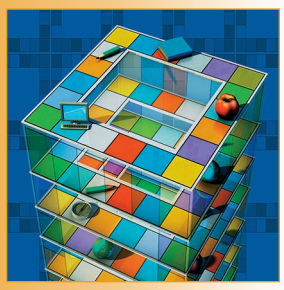

The Ariadne Infrastructure for Managing and Storing Metadata

Reusing digital resources for learning has been a goal for several decades, driven by potential time savings and quality enhancements. Although the rapid development of Web-based learning has increased opportunities for reuse significantly, managing learning objects and making them accessible still entails many challenges. This article presents and analyzes the standards-based Ariadne infrastructure for managing learning objects in an open and scalable architecture. The architecture supports the integration of learning objects in multiple, distributed repository networks. The authors capture lessons learned in four architectural patterns.

Stefaan Ternier, Katrien Verbert, Gonzalo Parra, Bram Vandeputte, Joris Klerkx, and Erik Duval
Katholieke Universiteit Leuven

Vicente Ordóñez and Xavier Ochoa
Escuela Superior Politécnica del Litoral

In the e-learning community, interest is growing in reusing learning objects (LOs),[1] defined as "any entity, digital or non-digital, that may be used for learning, education, or training." LOs are often described with standardized metadata using the IEEE Learning Technology Standards Committee (LTSC) Learning Object Metadata (LOM) standard. Users and systems can use metadata to retrieve LOs in various innovative ways (faceted search, social recommendation, and so on) and for purposes such as attribution and capturing life-cycle or license information.

The Ariadne infrastructure for managing LOs is a distributed network of repositories that encourages the sharing and reuse of such objects. Ariadne was initiated in 1996 by the European Commission's telematics for education and training program. Since then, an infrastructure has been developed in Belgium, Switzerland, France, and Ecuador to produce reusable learning content, including distributed storage and discovery. Since its launch, Ariadne's core software evolved from a highly coupled to a loosely coupled style, based on standards for distributed digital resource management.[2] Today, Ariadne supports several domain-specific networks, too. Here, we examine Ariadne's metadata management components and discuss how to apply them in various networks. We also look at various architectural patterns that generalize our work.

1089-7801/09/$25.00 © 2009 IEEE

Ariadne Components

The core Ariadne infrastructure has several components: the *repository* offers persistent management of LOs and metadata; the *federated search engine* supports transparent search within a network of heterogeneous repositories; the *finder* is a Web client for searching and publishing; the *harvester* collects metadata from external repositories; and the *metadata validation service* validates metadata against metadata application profiles.

Repository

The Ariadne repository features both metadata and object stores for persistently managing LOs and LOM instances. To enable stable search, publishing, and harvesting, the repository provides a search interface based on the Simple Query Interface (SQI) specification,[3] a publishing interface based on the Simple Publishing Interface specification,[4] and a harvesting interface based on the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH).[5]

SQI lets the repository interoperate with different query languages (for example, the ProLearn Query Language [PLQL],[6] the Contextual Query Language [CQL], or the Query Exchange Language [QEL][7]) and metadata standards (such as LOM, Dublin Core [DC], or MPEG). SPI also allows interoperability for ingesting LOs and metadata instances, and OAI-PMH enables metadata collection from various repositories.

Because SQI, SPI, and OAI-PMH hide the metadata storage paradigm's structure, the repository component enables loosely coupled integrations with external applications (which we describe in a later section).

Federated Search Engine and Registry

The federated search engine relies on SQI to offer transparent search to a network of repositories. This engine federates incoming queries to SQI-enabled repositories that it dynamically loads from a registry. The registry component is separate in this architecture, which

- lets other federations retrieve repository metadata available in the Ariadne federation;
- allows interchangeability with other components, thus enabling other registries to reuse the Ariadne federated search engine and promoting a strict separation between searching multiple repositories and managing different repositories in a federation; and
- enables repository metadata management, allowing for intelligent query routing. For example, systems won't send a query containing keywords from a medical thesaurus to a repository that contains only computer science materials.

The federated search engine awaits results from repositories, aggregates them, and sends them to the originating query tool.

