

D-Lib Magazine

September/October 2010

Volume 16, Number 9/10

The Simple Publishing Interface (SPI)

Stefaan Ternier, Open Universiteit, The Netherlands

David Massart, European Schoolnet (EUN), Belgium

Michael Totschnig, University of Economics and Business (WU Wien), Austria

Joris Klerkx, Katholieke Universiteit Leuven (K.U.Leuven), Belgium

Erik Duval, Katholieke Universiteit Leuven (K.U.Leuven), Belgium

Point of contact for this article: stefaan.ternier@ou.nl

doi:10.1045/september2010-ternier

Abstract

The Simple Publishing Interface (SPI) is a new publishing protocol, developed under the auspices of the European Committee for Standardization (CEN) workshop on learning technologies. This protocol aims to facilitate the communication between content producing tools and repositories that persistently manage learning resources and metadata. The SPI work focuses on two problems: (1) facilitating the metadata and resource publication process (publication in this context refers to the ability to ingest metadata and resources); and (2) enabling interoperability between various components in a federation of repositories. This article discusses the different contexts where a protocol for publishing resources is relevant. SPI contains an abstract domain model and presents several methods that a repository can support. An Atom Publishing Protocol binding is proposed that allows for implementing SPI with a concrete technology and enables interoperability between applications.

1 Introduction

Making new materials available in a repository was often too cumbersome. A new resource had to be manually uploaded to the repository and described by metadata before it could be published. In many learning object repositories, the number of mandatory metadata fields to be provided was often too large, leaving the user frustrated. Integrating the publication process into applications where learning resources are consumed (e.g. Learning Management Systems (LMS)) or produced (e.g. authoring tools) facilitates the metadata creation process.^[2] Authoring tools have a context in which this metadata is available. For instance, a document authoring tool can suggest metadata like title, author or contributor. Learning Management Systems have a context in which a resource is used. For instance, using course metadata, information about the intended end user can be derived. The existence of many Learning Management Systems and authoring tools for learning, building on standards (IMS Learning Design (IMS-LD)) ^[4] or the Shareable Content Object Reference Model (SCORM) ^[5], and the existence of many different kinds of learning object repositories, makes the standardization of publication services important for interoperability.

The eContent^{plus} programme of the European Commission has led to various federations of repositories that focus on enriching existing digital materials. Various eContent^{plus} projects, including ASPECT (Adopting Standards and Specifications for Educational Content) ^[6], MELT (Metadata Ecology for Learning and Teaching) ^[7], MACE (Metadata for Architectural Contents in Europe) ^[8], ICOPER (Interoperable Content for Performance in a Competency-driven Society) ^[9] and Share.TEC (Sharing Digital Resources in the Teaching Education Community) ^[10], resulted in federations of repositories that were used as test beds for SPI. With these repository architectures, interoperability between various components is important. This includes the following components:

- Repositories are responsible for metadata and learning object management.

- Harvested metadata stores make metadata records (harvested from various repositories) searchable
- Authoring tools create new learning objects.

Here SPI enables one component to push metadata or resources to another component. Section 2.3 (Metadata Harvesting) and Section 5 (Implementations and Lessons Learned) will illustrate the role played by SPI and how it complements metadata harvesting protocols.

This paper is structured as follows: Section 2 proposes three real-life examples of use of the Simple Publishing Interface (SPI). Next, the main functionalities of the interface are described in Section 3. As an abstract specification, SPI is typically bound to a specific technology in concrete implementations. Section 4 presents an Atom Publishing Protocol binding for SPI. The practical consequences of using SPI as a complement to the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH [19]) for acquiring metadata are discussed in Section 5. This discussion is followed by the more theoretical comparison between SPI and other protocols in Section 6. We conclude this article in Section 7.

2 Application domains

This section illustrates how SPI can be used in three different application domains.:

- Learning management systems that are traditionally used to publish resources to students via courses. By introducing SPI, the publication to a course and the publication for persistent management in a repository can be combined in one user interaction cycle.
- Authoring tools which can, through SPI, publish a resource to a repository where it is available for reuse.
- Learning object repository federations, which can build on SPI to realize an extensible architecture. For instance, the metadata store component (see Figure 1) could be interchanged with another component that supports SPI.

In the following, each of these application domains is illustrated with an example where SPI has actually been implemented and used.

