Supervision of Heterogeneous e-Learning Environments: the case of the International E-MI@GE Project

Supervision of International E-MI@GE

Julien Broisin, Valentin Butoianu, Philippe Vidal, Daniel Marquié
Institut de Recherche en Informatique de Toulouse
IRIT
Toulouse, France
{broisin,butoianu,vidal,marquie}@irit.fr

***Abstract*—**The International E-Miage project (IEM) is an engineering curriculum composed of more than twenty international sites that deliver official French degrees. The IEM computational environment comprises many tools, platforms and users that make it very difficult to establish an up-to-date global overview of the whole set of learning entities implied into this digital campus. Therefore, we present in this paper a tracking framework able to produce, exchange and store in a unified way all activities performed by users on the various IEM learning systems and resources. A rich Internet application has been designed to demonstrate how the collected data can be computed to build a real-time and detailed cartography of each IEM partner, or to produce useful statistics information about a (set of) partner.

Keywords- e-learning environments; distributed architecture; supervision; user context information

I. INTRODUCTION

The digital campus International E-Mi@ge (IEM) [5] aims at offering a distant teaching version of the MIAGE curriculum (*Méthodes Informatiques Appliquées à la Gestion des Entreprises*) in order to (1) reach a new public, (2) improve the current modes of teaching, in particular by providing online resources to traditional students and by catching academic and professional teachers attention on new teaching methods based on ICT, and (3) give the MIAGE curriculum the opportunity of being a privileged e-learning actor in the field of data processing and information systems.

We focus here on the computational aspect of the IEM project. The consortium currently comprises about 150 to 200 people (authors, teachers, learners, technical administrators, etc.) involved in the process of creation, indexation, presentation and consultation of the content, and is composed of about ten heterogeneous systems. These systems comprise several Learning Management Systems (LMS) deployed within the IEM sites and used to offer some learning services to learners and teachers, one Learning Object Repository (LOR) that stores all the versions of the sixty IEM SCORM packages, and a Content Management System (CMS) to allow members to exchange and share information related to the exams. Thus, the various phases from the creation to the deployment and maintenance of a learning resource involve an important number of e-learning actors and systems that makes it very difficult, even nearly impossible, to produce an up-to-date computational view of the whole set of learning systems and resources. This can lead to unacceptable mistakes such as a student that complained during an exam session to misunderstand the question because several notions where not even introduced into the online course. Indeed, the exams are common to all students of the consortium, but the LMS manager of each site has to deploy, at the beginning of each semester, the last version of all SCORM modules available within the learning object repository; in the above case, an author uploaded a new version of a module within the LOR without notifying the others sites that did not updated their resources on the learning management system.

A solution to this issue consists of tracking users' activities and analyzing these activities in order to supervise the whole set of IEM learning tools. Current systems that tackle this issue such as Mendeley [8] or APOSDLE [2] are restricted to a specific system and can hardly integrate data produced by a third-party application. Therefore, the main contribution of this paper resides in the design and integration of a tracking framework within the existing IEM computational environment. The resulting distributed architecture is able to record activities performed by users on each tool of the IEM environment, and to externalize these records to a central repository. Then, tracking information that has been collected can be visualized in a uniform way and correlated together to establish a cartography of the whole set of sites or to produce relevant statistical data.

The paper is organized as follows: the next section exposes IEM from a computing point of view and focuses on the heterogeneous nature of the tools implied into this e-learning environment. Section 3 introduces the main principles and architecture of the tracking framework, and demonstrates how the various tools integrated within the IEM project can be supervised in a uniform way. In section 4, we show how it is possible to get an overview of all actions and activities operated by users on these tools, and present an application dedicated to the visualization of traces and able to produce statistical information. Finally, we conclude and expose some future works.

ISBN: 978-1-61208-120-5

II. Architecture of the IEM Environement

The computing environment of the IEM project is illustrated in Figure 1. It comprises, from top to bottom:

- Some authoring tools dedicated to the creation of the pedagogical content.
- A Content Management System based on XOOPS to share exams and related information between the IEM administrative staff. Since this kind of system is not compliant with ubiquitous metadata standards dedicated to learning objects, it can hardly be used as a space to store the pedagogical IEM content.
- A Learning Object Repository stores both the pedagogical resources created by the authors and the metadata that describe the content. A graphical user interface associated to the repository allows users to retrieve, edit, download and index learning objects.
- A Learning Management System deployed within each site; some of them exploit INES (a specific platform developed by the University of Amiens) and others are based on the well-known MOODLE platform.