Finder

Figure 1 shows the Ariadne finder, which lets users search educational content and browse the results. It also lets them authenticate with OpenID and publish LOs.

The finder hides the protocols and standards for the user and supports faceted federation searching through its connection with the federated search engine.

Among its capabilities that facilitate sharing, the finder lets users subscribe to searches via syndication feeds so that they can easily receive updates for individual search results.

Harvester

Figure 2 shows the Ariadne harvester, which builds on OAI-PMH[5] and manages an internal registry of OAI-PMH targets. For each target, it maintains basic parameters, such as

- the base URL,
- an enumeration of the harvested OAI-PMH sets,
- the metadata prefix that identifies the standard or application profile, or
- the metadata provider.

After harvesting, this component publishes the metadata through SPI in one or more repositories. To manage harvesting in a flexible way, the harvester employs *incremental harvesting*, which uses the date-range queries offered in OAI-PMH, as well as a scheduling mechanism, so that incremental harvesting occurs regularly.

Metadata Validation Service

An important feature of the harvester is its integration with a metadata validation service. The harvester uses this service to validate each

Figure 1. Ariadne finder search and browse interface. The finder lets users search educational content and browse the results but hides protocols and standards from them.

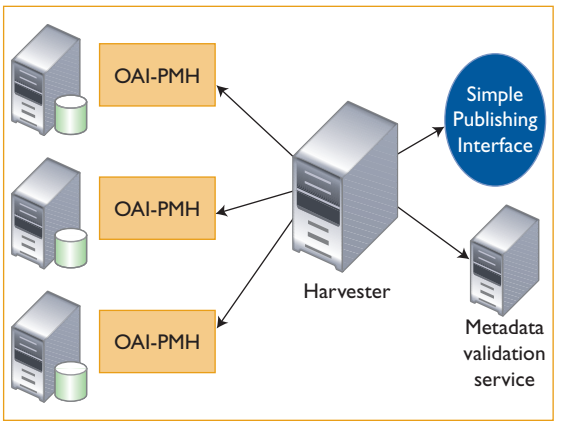

Figure 2. Ariadne harvester. The harvester manages an integral registry of Open Archives Initiative Protocol for Metadata Harvesting targets.

individual target's metadata against a specific validation scheme and automatically creates a validation report.

Ariadne currently harvests from more than 20 LO repositories. Typically, all repositories in a network export metadata instances that conform to an application profile. In our experience, mapping local metadata schemas to a global metadata application profile is an error-prone process. Erroneous metadata instances often render errors or user interface inconsistencies in tools. Because manually checking all harvested instances doesn't scale, we automate this process.

The metadata validation service integrates various state-of-the-art metadata validation components, such as XML Schema validation, Schematron (http://xml.ascc.net/resource/schematron/), and third-party vcard and vocabulary validators. For every application profile an implementation of this component supports, it maintains a validation scheme URI that identifies a specific configuration for the validation components.

A Global Repository Network

Several worldwide initiatives have implemented digital repositories to encourage reusing digital resources (see the "Related Work in Repository Architectures" sidebar for further details). Examples include Edna (www.edna.edu.au); the Multimedia Educational Resource for Learning and Online Teaching (Merlot; www.merlot.org); Lornet (www.lornet.org); the Center for Open Sustainable Learning (COSL; http://cosl.usu.edu); the Korean Education and Research Information Service (Keris; http://english.keris.or.kr/es_main/); the Japanese National Institute of Multimedia Education (NIME; www.nime.ac.jp/index-e.html); and the Latin-America Community of Learning Objects (Laclo; www.laclo.org).

These initiatives are members of the Global Learning Objects Brokered Exchange (Globe) alliance of educational repositories (http://globe-info.org). Globe provides a distributed network of LOs that builds on the IEEE LOM, SQI, and OAI-PMH standards.

Figure 3 illustrates how Ariadne federates searches to the Globe network. The federated search engine receives SQI queries from various search applications, such as the finder, and search portals, such as ProLearn (www.prolearn-academy.org). The engine then distributes the search requests to various types of SQI targets, including direct implementations on repositories or repository caches that contain harvested metadata.