2.1 The Moodle Bridge

Moodle (Modular Object-Oriented Dynamic Learning Environment) [11] is a heavily used open-source Learning Management System. At the time of writing, there are 53,987 currently verified active sites that have registered from 216 countries with over 37 million users. Teachers can use Moodle to give online courses and share learning materials with their students. Those materials are restricted to the Moodle instance that the teacher is using. At the time of writing, a teacher cannot publish materials for students and teachers, or reuse materials that are available in other Moodle instances or other Learning Object Repositories.

Therefore, a bridge has been developed in ARIADNE [12][13], a European foundation that aims to foster "Share and Reuse" of learning resources, that connects a Moodle instance with a Learning Object Repository [14]. This enables users to either publish in, or search for, relevant content from their usual environment (i.e Moodle) instead of having to go to a separate web site. It also offers users opportunities to use their Moodle context for enriching their search criteria.

To achieve maximum interoperability, we have used the SPI protocol for publishing and the Simple Query Interface (SQI) [31] for searching materials. The Moodle bridge therefore allows:

1. publishing material from Moodle to any LOR that supports SPI such as the ARIADNE Knowledge Pool System (KPS) [12], the Learning Resource Exchange (LRE) [22], the Open ICOPER Content Space (OICS) [9], etc.
2. searching and importing existing material stored in any repository that supports SQI [15] such as GLOBE (Global Learning Objects Brokered Exchange) [1] into the Moodle courseware.

2.2 OpenGLM and the OICS

The Open ICOPER Content Space (OICS) is a learning object repository that implements the ICOPER reference model [32]. It supports managing various kinds of outcome-oriented teaching resources such as learning outcomes, learning content, learning opportunities, teaching methods and learning designs. The OICS provides services to search, retrieve and recommend these resources. For publication, it relies on SPI.

In the ICOPER project different plugins and widgets have been implemented that build on the services offered by the OICS. Publication extensions were made for .LRN, Elgg and Moodle which allow users to publish additions to their OICS learning outcome profiles. By integrating SPI in their daily tool usage, users do not have to separately login into the OICS system,

which better suits their workflow.

The Open Graphic Learning Modeller (OpenGLM) [17] is a learning design authoring tool, developed at University of Vienna. OpenGLM provides the following functionality through the OICS repository:

- Retrieve learning content from OICS and include it in a learning design
- Retrieve learning outcome definitions from the OICS and add those to the learning design metadata
- Retrieve an existing learning design from OICS
- Publish a design (and its metadata) to the OICS repository
- Define a learning outcome and publish it to OICS

OpenGLM thus makes use of the whole range of SPI features that OICS provides: publication of learning objects, publication of metadata records and enrichment. By neatly integrating publication facilities into a desktop authoring tool it aims to be beneficial for the user's workflow.

2.3 Metadata harvester

Figure 1 generalizes the architecture that is currently in place in many learning repository networks. In this architecture, metadata is gathered from various participating repositories through OAI-PMH. Typically, within a network (for instance Share.TEC) partners contribute metadata for a domain, e.g. teacher education. Within Share.TEC, a LOM [18] application profile relevant for materials for teacher educators was created. All partners that offer metadata according to this application profile have set up an OAI-PMH [19] target to enable gathering their metadata. These partner repositories are represented by right-side repositories in Figure 1. The metadata harvester is a component in this architecture that periodically checks the partner repositories for new metadata and updates the harvested metadata store through a publication API (SPI). The harvested metadata store offers search access to various search tools. New partner repositories can be added to the network through the registry. The metadata harvester uses this registry to decide which repositories to harvest from. Note that a single registry can serve multiple federations.

