uses cases of the projects under consideration, application services offered, the respective extent of schema and representation handling, and messaging services and networking protocols used. The following account is inspired by the Interoperability Framework presented in Section 2.

Interoperability research is driven by initiatives of different levels of granularity. We consider most of these orthogonal to our efforts, therefore differences manifests primarily at the levels of motivation, use case and scope. OpenURL (NISO, 2002; <http://library.caltech.edu/openurl/>), for example, and the underlying framework emerged from the field of digital libraries, or scholarly journal repositories to be more specific. The OpenURL standard aims at realizing decentralized resource resolution architectures that are adaptive to the human seekers' organisational context and open to be extended in terms of repositories published and resolution services offered across organizational boundaries. OpenURL adopts a strict separation of referencing resources, i.e. creating annotated and globally valid resource identifiers, and resolving these references by organization-specific resolver facilities into, for instance, seeker-tailored library services (Van de Sompel & Beit-Arie, 2001). This highlights the major difference between OpenURL and SQI: While OpenURL targets identifier creation and user-centric identifier resolution across heterogeneous repositories, SQI realizes lookup infrastructures. An initiative of similar granularity, the Content Object Repository Discovery and Resolution Architecture (Rehak, 2005) is interested in providing a reference model for federating community- or organization-specific repositories and for pyramids of repository federations ('Federated CORDRA'). The model describes centralized services for discovery (e.g., registries for content, affiliated repositories, system specifics) and resolving resource references. The latter refers to a unique identifier infrastructure based on namespace hierarchies in and across federations (Rehak, 2005). Efforts such as OpenURL and CORDRA are complementary to SQI as the latter – for instance – might be re-used as discovery standard in a (federated) CORDRA network or for querying repositories that refer to their learning objects by OpenURL identifiers (a more in-depth investigation of the applicability of SQI to CORDRA including a comparison of SQI-related approaches has recently been conducted by researchers involved in the CORDRA project (CORDRA, 2004).

As for lookup standards or other initiatives at a comparable granularity level, Google Web API (<http://www.google.com/apis/>) and Amazon E-Commerce Service (www.amazon.com/gp/aws/landing.html) have recently gained momentum. In principle, while designed for the specific needs of exclusively publishing the respective set of services in a (web) service-oriented manner their interaction standard models could be re-used partly for more generic scenarios. Reacting to their emergence and in continuation of the Z39.50 family of standards, the development of a Search/Retrieve Web Service (SRW) and a Search/Retrieve URL (SRU) for learning object repositories was initiated by the Z39.50-International Next Generation (ZING) community (Board, 2004). The domain of application of SRW/U-compliant services is still centred around digital libraries and research repositories, mainly by providing backward or gateway compatibility to existing Z39.50 nodes, although the protocols and available tool kits are adaptable to other contexts as well. SQI differs in this respect as it has already been adopted in contexts involving metadata about learning activities, as contrasted to learning material. SQI also meets the requirements for cooperating with providers other than digital libraries and scholarly repositories such as corporate providers of professional training (see Section 4.3).

Another relevant initiative is the IMS Digital Repository Interoperability (DRI) Specification (IMS, 2003). It provides recommendations for realizing interoperability with respect to most common repository functions identified at a high level of abstraction, delegating many design choices to implementers. In contrast to CORDRA, it emphasizes types of interaction or lookup functions while CORDRA provides an architectural

perspective. The IMS DRI specification therefore served as a skeleton, for example, for the EduSource project (Hatala et al., 2004) and its underlying communication protocol – referred to as EduSource Communication Layer (ECL) protocol. The Open Knowledge Initiative (OKI) develops an open system to support on-line training. It provides a collection of Open Service Interface Descriptions (OSIDs) that are conceptualized as contracts in service-oriented environments and range from an OSID for repository interoperability to workflow and various learning service OSIDs. The concept of OSID is – like SQI – at a lower level of abstraction than the IMS DRI specification and is closely tied to ready-made and publicly available software components.

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) has been adopted beyond the digital library community and integrates with resolution standards like OpenURL (Dolog et al., 2003). OAI-PMH, in contrast to SQI or SRW/U, serves as facilitator for actual lookup services, also denoted as “Service Providers” in OAI terminology. A lookup standard that did not originate from the digital library or learning object community is OpenSearch (<http://opensearch.a9.com/>). Its main objective is the aggregation of lookup results of web-based search engines, therefore the focus lies – in contrast to SQI – on standardizing repository or rather search engine descriptions and result schemas for further processing. More recently, the issue of lookup has been discussed in connection with versatile web-based query languages independent from traditional data base management systems and their query exchanging and processing facilities: The Simple Protocol And RDF Query Language (SPARQL) specification hosts both a query language definition for RDF and a query exchange protocol (“SPARQL Protocol” in short) that is closely interwoven with the latter (DAWG, 2005b). This allows for specific query language constructs to be directly converted into required exchange patterns between a query source and a query target. The SPARQL protocol builds on concepts of (web) service-oriented architectures and targets a large set of use cases including the discovery of RDF-based descriptions of learning resources (DAWG, 2005a). The tight coupling of query language and query exchange / lookup protocol is a clear distinction between SPARQL protocol and SQI. The latter delivers arbitrary languages that are entirely independent from SQI as shuttle protocol.