Metadata for Architectural Contents (MACE) and Metadata Ecology for Learning and Teaching (MELT) are examples of networks that have adopted many Ariadne tools. MELT is a European eContent+ project focused on harvesting metadata in a school context. We supported Melt partners in setting up OAI-PMH targets that disseminate LOM records according to the MELT application profile. Ariadne's validation service lets developers automatically test their targets and ensures that metadata caches harvest only valid metadata instances.

The MACE network, also in Europe, aggregates repositories that specialize in LOs on architecture. The MACE-harvested metadata repository contains metadata from all repositories that participate in MACE (see Figure 3). To let the MACE community search in various ways (for example, using a map, faceted search, or a classification browser), the MACE enrichment toolkit lets indexers extend metadata instances. The network illustrates how supporting a rich metadata application profile that includes, for instance, GPS coordinates is beneficial when building innovative search applications.

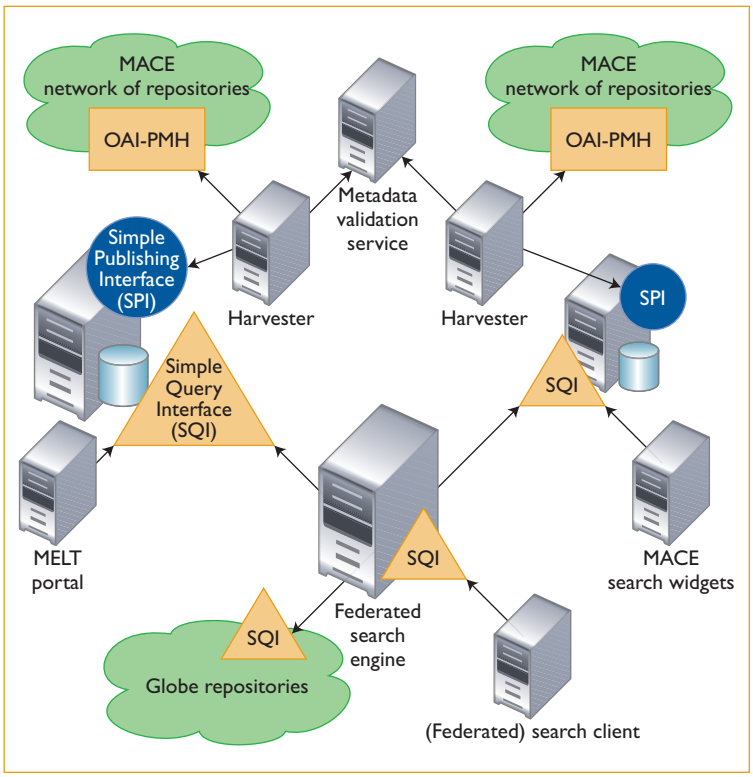

Figure 3. Connecting repositories through federated search and harvesting. Ariadne federates searches to Globe repositories and the Metadata for Architectural Contents (MACE) and Metadata Ecology for Learning and Teaching (MELT) repository networks.

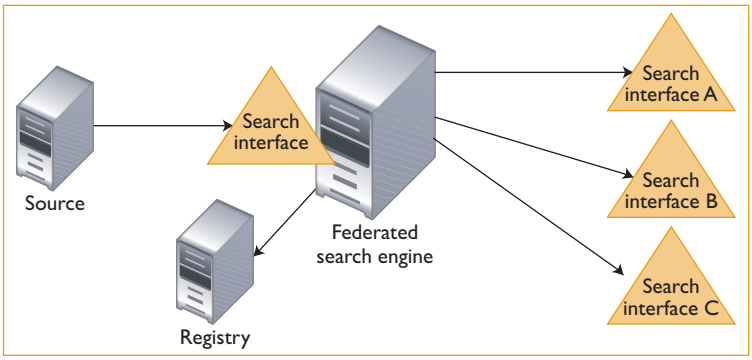

Figure 4. Federated search pattern. Search clients can use this pattern to access an entire network via one interface.