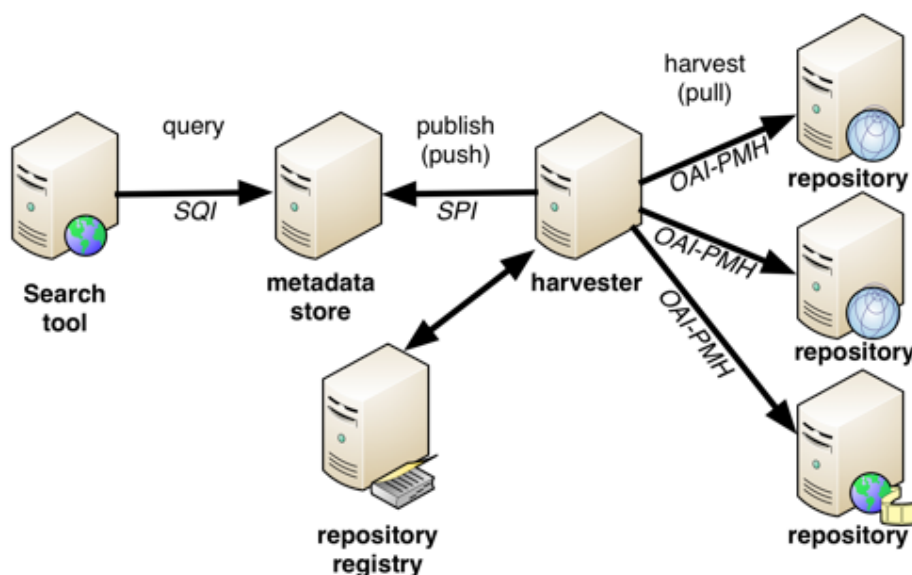


Figure 1: Federation of Repositories.

Although this architecture is similar to MACE, MELT, ASPECT, iCoper and other projects, the components that make up this architecture often are not. Having a standard in place for every connector promotes the interchangeability of components. A strict separation between the responsibilities of each component is important when setting up an extensible, open architecture for metadata management. The harvesting component, for instance, is responsible for the periodic harvesting and is unaware of metadata management (responsibility of local repositories) and is unaware of the search logic that is offered by the repository cache.

3 SPI Model

The Simple Publishing Interface (SPI) is used to push digital resources and/or their metadata into a repository. SPI makes

relatively few assumptions about the resources and metadata that can be published. These assumptions are illustrated by the UML class diagram in Figure 2.

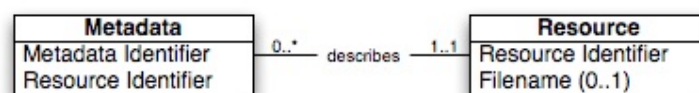


Figure 2 - Digital resources and their metadata

Every resource must have an identifier and may have an associated filename. A resource can be described by zero or more metadata instances. Every metadata instance describes exactly one resource. It must have a metadata identifier that identifies the metadata instance itself and must have the resource identifier of the resource it describes.

SPI does not assume that a resource and its metadata need to be published in the same repository. As a consequence, SPI supports four operations:

1. Submitting a resource to a repository
2. Deleting a resource from a repository
3. Publishing a metadata record to a repository
4. Deleting a metadata record from a repository

There is no explicit operation to update a resource or a metadata instance. An SPI binding can offer this functionality out-of-band as an SPI extension, or this functionality can be simulated by deleting the resource and re-publishing it.

Besides extending the specification, an SPI binding can also omit operations. Similarly, an implementation of a binding can (if the binding allows this) omit operations. Depending on the content managed by a repository (i.e., resources only, metadata only, or resources and metadata), a repository can support different combinations of these operations. For instance, a *referratory* – a repository that only manages metadata – will only offer operations to submit and delete metadata and omit the operations to manage resources.

This section provides an overview of the Simple Publishing Interface and the main principles it is based on. For a detailed description of the specification see the SPI specification [20].

3.1 Submitting a resource

Submitting a resource involves sending a binary stream to a target. Depending on the binding that is used, this byte-stream can be encoded in various ways. SPI proposes two ways of submitting a resource to a repository: "by-value" and "by-reference".

In by-value publishing, the resource is directly embedded (after encoding) in a message sent to a repository. In by-reference publishing, the message sent to the repository only contains a reference (e.g., a URL) to the submitted resource. It is then the responsibility of the repository to use this reference to retrieve the resource and store it.

By-value publishing is useful for a standalone application (e.g. an authoring tool), which is generally not associated with a web server from which a repository can obtain a resource. Embedding a resource in a message passed to the repository is beneficial for publishing a resource from a desktop application. It lowers the threshold for publication because uploading the resource to a third party component that hosts the referenced resource is unnecessary. By-reference publishing is particularly suited to publishing large resources, since embedding large files into a single message may cause degraded performance, resulting in a need for a distinct method (e.g., FTP, HTTP, SCP, etc.) for a large resource.