SQI comprises two central atomic services as expanded in more detail in Section 3: a lookup or query shuttle service and a combined session, authentication and authorization management service. As for lookup, this allows for two basic modes of operation, stateless and stateful, and therefore synchronous or asynchronous exchanges of queries and corresponding result sets. In this respect, SRW/U shares similarities with SQI, but also shows major differences. SRW/U is purely synchronous (source-initiated), i.e. query results are returned within a single transaction. Query results, however, can be cached optionally and referred back to be based on a result set identifier up to a configurable time-to-live. While limited in its basic lookup service, SRW/U offers differently flavoured lookup refinements in addition, i.e. an index browsing (‘scan’) and advertising (‘explain’) service that exposes repository capabilities on demand (Board, 2004). In the same way, OAI-PMH takes in the query primitive ‘Identify’ that requests basic repository blueprints (Dolog et al., 2003). The OpenSearch standard, on the other hand, requires depositing repository descriptions with an aggregator’s registry. Nevertheless, all mentioned specifications including the EduSource ECL protocol are limited to synchronicity of their basic lookup services, apart from SQI.

The IMS DRI specification outlines five core commands in terms of pre-defined conversational styles, i.e. Search/Expose, Gather/Expose, Alert/Expose, Submit/Store, and Request/Deliver, on a highly abstract level. SQI therefore conforms to the Search/Expose conversation pattern while EduSource, that explicitly implements IMS DRI specification to a great extent in a service-oriented manner, provides a wider spectrum of lookup options, i.e. Search/Expose, Submit/Store, Request/Deliver and Gather/Expose. The SPARQL Protocol itself does not distinguish different types of lookup as they are imposed by the hosted RDF Query Language. The latter comprises four distinctive ‘query forms’ for requesting only variable bound responses (‘select’), for generating custom return RDF graphs (‘construct’), for requesting context information about resources (‘describe’) and for limiting results to binary truth statements (‘ask’) (DAWG, 2004).

Most related interoperability standards such as SRW/U, OpenSearch, OAI-PMH or the SPARQL Protocol do not explicitly include a formally conceptualized session or access management service, though they can be complemented by opting for an application profile or by adopting an appropriate specification (CORDRA, 2004). Apart from SQI, the OKI project offers closely coupled Authentication and Authorization OSIDs that form a generic access framework (OKI, 2005.). Moreover, there are some OSIDs defining basic services such as lookup routing.

As far as composite services are concerned, EduSource provides the Splash Federated Search facility over a registry of ECL-compliant repositories (EduSource, 2003; Hatala et al., 2004). SQI, however, has proven its applicability for derived or enhanced services such as network for federated search, end-user software and LMS

plug-in facilities, and higher level location and storage resolution (see Section 4). OpenSearch defines an aggregation service for result sets conforming to RSS as existing syndication standard and therefore raises the issue of post-processing of lookup results as a composite service. More tailored composite services are devised by OKI by providing OSIDs for a variety of learning administration scenarios, such as assessment, course management, grading and the like (OKI, 2005).

The various approaches subject to this comparison also differ with respect to their specificity or neutrality towards underlying representation models and metadata schemas, i.e. prescribed specifications and/or application profiles. The degree of dependency varies widely for the levels of application services, messaging and transport protocols. SRW/U is entirely bound to XML as representation model at all of these levels: The SRW/U protocol is represented in XML, its SOAP flavor (SRW) in particular. SRW/U does not prescribe a specific metadata specification and therefore operates over various XML bindings of metadata standards such as Dublin Core, MARC and LOM. Furthermore, SRW/U was designed in mutual dependence for a specific query language, i.e. the Common Query Language (CQL) that provides both expressive power and readability to human seekers. As for shaping query results, SRW/U offers support for user selection of the schema used. In OAI-PMH, XML is also used at the messaging layer yet exclusively as representation form of responses to OAI-PMH requests. Similarly, OAI-PMH allows to use arbitrary XML bindings of various metadata standard with Dublin Core, MARC and RFC1807 being provided off-the-shelf. In contrast, SQI is only partly dependent on XML as representation model *by design*, provided that SQI is realized as SOAP-based web services. SQI equally supports the selection of result schema although the generic character of SQI would not require it per se: Some query languages offer constructs to achieve schema specificity of result sets, e.g. QEL or SPARQL. The IMS DRI specification contains explicit recommendations for the usage of XML (and therefore XQuery) at the mediating representation model and at the messaging layer, i.e. SOAP. As a consequence, the ECL protocol relies exclusively on XQuery and enforces compatibility of affiliated repositories. The latter supports various metadata schemas, including XML bindings of LOM, Cancore and Dublin Core. The SPARQL language and protocol are closely tight to RDF as representation model with XML solely being used for some query forms (apart from RDF/XML as notational form). At the messaging layer, XML is integral part of some proposed protocol bindings such as SOAP. OpenSearch adopts XML in the shape of RSS as response format.