Architectural Patterns for Searching Repositories

While developing Ariadne, we identified several architectural software patterns that are useful not only for building new software applications but also for understanding existing distributed repository initiatives.

Federated Search Pattern

The *federated search pattern* lets search clients avoid maintaining connections with several repositories by giving them access to one search interface through which they can search an entire network. This can minimize communication and dependencies between a source application and the different networks that offer search interfaces. Figure 4 illustrates this single-interface-based pattern.

A registry maintains a list of repositories with which the federated search engine can interact. Furthermore, this registry can document

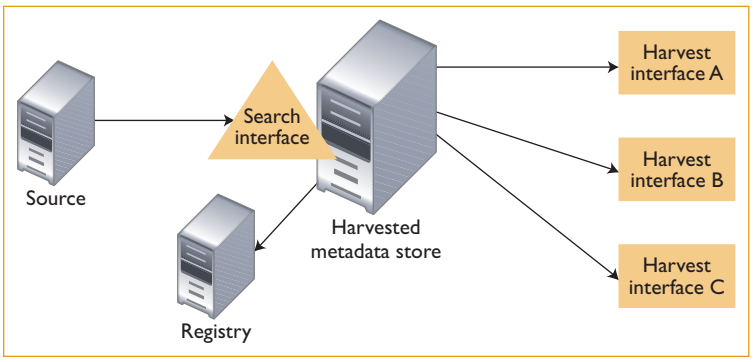

Figure 5. Search on harvest pattern. Metadata are harvested in an intermediate repository to enable search access to the repository.

properties that let search clients select a subset of search interfaces.

System architects can employ the federated search pattern when

- a source needs to access multiple search interfaces transparently;
- searching the most recent information is important (when searching harvested metadata, users aren't always searching the latest version);
- a metadata cache can't harvest the metadata (search engines such as Google have proprietary search or recommendation algorithms based on metadata they aren't willing to expose); or
- a single metadata store can't manage all the metadata in a network.

Given that repositories expose standards that are supported within the federation, managing and maintaining search access to repositories is easy.

On the other hand, when searches are federated to external parties, the federated search engine's response time relies on those parties' response time. Improving this response time might thus require business-level agreements with network partners. This contrasts with harvesting, in which we can set up a local search index and issue optimizations on the cache.

This pattern doesn't specify which machine should host the federated search engine. A client-side implementation shifts computing power to a search client and reduces infrastructure costs at the client's expense. A server-side implementation comes at the cost of setting up and maintaining such an engine but makes deploying updates to the searching strategy transparent to users.

Search on Harvest Pattern

The *search on harvest pattern* uses an intermediate repository to support searching a repository that exposes a harvesting interface.

Implementing a search API on a repository might not always be possible. A system architect might, for instance, be concerned with the load from a large number of queries. Some communities use this pattern to set up a searchable repository that contains all metadata in the community. As Figure 5 illustrates, the *harvest interface* exposes metadata to harvesters. The *harvested metadata store* contains all metadata available in the network. This component preferably exposes this metadata through a search interface.

We can use the search on harvest pattern to

- create a search engine by harvesting all metadata into a metadata store; or
- give a client search access to a repository that only supports harvesting. Without this pattern, a client would have to harvest all metadata locally. By using this pattern, many clients can benefit from the same search interface.

Because metadata are stored in a local cache, system architects are free to decide which query languages the metadata store will provide to the source applications and what technology to use to implement searching.

An architect can also set up redundant harvested metadata stores, which improves the search service's reliability, scalability, or availability.

This pattern has certain implementation issues:

- *Selective harvesting*. When a harvesting interface implements selective harvesting, a harvester can limit communication. For instance, allowing selective harvesting based on the last modification date, a harvester can copy only modified records.
- *Stale data*. Users might delete or change the metadata on the original repository while an old copy is still available in the harvested metadata store.
- *Host location*. Similar to the federated search pattern, this pattern doesn't specify which machine should host the harvested metadata store. If only one user application is to use the harvested metadata, both the applica-

Table 1. Overview of architectural design patterns.	
Federated search pattern	Search clients have access to several repositories via a single search interface through which they can search an entire network.
Search on harvest pattern	Metadata are harvested in an intermediate repository to enable search access to the repository. We can use this pattern when it isn't possible to implement a search API on top of a repository.
Search adapter pattern	One search API is converted into another to enable uniform search access.
Harvest adapter pattern	A search interface is converted into a harvest interface. Implementing the harvest adapter pattern makes repositories that already offer a search interface harvesting-compliant.

tion and the harvested metadata store should be hosted on the same machine, limiting network traffic. When many applications require search access, they can share one harvested metadata store.