3.2 Submitting a metadata instance

The submission of a metadata instance to a repository is similar to the submission of a resource by-value. The metadata instance itself is directly embedded in a message sent to the repository. Since several metadata can co-exist for a single resource, the operation offers the possibility to specify the identifier of the metadata instance and the identifier of the resource it describes. Publishing a new metadata instance for a resource can therefore be realized by publishing it using a new metadata identifier.

3.3 Deleting resources and metadata

SPI supports two delete operations for resource and metadata. These operations are straightforward. The identifier of the object (resource or metadata instance) to delete is submitted to the repository that takes care of carrying out the deletion.

4 SPI binding to APP

The SPI is an abstract model that seeks consensus on which high level methods to offer for publishing metadata and resources. A binding to a technology makes these methods more concrete and defines how applications can interoperate practically. SPI has been bound to the Atom Publishing Protocol (APP or AtomPub) [16] for several reasons.

AtomPub is a simple HTTP-based protocol for publishing Atom entries to the web. Although Atom as a format for web feeds is well known, AtomPub can do much more than serve as a protocol for blog clients. Google's gdata API [33] illustrates this and builds on AtomPub, where Atom is used as a container to create, read, update or delete items in over 20 Google applications.

More in the realm of repositories, SWORD (Simple Web-service Offering Repository Deposit)[21] is a profile of AtomPub that defines a number of Atom elements and HTTP extensions to support the publication of resources or content packages. Both SWORD and SPI initiatives started around the same time. In a later stage of the SPI project, it was decided to offer support for AtomPub in order to enable interoperability with the SWORD specification.

SPI is in essence an abstract protocol that can be bound to existing technologies (APP, SOAP, XML-RPC), while SWORD is an extension of existing technology (APP). On a more practical level, the main difference between SWORD and SPI is in the submission of metadata. In SWORD, metadata is available in a package and thus submitted to the repository as a part of the resource. In contrast, the SPI model makes a clear distinction between metadata and content at submission time. Thus, in SPI, metadata is made available through the atom entry and not inside the resource.

The rationale for this strict separation comes from the SPI application domain. Where SWORD is primarily concerned with depositing data, SPI is also intended for application scenarios where only metadata is considered (see the Metadata Harvesting section). The SPI model supports 3 publishing application scenarios:

- (a) Referratory publish: Publish metadata only
- (b) Publish content only
- (c) Publish metadata linked to content

Scenarios (a) and (b) are easy to realize with AtomPub. Through an HTTP POST message, one can either transport an Atom entry (with educational metadata) or binary content (representing the resource) to a repository. For a metadata or resource submission, SPI makes it possible for the source to propose an identifier by using an X-Identifier HTTP header.

Scenario (c) can be realized in two ways: a source can publish a resource first and then update the metadata (c1), or a source can publish the metadata first followed by attaching a resource (c2). Scenario (c1) is a standard AtomPub scenario and is described in section 9.6 of RFC 5023 [16]. The second scenario (c2) is not explicitly mentioned in RFC 5023. Here, the binding defines that an atom entry including metadata is submitted first, followed by attaching a resource as binary content.

The SWORD specification assumes content packages are published. In this model, resource and metadata are aggregated and sent as one package to the repository. This corresponds to scenario (b).

5 Implementations/Lessons learned

The Learning Resource Exchange (LRE) [22] is a service that allows European teachers to get access to digital educational content from many different countries and providers. The LRE content providers produce metadata, i.e. machine-readable descriptions of the educational content they want to share with the LRE teachers. Each content provider exposes its metadata so that it can be easily accessed by the LRE. The LRE collects metadata from the different content providers and compiles them to produce a digital catalog of learning resources that can be consulted by LRE teachers.

Before the introduction of SPI, OAI-PMH was the main mechanism used by the LRE to collect metadata. OAI-PMH is a pull mechanism meaning that the LRE takes the initiative to ingest new metadata by calling the OAI-PMH target of the repository to harvest. However, some content providers refused to deploy OAI-PMH targets, because they regarded hosting a web service as a potential source of security problems. To accommodate these providers, SPI was proposed as a way to safely push metadata into the LRE by calling an LRE service without having to deploy and run any service on the providers' end.