The interoperability standards can be realized differently at the messaging and transport layer. We distinguish between different architectural styles that underlie interaction scenarios, i.e. Remote Procedure Call (RPC), Representational State Transfer (REST) and Document / Message Exchange (D/ME), and embodiments of these styles in terms of protocols (XML-RPC, SOAP, HTTP). SQI and OKI OSIDs reflect the RPC style as a large proportion of the comparable work does, e.g. OAI-PMH, SRW/U, and SOAP is predominantly used as binding protocol. In contrast to SQI and OKI, OAI-PMH, SRW/U (i.e. its SRU flavour) offer alternatives conforming to the REST that are therefore realized as HTTP transactions. OpenSearch, however, builds on REST exclusively.

Implementation Case Studies

Since the first stable version of the specification was made available in March 2004 many learning repositories have taken advantage of SQI to connect them to the outside world. Under the auspices of the CEN/ISSS Learning Technologies Workshop the following projects took advantage of the SQI specification.

Ariadne

The basic mission of Ariadne is to enable better quality learning through the development of learning objects, tools and methodologies that enable a "share and reuse" approach for education and training. The Ariadne Learning Object Repository is called the "Knowledge Pool System" (KPS): the basic idea is that the KPS holds learning objects and metadata that describe them, so that they can be more easily managed and made available in appropriate contexts. Typical Ariadne user communities include higher education as well as corporate training.

The core of the Ariadne Knowledge Pool System (KPS), (Duval et al., 2001) is a distributed network of Learning Object Repositories that replicate both (the publicly available subset of) content and metadata. On top of this core infrastructure, Ariadne provides its members with a set of tools that are loosely coupled with the KPS (Ternier & Duval, 2003). Through these tools, the user community can transparently manage learning objects.

Examples of such applications that were coupled with the Ariadne KPS are Learning Management Systems (like Blackboard, Moodle and Ines) or authoring tools (like Phoenix or Powerpoint). Integrating SQI into these applications provides learners and teachers with transparent access to Learning Objects. Using e.g. the PowerPoint plug-in, a teacher can easily reuse pictures, slides that are already stored in the Ariadne knowledge pool, while the LMS plug-in enables reuse of complete learning objects like courses or presentations.

More generally, a distinction is made between two search scenarios that pose different requirements to a search service: searching within the Ariadne KPS only and searching beyond the borders of Ariadne.

While implementing an SQI target on the Ariadne KPS, we experienced that a synchronous target lowers the threshold for integrating search functionality into an application such as Microsoft PowerPoint. In this scenario, an application sends queries to a synchronous target and only fetches additional results when they are requested by the user. As a single Ariadne repository acts in this context as if it were a centralized system and as searching these repositories is optimized with search indices, there is no need to cache results for a query. This motivated the use of a stateless scenario. In a first phase, an SQI target was constructed that implements all SQI methods. Next, persistent sessions were introduced. As some applications only need a subset of the capabilities of a repository, we created persistent sessions that store a profile on the server. As an example, the PowerPoint plug-in uses a session identifier that is pre-configured to only search within graphical objects using Lucene queries. The Ariadne SQI target comes with three query languages. A simple query language enables a source to send a list of search terms that are then used by the target to conduct a full metadata search. Next, a proprietary query format provides Ariadne tools with high level advanced queries. Finally, the Ariadne target also supports the Lucene query language.