Although the search on harvest pattern suffers from stale data, we favored it in practice over the federated search pattern because it creates an environment in which controlling search services is easy.

Search Adapter Pattern

The *search adapter pattern* converts one search API into another, letting search applications search repositories that they couldn't otherwise reach due to incompatible interfaces.

Sometimes an application can't search a repository because it doesn't support any of the search protocols that repository provides. If developers can't extend the set of protocols that the source or the target application supports, the search adapter pattern can solve this problem. This pattern involves a wrapper service that translates requests from one protocol into another.

The search adapter pattern works well when a limited number of search specifications dominate the market. Creating search adapters rather than extending the target with the protocol lets other repositories reuse the adapter. However, the disadvantage of using a search adapter is potential information loss.

The main implementation issue is *separation of concerns*. We can use SQI in combination with other query languages and metadata schemas. Many other APIs, such as Search/Retrieve via URL and Search/Retrieve via Web Service (SRU/W)[8] or the Merlot query API, hardcode the metadata schema or query language. When the metadata and query language aren't separated, adapter implementations aren't generic. An implication is that all implementations work only for a given profile of the search service (for example, SQI/SOAP with PLQL[6] and LOM).

Harvest Adapter Pattern

The *harvest adapter pattern* converts a search interface into a harvest interface, enabling a harvest consumer to retrieve metadata instances from a repository that already offers a search interface.

Infrastructures for harvesting often require repositories to implement a protocol that supports a given metadata schema. In OAI-PMH, for instance, the DC metadata schema is mandatory, such that all OAI-PMH can rely on it. Implementing the harvest adapter pattern makes repositories that already offer a search interface harvesting-compliant. We might also use this pattern when a search interface doesn't feature certain criteria (if a query language isn't expressive enough or response time is too slow, for example). By harvesting through the search interface, an application can build a search index that fits the search application's requirements.

Query services, that don't maintain the same result set over subsequent requests risk giving inconsistent results. So, this pattern is applicable only on top of query services that keep the cursor consistent over subsequent result retrieval. Additionally, a trade-off exists when applying this pattern: some search interfaces don't return all metadata through a search protocol. For example, if a repository ranks results based on usage metadata that aren't returned, searches on the harvested metadata will be inferior. Other implementation issues are discussed elsewhere.[9,10]

Table 1 summarizes the architectural design patterns for searching distributed repositories we've deduced in our work. The Ariadne federated search engine is an instance of the federated search pattern. We connected the MACE and MELT communities using the search

on harvest pattern (see Figure 3). Finally, we applied the search adapter pattern to many Globe repositories that already provided a search API.

To manage the repositories that systems can search in a network, our future work will focus on setting up a repository registry. Recording the standards that a repository supports (such as SQI or OAI-PMH) will let middle-layer applications intelligently distribute queries. This requires

- an open initiative that interested parties can join and
- modeling federations. As Globe forwards queries to other federations, the registry should record the repositories that are available through them.

As a final remark, we note that the shift in architecture from managing a single repository to creating and managing a repository network is the only logical step to enable an LO economy in which an abundance of learning material replaces the current perceived scarcity due to repository isolation.

Acknowledgments

Part of this work has been funded by the European Commission in the eContent+ program ECP-2005-EDU-038098 (MACE) and ECP-2005-EDU-038103 (MELT).