Prior to its storage in the LRE central repository, each harvested metadata instance is checked for conformance with the LRE

metadata application profile [30]. It is also compared to existing metadata records in order to check if it refers to new learning content or to learning content for which a description already exists in the LRE. As a result of this process, each LRE metadata record is uniquely identified and kept up-to-date [29].

Because several of the LRE partner repositories requested to push metadata to the LRE, this metadata flow had to be revised. Therefore, SPI was introduced as a complementary way to ingest metadata. SPI is a push mechanism, meaning that the initiative of publishing new metadata records in the LRE is taken by the metadata source and not by the LRE. However, in order to work properly, an SPI target requires a different identifier system than the one used by the LRE to manage metadata records in its central repository.

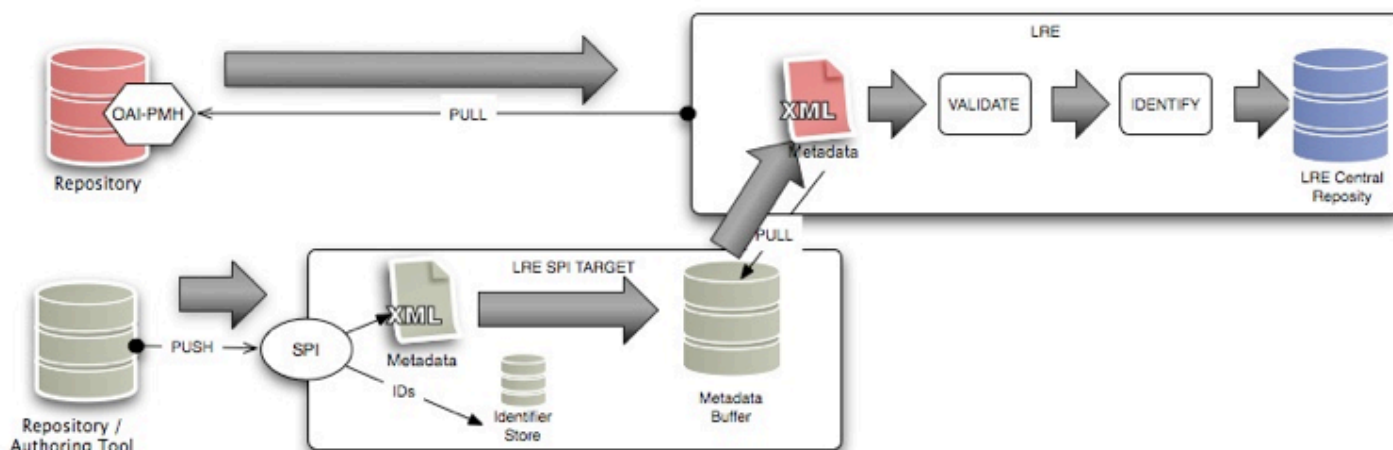


Figure 3 - Harvesting versus publishing metadata into the LRE

Figure 3 illustrates the differences in workflow between publishing and harvesting. The big gray arrows indicate the metadata flows, whereas the thin black arrows indicate the direction of the calls between systems and their components (from the caller to the call recipient). SPI allows content providers to manage the resources they publish using their own identifiers. The latter are not necessarily consistent with the globally unique identifiers used by the LRE. Therefore, when a source pushes metadata instances into the LRE using SPI, it is necessary to store these instances in a metadata buffer, to keep their identifiers as they were provided by the source, and to maintain the correspondence between these identifiers and the metadata instances they identify so that later they can be updated by the source. In turn, the metadata contained in the SPI metadata buffer can be "harvested" by the LRE and handled by the LRE as the metadata obtained using OAI-PMH.

6 Relationship to existing standards and specifications

By defining methods for the publication of learning resources and metadata records, SPI tries to complement existing specifications, which either do not support metadata in a generic way, or are not defined as a platform independent protocol.

OAI-PMH has played a long-standing role in the dissemination of metadata. While the official specification is tied to the Dublin Core [23] as a mandatory default metadata format, it supports inclusion of metadata of any format in an XML-encoded byte stream. SPI has been used as a complement to OAI-PMH. Figure 1 shows a harvester component that pushes the OAI-PMH metadata records to a metadata store through SPI, and a repository can offer read access through OAI-PMH, while offering write access through SPI.