In order to provide Ariadne members access to other repositories, a federated search engine (Ternier et al., 2005) has been developed. This federated search engine acts as an intermediary service and makes it possible to transparently search into different repositories. The service exposes itself as a complete SQI target and acts as virtual repository that can be queried by a source. This facilitated the integration of the federated search engine into Ariadne clients as the interface of the federated search engine is the same as the Ariadne SQI Target. In the back-end, the federated search engine asynchronously distributes the query to different SQI enabled repositories. In this scenario, all repositories register their results using the results listener that is implemented on the federated search engine, which makes it easy for this engine to manage results that come from a large set of repositories. When a source invokes a query on this target, a state is being maintained for 15 minutes on the server. During this time, repositories can keep storing results for a query and the client can poll for additional results. In this stateful implementation, invoking the `getTotalResultsCount` method will result in different results as over time the amount of results that get registered will increase. Currently, searches are distributed into the following repositories: Ariadne, EdNA Online, EducaNext, Merlot, Pond, RDN, Smete, Voced, Citeseer and LionShare. As all these repositories support different query languages, which do not always easily map into one another, we started with an approach where a least common denominator of all query languages was implemented. In this approach, the query only consists of search terms which are translated by each repository into a query it can process.

Currently, Ariadne requires each partner to return a minimum of metadata fields, encoded as LOM XML: a URL, an identifier, a title and an identifier of the originating repository. The URL should resolve to the learning object. If access to the learning object is prohibited, the URL resolves to contact information. A repository identifier is necessary to give credits in a proper way to repository that yielded the results. Apart from the data elements mentioned above, all other LOM metadata fields are optional in the results.

Celebrate and iClass

The iClass project aims at exploiting the potential of ICT to support a personalized, flexible and learner-centric approach. This pedagogical approach strives to facilitate empowerment of both pupils and school teachers, while producing personalized learning experiences. Based on this, the iClass project seeks to establish a framework to deliver a personalized, adaptable, and adaptive learning experiences in a collaborative environment for learners. The framework is based on an advanced learning system, named 'Intelligent distributed Cognitive-based Learning System for Schools' (iClass). The latter takes advantage of an ontology-based architecture to sequence knowledge and adapt it to the learner's level of understanding and learning style by dynamically creating individualized learning objects.

The cost necessary to acquire, deploy, and integrate an e-learning infrastructure such as iClass makes it an operation that cannot be renewed, especially by schools, each time that new tools are available. Moreover the cost of acquiring and/or producing learning contents adapted to such an infrastructure is often more important than the cost of the infrastructure itself. In this context, interoperability between learning systems, by allowing legacy systems to get access to content developed in new systems (and conversely), makes it easier for schools to accommodate to the technological evolution.

The iClass adapter is a component of the iClass system. It enables the end-users of ‘non-iClass’ systems, such as the learning management systems and learning content management systems that are members of the Celebrate federation (van Assche & Massart, 2004), to search and get access to iClass contents (i.e., metadata and learning objects).

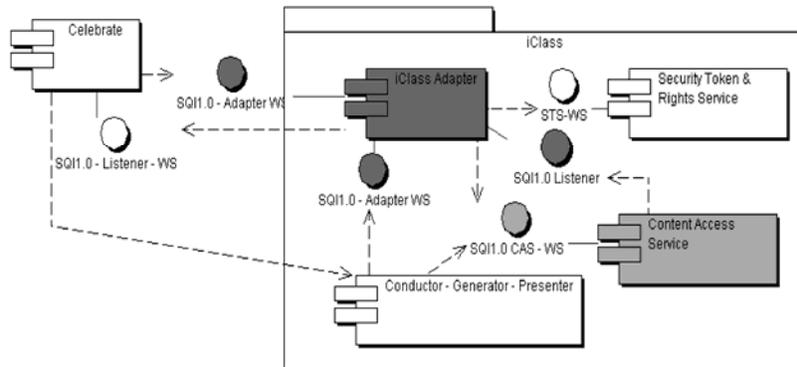


Figure 3. iClass and its adapter.

The diagram of Figure 3 depicts the main components of iClass that interacts with the adapter. The adapter is the entry point to the iClass system for external systems (e.g., system members of Celebrate). The adapter is also connected to:

- A content access service (CAS), which manages the access to metadata,
- A conductor service, which is in charge of coordinating the different tasks necessary to compose learning objects that are adapted to the user that requested them, and
- A rights management service, which in charge of authenticating and authorizing users and requests.

Metadata are stored in a peer-to-peer network of metadata repositories accessible thanks to the CAS. Usually, obtaining a learning object is a two-step process:

1. Searching and evaluating metadata: Selecting a learning object that satisfies user needs on the basis of the description provided in the metadata;
2. Consuming the learning object: Getting the selected learning object at the location (usually a URL) provided in the metadata.