References

1. R. Robson, "Context and the Role of Standards in Increasing the Value of Learning Objects," *Online Education Using Learning Objects*, R. McGreal, ed., Routledge/Falmer, 2004, pp. 159–167.
2. S. Ternier and E. Duval, "Interoperability of Repositories: The Simple Query Interface in Ariadne," *Int'l J. E-Learning*, vol. 5, no. 1, 2006, pp. 161–166.
3. B. Simon et al., "A Simple Query Interface for Interoperable Learning Repositories," *Proc. 1st Workshop Interoperability of Web-Based Educational Systems*, CEUR, 2005, pp. 11–18.
4. S. Ternier et al., "A Simple Publishing Interface for Learning Object Repositories," *Proc. World Conf. Educational Multimedia, Hypermedia, and Telecommunications*, Assoc. for the Advancement of Computing in Education, 2008, pp. 1840–1845.
5. C. Lagoze and H. Van de Sompel, "The Open Archives Initiative: Building a Low-Barrier Interoperability Framework," *Proc. 1st ACM/IEEE-CS Joint Conf. Digital Libraries*, ACM Press, 2001, pp. 54–62.
6. S. Ternier et al., "Interoperability for Searching Learning Object Repositories: The ProLearn Query Language," *D-Lib Magazine*, vol. 14, nos. 1–2, 2008.
7. W. Nejdl et al., "Edutella: A P2P Networking Infrastructure Based on RDF," *Proc. 11th Int'l Conf. World Wide Web*, ACM Press, 2002, pp. 604–615.
8. S.H. McCallum, "A Look at New Information Retrieval Protocols: SRU, OpenSearch/a9, CQL, and XQuery," *World Library and Information Congress: 72nd IFLA General Conf. and Council*, Int'l Federation of Library Assoc. and Institutions (IFLA), 2006.
9. R. Sanderson, J. Young, and R. LeVan, "SRW/U with OAI: Expected and Unexpected Synergies," *D-Lib Magazine*, vol. 11, no. 2, 2005.
10. T.W. Cole et al., *Implementation of a Scholarly Information Portal using the Open Archives Initiative Protocol for Metadata Harvesting*, tech. report, Univ. Illinois at Urbana-Champaign, 2003.

Stefaan Ternier is an assistant professor at the CELSTEC Center at the Open University of the Netherlands. Previously, he worked as a researcher in the computer science department at Katholieke Universiteit Leuven, Belgium. His research interests are in architectures for learning objects and interoperability. Ternier was involved with the Metadata for Architectural Contents (MACE) and Metadata Ecology for Learning and Teaching (MELT) eContent+ projects and the Global Learning Objects Brokered Exchange (Globe). Contact him at stefaan.ternier@ou.nl.

Katrien Verbert is a postdoctoral researcher in the computer science department at Katholieke Universiteit Leuven, Belgium. Her research interests include learning object content models, metadata, learning object reuse, and interoperability. Verbert is involved with the Responsive Open Learning Environments (ROLE) and Sustaining Technology Enhanced Learning Large-scale Multidisciplinary Research (Stellar) projects and the Standard for Learning Technology – Conceptual Model for Resource Aggregation for Learning, Education, and Training (RAMLET) IEEE Learning Technology Standards Committee standardization project to develop a reference model for resource aggregation. Contact her at katrien.verbert@cs.kuleuven.be.

Gonzalo Parra is a researcher in the computer science department at Katholieke Universiteit Leuven, Belgium. His research interests include metadata generation and manipulation, contextual attention metadata, and flexible access to information based on open standards. Parra has a master's degree in engineering from Escuela Superior Politécnica del Litoral. Contact him at gonzalo.parra@cs.kuleuven.be.

Related Work in Repository Architectures

As an open source, standards-based architecture for managing digital resources, Ariadne has some commonalities with the Flexible and Extensible Digital Object and Repository Architecture (Fedora)[1] and the Adore Digital Object Repository.[2] Fedora is an architecture for digital libraries, institutional repositories, and learning object repositories, designed to manage complex digital objects. It uses an RDF-based model to represent relationships among digital objects and their components. Similarly, Adore provides a standards-based repository for managing and accessing digital objects. Objects are encoded in XML using the Metadata Encoding and Transmission Standard (METS) in Fedora and the MPEG-21 Digital Item Declaration Language (DIDL) in Adore, whereas Ariadne uses Learning Object Metadata for describing digital resources and their interrelationships. The Fedora and Adore repository services for managing digital resources relate to repository services in Ariadne. Fedora features a deposit API through which applications can deposit objects on a Fedora server, which relates to the Simple Publishing Interface specification. As in Ariadne, Fedora and Adore use the Open Archives Initiative Protocol for Metadata Harvesting. In Adore, object dissemination services are also available through OpenURL.[3]