Although both SPI and OAI-PMH can serve the same purpose: i.e. disseminating metadata records, they differ in some key aspects:

- OAI-PMH features a pulling mechanism, while SPI implements a push protocol.
- Although OAI-PMH is mostly used to disseminate metadata, approaches exist to harvest resources with this framework [34]. Here, both by-value and by-reference publishing are presented as two not mutually exclusive ways to pull resources from a repository. Similar to SPI, by-value refers to embedding a base64-encoding of the datastream inside the OAI-PMH XML response.

The JISC-funded SWORD project aimed to "improve the efficiency and quality of repository deposit and to diversify and

expedite the options for timely population of repositories with content whilst promoting a common deposit interface and supporting the Information Environment principles of interoperability" [35]. SWORD is defined as a profile of the ATOM publishing protocol. It deals with metadata only insofar it is part of a package that can be extracted by the repository. While in a recent position paper [25], the need for the client to bundle content and metadata is recognized as a limitation, the inclusion of metadata in the description of the item is still explicitly excluded as a solution. With SPI, we have taken the idea that an ATOM entry resource contains metadata about a media resource literally, and use the inclusion of XML encoded metadata into the ATOM entry as a simple mechanism for allowing clients to provide a rich set of metadata about resources beyond the fields defined by the ATOM standard.

The Content Management Interoperability Services (CMIS) [28] specification primarily strives for interoperability between enterprise content management systems (CMS) and desktop authoring tools. Similar to SPI, CMIS defines an abstract model that can be bound to different technologies (including APP). Instead of the relation between resources and metadata, it is concerned with a domain model for CMSs and operations to interoperate with this model. This CMIS domain model focuses on a hierarchical structure of content, access control, versioning, folder structures, etc., and is thus more fine-grained than SPI.

7 Conclusions

This article presents SPI as a protocol for publishing learning objects and their metadata to heterogeneous repositories. This work is in line with previous achievements in defining metadata standards, query protocols [15] and a query language [24]. In this article we have demonstrated how SPI enables different publication scenarios:

- By-value publishing enables desktop clients (such as OpenGLM) to publish resources to a repository.
- By-reference publishing is more useful for web based tools (e.g. LMS's). This mode is however not available in the current AtomPub binding of SPI.
- SPI has been validated in various eContentplus projects where it facilitated interoperability between loosely coupled components in a federation of repositories. This large test-bed demonstrates the practical usability of the protocol.

This work builds on existing standards and technologies and we believe that this is an important step forward for the technology enhanced learning community, as it contributes to the existing set of building blocks for federations of learning object repositories.

8 Acknowledgements

This work has been supported in part by the European Committee for Standardization (CEN) [3] under the CEN/Expert/2009/36 contract. The authors would like to acknowledge contributions from the following projects co-funded by the European Commission's eContentplus programme: Metadata for Architectural Contents in Europe (MACE), Metadata Ecology for Learning and Teaching (MELT), Adopting Standards and Specifications for Educational Content (ASPECT), Interoperable Content for Performance in a Competency-driven Society (ICOPER) and Sharing Digital Resources in the Teaching Education Community (Share.TEC).

9 References

- [1] Global Learning Objects Brokered Exchange (GLOBE): <http://www.globe-info.org/>.
- [2] Erik Duval & Wayne Hodgins, "A LOM Research Agenda", WWW2003 - Twelfth International World Wide Web Conference, 20-24 May 2003, Budapest, Hungary.
- [3] CEN Workshop on 'Learning Technologies' (WS/LT): <http://www.cen.eu/CEN/sectors/sectors/iss/activity/Pages/wslt.aspx>.
- [4] IMS Learning Design: <http://www.imsglobal.org/learningdesign/index.html>, last retrieved: August 2010.
- [5] SCORM "Shareable Content Object Reference Model": <http://www.adlnet.gov/Technologies/scorm/default.aspx>, last retrieved: August 2010.
- [6] ASPECT "Adopting Standards and Specifications for Educational Content": <http://www.aspect-project.org/>, last retrieved: August 2010.
- [7] MELT "Metadata Ecology for Learning and Teaching": <http://info.melt-project.eu/>, last retrieved: August 2010.