The current version of the adapter only supports queries expressed in an ad hoc query language referred as ‘very simple query language’. This language consists of a set of keywords that are searched against the field title, description, and keywords of the learning object descriptions --For the second release of the adapter, it is planned to add support to XQUERY. The queries return results in a format based on the Celebrate application profile, an application profile of the IEEE Learning Object Metadata standard, which was especially developed to take into account the requirements of schools in Europe (Nirhamo & Van Assche, 2003). This metadata format provides an identifier of the described learning object rather than its location. Actually, the adaptive and multimedia nature of the iClass learning objects makes it difficult to get access to learning objects directly. This is why an extra step is required to resolve the location of a learning object identified in the metadata. This ‘resolve-location’ step is used to send a request to the conductor service that takes care of generating the requested object. The newly assembled learning object is then moved to a streaming server (called iClass presenter) and a URL from which, the object can be consumed is returned by the conductor. Since this process has potentially a certain duration, the conductor answers to these requests asynchronously in order to ensure adequate performance of the caller, in this case the iClass adapter. It is only when a learning object is available at the presenter that its location is known and can be returned.

Since there does not exist an open interface for performing this step asynchronously and rather than creating an ad hoc interface, it was decided to use SQI for this task as well by taking advantage of SQI independence in terms of query languages and results formats. This is achieved by adding a new 'query language' (for requesting a location) and a new 'results format' (for returning a location) to the list of languages and formats supported by the iClass adapter (Massart, 2005). The new query language is named 'ICLASS-LO-ID'. A query in this ad hoc language consists of the requested learning object's identifier as found in the metadata. The results format is named 'URL'. A result in this format consists of a URL pointing to the requested learning object.

This solution permits the minimization of the cost of implementing a 'resolve-learning-object-location' step for those learning object repositories that already use SQI for searching metadata. It is currently implemented as an extension of the SQI gateway of Celebrate.

Elena's Smart Space for Learning

The third case study is instantiated in the context of personnel development. In order to achieve interoperability among heterogeneous systems for corporate personnel development, the Elena project (Simon et al., 2004) realised a novel search and retrieval infrastructure for learning resources called Smart Space for Learning. This infrastructure allows for the integration of heterogeneous educational nodes in a semantic network and provides 'smart' access technology for it (Simon et al., 2003). Integrated into a process-support for learning goal definition, personalized search, and feedback tools, the educational semantic network (the 'space') plays a crucial role for supporting corporate personnel development.

A driving force for the design of the Elena Smart Space for Learning has been an extensive requirements analysis emphasising the need for integrating divergent resource providers into one common access tool (Gunnarsdottir et al., 2004). The broad variety of learning resources made available and accessible extends the scope of learning resources available. Hereby, potential learners are not dependent on offerings by a particular provider or are restricted to a specific learning format, e.g. a costly classroom-based course. On the contrary, they can expand their search to several types of learning formats and providers, e.g. books from Amazon media store.

Another driving force for setting up a smart space for learning is the provider's interest in disseminating their educational services and products via electronic distribution channels. The project team has evidenced a trend that these providers are increasingly offering electronic distribution channels in machine-processable formats such as XML while technology leaders even offer web services that allow for querying.

The infrastructure of Elena's Smart Space for Learning is built on three corner stones:

1. The Simple Query Interface (SQI) as interaction standard for both providing and consuming learning resources' metadata within SQI-compliant networks. Elena opted for a service-oriented architecture and therefore makes use of web service bindings (SOAP) for SQI.
2. A global metadata schema ('Elena schema') against which queries are issued and in which search results are expressed. This schema is realised as LOM application profile that enriches the LOM element battery by mixing in other metadata standards such as OpenQ, a metadata standard specifically designed for exchanging course information (Rivera et al., 2004). The rationale behind developing a LOM application profile is to design a schema according to Elena's use case requirements while preserving LOM compatibility. As for representation model, Elena employs RDF bindings for the global schema. Users' queries against this global schema are expressed in the Query Exchange Language (QEL, see Nilsson & Siberski, 2003), initially devised as a building block of Edutella, and semantic mappings are provided to convert from global to local schemas (Olmedilla & Palmèr, 2005).
3. Re-usable software components for integrating existing systems with minimum effort. They are applied for aligning the global and various local representation models (representation management and matching) and for mediating between query languages involved (query transformation).

So far, we have connected a number of providers to our network that can all be accessed from the EducaNext Portal for Exchanging Learning Resources (<http://www.educanext.org/networksearch/>) and from the personnel development portal Human Capital Development (HCD) Online (<http://www.hcd-online.org/networksearch/>). The recurrent pattern of linking a provider's system to a SQI-compliant network comprises five steps: In the first two steps, a SQI Target and a SQI Session Management object are to be created and registered with the underlying web service framework. Taking a white box perspective, the SQI target object is then to be populated with a mapping that is established between the respective local metadata representation and schema and the

global ones. Next, we devise a query translation facility that makes use of this mapping. Finally, the binding of the local results format to the global schema is realised, again in accordance with the previously defined mapping. The diversity that we experienced when applying this pattern in various concrete integration scenarios is considerable.