Other architectures include the Content Object Repository Discovery and Registration/Resolution Architecture,[4] Open Knowledge Initiative (OKI), Lorenet (www.lorenet.nl), Metadata-Based Repository Search Components in Open Source (http://meresco.org), LionShare (http://lionshare.its.psu.edu), and Edutella.[5] Like Fedora and Adore, Lorenet and Meresco use the search on harvest pattern we describe in the main text, whereas LionShare and OKI have adopted the search adapter pattern. Edutella[5] is a peer-to-peer network for interconnecting learning object repositories. Like Adore and Fedora, it builds on RDF.

We're currently redeveloping the core Ariadne infrastructure using Semantic Web technologies to explore their potential. This Semantic Web version will let us compare the current traditional approach with emerging Semantic Web-based approaches for knowledge sharing and reuse.

References

1. T. Staples, R. Wayland, and S. Payette, "The Fedora Project: An Open-source Digital Object Repository Management System," *D-Lib Magazine*, vol. 9, no. 4, 2003.
2. H. Van de Sompel et al., "Adore: A Modular, Standards-Based Digital Object Repository," *The Computer J.*, vol. 48, no. 5, 2005, pp. 514–535.
3. H. Van de Sompel and O. Beit-Arie, "Open Linking in the Scholarly Information Environment Using the OpenURL Framework," *D-Lib Magazine*, vol. 7, no. 3, 2001.
4. H. Jerez et al., "Adl-r: The First Instance of a Cordra Registry," *D-Lib Magazine*, vol. 12, no. 2, 2006.
5. W. Nejdl et al., "Edutella: A P2P Networking Infrastructure Based on RDF," *Proc. 11th Int'l Conf. World Wide Web*, ACM Press, 2002, pp. 604–615.

Bram Vandeputte is a researcher in the computer science department at Katholieke Universiteit Leuven, Belgium. His research interests include metadata, learning objects, harvesting, metadata validation, and flexible access to repositories. Vandeputte is responsible for the development of the Ariadne harvester and the validation service. Contact him at bram.vandeputte@cs.kuleuven.be.

Joris Klerkx is a postdoctoral researcher in the computer science department at Katholieke Universiteit Leuven, Belgium. His research interests include information visualization, metadata, learning objects, and flexible access to a global learning infrastructure based on open standards. Klerkx is involved with the MACE, MELT, Adopting Standards and Specifications for Educational Content (Aspect) and Interoperable Content for Performance in a Competency-driven Society (ICOPER) eContent+ projects, and Globe. Contact him at joris.klerkx@cs.kuleuven.be.

Erik Duval is a professor in the computer science department at Katholieke Universiteit Leuven, Belgium. His research interests include metadata, learning objects, a global learning infrastructure based on open standards, and mass personalization. Duval has a PhD in engineering from Katholieke Universiteit Leuven. He serves as co-president of the Ariadne Foundation and chair of the IEEE LTSC working group on learning object metadata. Contact him at erik.duval@cs.kuleuven.be.

Vicente Ordóñez is a member of the research group on technology-enhanced learning at the Information Technology Center at Escuela Superior Politecnica del Litoral (ESPOL), Ecuador. His main research interests include multimedia processing and understanding. Ordóñez has a master's degree in engineering from ESPOL. Contact him at vicente.ordonez@gmail.com.

Xavier Ochoa is a professor of electrical and computer engineering at Escuela Superior Politecnica del Litoral, Ecuador. His main research interests include measuring the learning object economy and its impact on learning. Ochoa has a PhD in engineering from the Katholieke Universiteit Leuven. He coordinates the research group on technology-enhanced learning at the Information Technology Center at ESPOL. Contact him at xavier@cti.espol.edu.ec.