- [8] MACE "Metadata for Architectural Contents in Europe": <http://portal.mace-project.eu/>, last retrieved: August 2010 .
- [9] ICOPER "Interoperable Content for Performance in a competency-driven Society": <http://www.icoper.org/>, last retrieved: August 2010.
- [10] Share.TEC "Sharing Digital Resources in the Teaching Education Community": <http://www.share-tec.eu/>, last retrieved: August 2010.
- [11] MOODLE: <http://moodle.org/>, last retrieved: August 2010.
- [12] S. Ternier, K. Verbert, G. Parra, B. Vandeputte, K. Klerkx, E. Duval, V. Ordonez, and X. Ochoa, *The ARIADNE infrastructure for Managing and Storing Metadata*, IEEE Internet Computing, 13(4):pp. 18-25, July/August 2009.
- [13] J. Klerkx, Bram Vandeputte, Gonzalo Parra, J.L. Santos, F. Van Assche and Erik Duval, *How to Share and Reuse Learning Resources: the ARIADNE Experience*. Accepted for publication in Proceedings of EC-TEL, Barcelona, September 2010.
- [14] J. Broisin, P. Vidal, M. Meire, and E. Duval, *Bridging the Gap between Learning Management Systems and Learning Object Repositories: Exploiting Learning Context Information*. In ELETE 2005, IEEE Computer Society, p. 478-483, Lisbon, July 2005.
- [15] B. Simon, D. Massart, F. Van Assche, S. Ternier, E. Duval, *A simple query interface specification for learning repositories*. CEN Workshop Agreement (CWA 15454). <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/WS-LT/cwa15454-00-2005-Nov.pdf>.
- [16] AtomPub (RFC 5023) "Atom Publishing Protocol": <http://bitworking.org/projects/atom/rfc5023.html>.
- [17] S. Neumann, P. Oberhuemer, *Supporting instructors in creating standard conformant learning designs: the Graphical Learning Modeler*. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008, Vienna, Austria, pp. 3510-3519. AACE (2008).
- [18] IEEE LOM "IEEE Standard for Learning Object Metadata": <http://ltsc.ieee.org/wg12/>.
- [19] C. Lagoze, H. Van de Sompel, M. Nelson, S. Warner, *The Open Archives Initiative Protocol for Metadata Harvesting - Version 2.0*, <http://www.openarchives.org/OAI/openarchivesprotocol.html>.
- [20] *The Simple Publishing Interface (SPI) Specification*. CEN Workshop Agreement (CWA 16087), <ftp://ftp.cen.eu/CEN/Sectors/TCandWorkshops/Workshops/CWA16097.pdf>
- [21] J. Allinson, S. François and S. Lewis, *SWORD: Simple Web-service Offering Repository Deposit*, <http://www.ariadne.ac.uk/issue54/allinson-et-al/>.
- [22] D. Massart, The EUN Learning Resource Exchange (LRE). *The 15th International Conference on Computers in Education (ICCE2007) Supplementary Proceedings*, volume 1, pages 170-174, November 2007.
- [23] The Dublin Core Metadata Initiative: <http://dublincore.org/>.
- [24] S. Ternier, D. Massart, A. Campi, S. Guinea, S. Ceri, E. Duval, *Interoperability for Searching Learning Object Repositories: The ProLearn Query Language*, *D-Lib Magazine*, January/February 2008, Volume 14 Number 1/2, [doi:10.1045/january2008-ceri](http://dx.doi.org/10.1045/january2008-ceri).
- [25] R. Jones, Symplectic Ltd. *SWORD v2.0: Deposit Lifecycle*, <http://sword2depositlifecycle.jiscpress.org/files/2010/07/SWORDv2.pdf>, July 2010.
- [26] JSR 283 "Content Repository for Java™ Technology API Version 2.0": <http://jcp.org/en/jsr/detail?id=283>, September 2009.
- [27] R. T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, PhD thesis, 2000.
- [28] R. Cover, *Vendors Publish Content Management Interoperability Services (CMIS) Standard*, The Cover Pages, <http://xml.coverpages.org/ni2008-09-10-a.html>, September 2008.
- [29] D. Massart, "Towards a pan-European learning resource exchange infrastructure". In Y. Feldman, D. Kraft, and T. Kuflik, editors, *Proceedings of the 7th conference on Next Generation Information Technologies and Systems (NGITS'2009)*, LNCS

5831, pages 121-132, Haifa, Israel, June 2009, Springer.