The quality of partnerships, ranging from highly protective and restrictive to more open ones – open refers to service hosting, disclosure of local metadata schemas and provided interfaces – required differing decisions at the architectural level: In restraining scenarios (e.g., Amazon media store and Seminarshop.com) gateway configurations appear as appropriate, in less restrictive cases (e.g., Moodle, ULI, etc.) intermediaries are not required, therefore reducing the complexity of implementation, maintenance and guaranteeing service quality. Characteristics of the local architectures affect architectural decisions similarly. A case in point is the integration of the Edutella peer-to-peer network that operates inherently in an asynchronous manner while SQI consumers might expect synchronous interaction (Olmedilla & Palmèr, 2005).

As for heterogeneity in terms of metadata representations and schemas, academic and commercial learning (content) management systems (e.g., EducaNext, CLIX) or course databases (course catalogue of the Vienna Executive Academy) predominantly rely on relational data base systems. Data base systems of different vendors, in our case Oracle, Postgresql, MySQL, and Firebird, cause, for instance, discrepancies in the encoding of result sets. Bridging and translating between representation models entail the risk of losses in terms of expressive power with respect to semantics for instance. Proprietary and/or non-standards compliant metadata schemas increase the complexity of mapping to the global schema (e.g., Seminarshop.com). We resolved these issues with respect to heterogeneity both by means of materialised or persistent integration and virtual or non-persistent integration. The former involves the periodic replication and transformation of a targeted metadata repository into a persistent and global-schema-compliant replica. Examples are the Executive Academy of the Vienna University of Economics whose relational repository is replicated into a single triple relation. Other examples include Knowledgebay and LASON whose replication is based on a native XML database. Virtual integration does not involve persistent storage of replicated, intermediate or transformed metadata but a recurring on-the-fly mapping process. This is the case for most integration cases in the current network setting (e.g., Amazon media store, BFI). In rare cases metadata repositories are already aligned to a certain extent both with respect to the representation model and the schema used (ULI).

In either case, whether considering materialised or virtual integration, query translation is an issue. The case of Amazon media stores demonstrates how to translate between a formal and highly powerful query language such as QEL and a parameterized canned query format as the Amazon PowerSearch syntax that cannot be considered a fully featured query language. Re-formulating queries expressed in RDF query languages such as QEL and RDQL into SQL is facilitated by re-usable software components provided by Edutella or similar efforts, such as D2RQ (Bizer & Seaborne, 2004). We developed and matured a comparable component for translating between QEL and XQuery that turned out to be highly re-usable in various, virtual and materialised integration contexts (Miklós & Sobernig, 2005).

Discussion and Concluding Remarks

In order to achieve interoperability among learning repositories, implementers require a common communication framework for querying. This paper discussed a set of methods referred to as SQI as a universal interoperability layer for educational networks. Although the effectiveness of the specification has been proven by several implementations (see previous section), lessons drawn from these and the discussion of comparable research efforts (see Section 3.8) suggest that some issues need to be further investigated. They equally relate to usage scenarios and application services provided.

Scope for application: We consider the adoption of SQI in different contexts, such as exchanging (language versions of) vocabularies, evaluation data about learning / training service providers and as building block for an interaction standard for collaboration services in various learning scenarios.

Advertise or Explain primitive: Specifications such as SRW/U or OAI-PMH provide such primitives while OpenSearch adopted some sort of capability schema for describing affiliated repositories (Board, 2004; Dolog et al., 2003; OpenSearch, 2005). SQI currently lacks such functionality and should therefore be extended accordingly, for instance, by adding an interface method that returns a SQI profile record that holds information on the query languages and results formats supported. This might alternatively be distributed among a set of

interrelated interface methods such as `getSupportedQueryMode`, `getSupportedQueryLanguages`, `getSupportedResultsFormats`.

State Inference / Modifier primitives: We evaluate the possibility of supporting search status management, for example, for cancellation of search, or query status reporting. This would allow a user at a query source to infer from and interfere with an active search process at the target.

Coupling result processing: OpenSearch primarily focuses on providing syndication of search results retrieved (OpenSearch, 2005). In the context of SQI it is important to find means for leveraging ranking mechanisms to the configuration of the actual lookup. This allows for reducing network traffic, since metadata that is anticipated not to be of high interest to the query issuing user can be omitted. At the same time, the quality of results subject to a process of pre-ranking is improved and the need for post-retrieval processing is considerably reduced.

Consolidated interaction model: In its current shape, the SQI specification involves some transaction overhead with respect to passing necessary arguments to the lookup procedure calls. In order to be able to set fundamental sets of arguments at once, a unifying method (`setQueryParameters`) could be introduced. Moreover, the conceptualization of a session management service at the level of SQI might be questioned: Currently, the HTTP specification and web service standards built on top of it do not explicitly provide session handling. Non-standardized practices based on cookie transactions, session-specific HTTP parameters and extensions to the SOAP Header element are proposed and realized. From this angle, a basic session management facility as described by SQI can be considered justified.

Acknowledgements

The SQI specification has been developed and financially supported under the auspices of the CEN/ISSS Workshop on Learning Technologies. This work is supported by European Commission via the IST projects CELEBRATE (<http://celebrate.eun.org/>), Elena (<http://www.elena-project.org/>), ICLASS (<http://www.iclass.info/>) and PROLEARN (<http://www.prolearn-project.org/>). We acknowledge contributions and comments to the SQI specification from Christian Werner (L3S Research Center), Dan Rehak (Carnegie Mellon University), Griff Richards (Simon Fraser University), Gerhard Müller (IMC), Julien Tane (Universität Karlsruhe), Marek Hatala (Simon Fraser University), Matthew J. Dovey (Oxford University), Michel Arnaud (Université de Paris X Nanterre), Nikos Papazis (NCSR), Peter Dolog (L3S Research Center), Sascha Markus (IMC), Stefan Brantner (BearingPoint Infonova), Stefano Ceri (Politecnico Milano), Simos Retalis (University of Piraeus), and Teo van Veen (Koninklijke Bibliotheek), Zoltán Miklós (Wirtschaftsuniversität Wien, Technische Universität Wien).

References

Bizer, C., & Seaborne, A. (2004). D2RQ - Treating Non-RDF Databases as Virtual RDF Graphs. In *Short Paper/Poster Proceedings of ISWC2004*, retrieved March 23, 2006 from <http://www.wiwiss.fu-berlin.de/suhl/bizer/d2rq/>.

Board, S. E. (2004). *SRW: Search/Retrieve Webservice Version 1.1 (Consolidated Specifications)*: retrieved March 23, 2006 from <http://srw.cheshire3.org/SRW-1.1.pdf>.

CEN/ISSS (2005). *A Simple Query Interface Specification for Learning Repositories (CEN Workshop Agreement #15454)*. Brussels, Belgium, retrieved 23 March 23, 2006 from <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/WS-LT/cwa15454-00-2005-Nov.pdf>.

CORDRA (2004). *CORDRA: Technical Introduction and Overview*: retrieved March 23, 2006 from <http://www.lsal.cmu.edu/lsal/expertise/projects/cordra/intro/intro-v1p00.html>.

DAWG (2004). *SPARQL Query Language for RDF*, retrieved March 23, 2006 from <http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/>

DAWG (2005a). *RDF Data Access Use Cases and Requirements*: W3C.

DAWG (2005b). *SPARQL Protocol for RDF*: W3C.