[30] D. Massart, E. Shulman, and F. Van Assche, *Learning Resource Exchange Metadata Application Profile version 4.5*. European Schoolnet, 2010, <http://lre.eun.org/node/6>.

[31] B. Simon, D. Massart, F. Van Assche, S. Ternier, E. Duval, "A simple query interface specification for learning repositories", CEN Workshop Agreement (CWA 15454).

[32] B. Simon, M. Pulkkinen, *ICOPER Reference Model Specification Draft*, <http://www.icoper.org/results/deliverables>, May, 2010.

[33] Google Data Protocol: <http://code.google.com/apis/gdata/>.

[34] H. Van de Sompel, M. L. Nelson, C. Lagoze, S. Warner, "Resource Harvesting within the OAI-PMH Framework", *D-Lib Magazine*, November 2004, Volume 10 Number 12, [doi:10.1045/december2004-vandesompel](https://doi.org/10.1045/december2004-vandesompel).

[35] SWORD Project: http://www.ukoln.ac.uk/repositories/digirep/index/SWORD_Project.

About the Authors



Stefaan Ternier is an assistant professor at the Centre for Learning Sciences and Technologies (CELSTEC) at the Open University of the Netherlands. His research interests are in architectures for learning objects and in mobile learning. Stefaan is involved in European projects, like Share.TEC, that deals with sharing digital resources in the teaching education community, and GRAPPLE, that covers adaptive personalized learning environments. Furthermore, he has been active in the CEN ISSS Learning Technologies Workshop where he coordinated the Simple Publishing Interface (SPI) standardization work.



David Massart is Senior Manager at the European Schoolnet where he leads research and development activities around the [Learning Resource Exchange](#) (LRE). He is the [ASPECT](#) project manager, and is active in the IMS Global Learning Consortium as a member of the Technical Advisory Board Steering Committee and a co-chair of the [Learning Object Discovery and Exchange](#) (LODE) working group. David is (co-)author of scientific papers, technical reports and specifications in the field of the discovery and exchange of learning resources. Since 2007, he has organized an International workshop on Search & Exchange of e-learning Materials (SE@M).



Michael Totschnig, PhD, currently works for Wirtschaftsuniversität Wien (WU) as a senior researcher in the ICOPER project, where he is managing the development of the Open ICOPER content space, a repository architecture providing access to a critical mass of high-quality, re-usable content. He has been working as a system architect for Learn@WU, the e-learning platform of WU, where he was responsible for designing the content authoring environment and for standards implementation (SCORM and IMS QTI). At Knowledge Markets Consulting GesmbH he has managed the development of learning object repository technologies, based on established open source components, that have been deployed for Bildungspool, the Austrian national repository for learning resources for schools, and for Educnext, an exchange platform for higher education learning material. He earned his PhD in communication at the Université du Québec à Montréal. He is participating in the work on the SPI standard of the CEN/ISSS workshop on learning technologies.



Joris Klerkx is a researcher at the Computer Science department of the Katholieke Universiteit Leuven. Dr. Klerkx's research interests include user experience design (i.e. information visualisation, faceted search, multi-touch, mobile devices), metadata, and flexible access to a global learning infrastructure based on open standards in general. Joris serves on the ARIADNE Foundation, and has been involved in numerous projects that focus on educational content discovery such as eContent*plus* projects as MACE, MELT, ASPECT and ICOPER, the EU FP7 projects STELLAR, ROLE, and iTec. He is also involved in several CEN WS-LT work items.



Jerik Duval is a professor of computer science at the Katholieke Universiteit Leuven, Belgium. His research focuses on management of and access to data and content. This work is applied in the fields of technology-enhanced learning, music information retrieval, and technology support for Science2.0®. More specifically, his interests include metadata, reusable content, global infrastructures based on open standards, and mass personalization (The Snowflake Effect). Professor Duval serves on the ARIADNE Foundation, chairs the IEEE LTSC working group on Learning Object Metadata, is a fellow of the AACE, and a member of the ACM and the IEEE Computer Society. He cofounded two spin-offs that apply research results for access to music and scientific output.

Copyright © 2010 Stefaan Ternier, David Massart, Michael Totschnig, Joris Klerkx, and Erik Duval

PRINTER-FRIENDLY FORMAT

[Return to Article](#)