- Dolog, P., Gavriiloaie, R., Nejd, W., & Brase, J. (2003). Integrating Adaptive Hypermedia Techniques and Open RDF-based Environments. In *Proceedings of 12th International World Wide Web Conference*. Budapest, Hungary: ACM.
- Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., Wentland Forte, M., Ebel, N., Macowicz, M., Warkentyne, K., & Haenni, F. (2001). The Ariadne Knowledge Pool System. *Communications of the ACM*, 44 (5), 72-78.
- EduSource (2003). *ECL Protocol (Version 0.3)*. retrieved March 23, 2006 from http://www.edusplash.net/technical/ecl/ECL_Protocol_0.3.03.pdf
- Gunnarsdottir, S., Kieslinger, B., K uchler, T., & Simon, B. (2004). From e-Learning to Learning Management: Results of an International Survey. In *Proceedings of 4th International Conference on Knowledge Management*. Graz, Austria.
- Hatala, M., Richards, G., Eap, T., & Willms, J. (2004). The Interoperability of Learning Object Repositories and Services: Standards, Implementations and Lessons Learned. In *Proceedings of the 13th World Wide Web Conference*. New York City, USA.
- Henney, K. (2000, December). A tale of two patterns. *Java Report*, 84-88.
- IEEE (1990). *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York, NY: IEEE.
- IMS (2003). *IMS Digital Repositories Interoperability - Core Functions Information Model*: retrieved March 23, 2006 from http://www.imsglobal.org/digitalrepositories/driv1p0/imsdri_infov1p0.html.
- Massart, D. (2005). Accessing Learning Contents Using a Simple Query Interface Adapter. In *Proceedings of the ProLearn-iClass Thematic Workshop on Learning Objects in Context*. Leuven, Belgium.
- Mikl os, Z., & Sobernig, S. (2005). Query Translation between RDF and XML - A Case Study in the Educational Domain. In Olmedilla, D., Saito, N. & Simon, B. (Eds.), *WWW Workshop on Interoperability of Web-Based Educational Systems*, Chiba, Japan: CEUR, 27-35.
- Najjar, J., Ternier, S., & Duval, E. (2004). Interoperability of Learning Object Repositories: Complications and Guidelines. *IADIS International Journal of WWW/Internet*, 2, 1-16.
- Nilsson, M., & Siberski, W. (2003). *RDF Query Exchange Language (QEL) - Concepts, Semantics and RDF Syntax*: retrieved March 23, 2006 from <http://edutella.jxta.org/spec/qel.html>.
- Nirhamo, L., & Van Assche, F. (2003). *The CELEBRATE Metadata Application Profile. Celebrate Deliverable D 3.2 (IST-2001-35188)*: Available March 23, 2006 at http://celebrate.eun.org/docs/D3_2.pdf.
- NISO (2002). *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification*: NISO Press, retrieved March 23, 2006 from <http://www.niso.org/standards/resources/Z39-50-200x.pdf>.
- OKI (2005). *Open Knowledge Initiative*: <http://www.okiproject.org/>.
- Olmedilla, D., & Palm er, M. (2005). Interoperability for Peer-to-Peer Networks: Opening P2P to the rest of the World. In Olmedilla, D., Saito, N. & Simon, B. (Eds.), *Workshop on Interoperability of Web-Based Educational Systems*. Chiba, Japan.
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The PageRank Citation Ranking: Bringing Order to the Web. Stanford Digital Library Technologies Project*.
- Quemada, J., & Simon, B. (2003). A Use-Case Based Model for Learning Resources in Educational Mediators. *Educational Technology & Society*, 6 (4), 149-163.
- Raghavan, S., & Garcia-Molina, H. (2001). Crawling the Hidden Web. In *Proceedings of the 27th International Conference on Very Large Databases*.

- Rehak, D. R., Dodds, P., & Lannom, L. (2005). A model and Infrastructure for Federated Learning Content Repositories. *Paper presented at the WWW Workshop on Interoperability of Web-based Educational Systems*, Chiba, Japan, retrieved May 1, 2006 from <http://cordra.net/information/publications/2005/www2005/cordrawww2005.pdf>.
- Rivera, G., Simon, B., Quemada, J., & Salvachua, J. (2004). Improving LOM-Based Interoperability of Learning Repositories. In Duval, E. & Spyns, P. (Eds.), *Proceedings of the OTM 2004 Workshop on Ontologies, Semantics and E-learning, October, 2004*: Springer LNCS (3292).
- Simon, B., Dolog, P., Miklós, Z., Olmedilla, D., & Sintek, M. (2004). Conceptualizing Smart Spaces for Learning. *Journal of Interactive Media in Education, 2004* (9).
- Simon, B., Retalis, S., & Brantner, S. (2003). Building Interoperability among Learning Content Management Systems. In *Proceedings of the 12th World Wide Web Conference*. Budapest.
- Ternier, S., & Duval, E. (2003). Web services for the ARIADNE Knowledge Pool System. In *Proceedings of the 3rd Annual Ariadne Conference*. Leuven, Belgium.
- Ternier, S., Olmedilla, D., & Duval, E. (2005). Peer-to-Peer versus Federated Search: Towards more Interoperable Learning Object Repositories. In *Proceedings of ED-MEDIA 2005*.
- van Assche, F., & Massart, D. (2004). Federation and brokerage of learning objects and their metadata. In Kinshuk, Looi, C. K., Sultinen, E., Sampson, D., Aedo, I., Uden, L. & Kähkönen, E. (Eds.), *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies*, Los Alamitos, CA: IEEE Computer Society, 316-320.
- Van de Sompel, H., & Beit-Arie, O. (2001). Open Linking in the Scholarly Information Environment Using the OpenURL Framework. *D-Lib Magazine, 7* (3), retrieved May 1, 2006 from <http://www.dlib.org/dlib/march01/vandesompel/03vandesompel.html>.
- Wiederhold, G., Wegner, P. & Ceri, S. (1992). Toward Megaprogramming. *Communications of the ACM, 35* (11), 89-99.