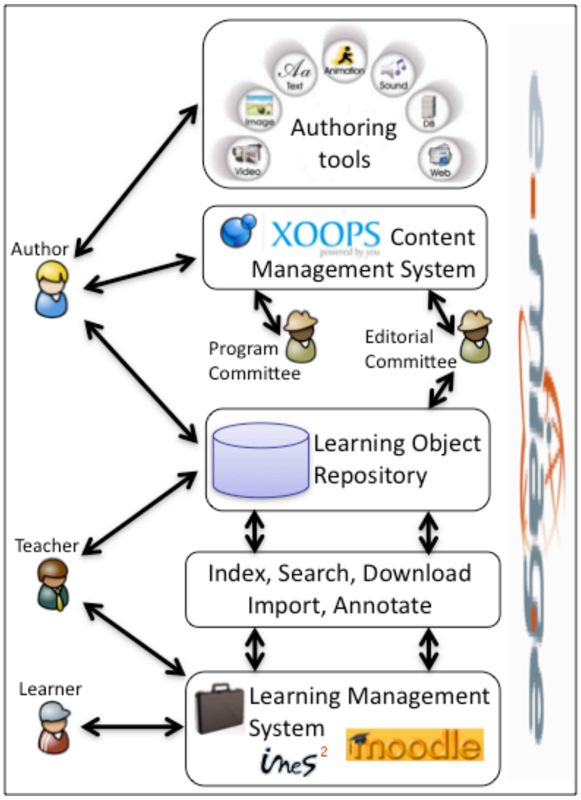


Figure 1. The computing environment of the IEM project.

In addition to these components, we developed an intermediate layer between the storage and the learning environments comprising a set of services that allow transparent communication between these two kinds of systems [3]. These services that are accessible from the LMS focus on learning objects and comprise:

- The indexation service: it allows users to index new learning objects into the LOR so that they can be easily reused by all sites.
- The search service: users can browse the LOR and consult metadata associated with the results returned by the search engine.
- The download service: through this service, end-users are able to download on the local host the content matching with the metadata returned by the search service.
- The importation service: it allows users to import the pedagogical content stored into the LOR within the current course of the LMS so that it is straightly available to learners.
- The annotation service: once a learning object has been deployed within a course, learners, teachers and tutors have the opportunity of suggesting comments and ratings related to the content.

Thus, the IEM project implies a large number of e-learning tools and actors geographically distributed that make it very difficult and time-consuming to establish an overview of the computational environment. To overcome the issues of heterogeneity and repartition of applications, our proposal consists of supervising all activities performed by users on each system and resource implied within the IEM computing environment, and then to externalize and transfer these data to a central component. Two main objectives must then be achieved: the common representation of data produced by the heterogeneous IEM tools, and the elaboration of a distributed architecture ensuring the production, exchange and storage of the tracking information.

III. The E-MI@GE Tracking Framework

The works presented here are based on some previous works in which we introduced a generic framework able to capture users' activities within a web-based learning environment [4]. In the following sections, we briefly present the general approach and focus on how it is applied to the E-MI@GE context.

A. The Basis for our Approach

A standard initiated by the Distributed Management Task Force (DMTF) brings a solution to unify supervision of distributed computing environments, and facilitates exchange of data across otherwise disparate technologies and platforms. To uniformly represent data that are supervised, the DMTF standard adopts a Common Information Model (CIM) [14], a metamodel that exploits object concepts for modeling systems, networks and applications. Moreover, the DMTF recommends a distributed architecture called Web-Based Enterprise Management (WBEM) [16] to support the CIM metamodel. The main components of the WBEM architecture include (1) a manager responsible for storing and managing the supervised information (stored as instances of the CIM models), (2) some providers which are pieces of software that communicate with some supervised resources such as disk drives and CPUs, and (3) a client application which interacts with the manager in order to retrieve and exploit the collected information.

Starting from this basis, we elaborated several models extending the native CIM models and focusing on learning systems, resources and activities.

B. A Common Representation of Data to Supervise

Our previous works defined three generic models to represent entities to supervise within a web-based learning environment:

- The model *TEL_Environment* focuses on learning systems and resources. Systems are described in terms of *name*, *location*, *version* and *description*, whereas resources are represented according to their *title*, *description*, *dates of creation*, *modification* and *deletion*.
- The model *TEL_User* is precisely detailed in [12], and includes the IMS-LIP standard [10] together with some additional cognitive and metacognitive information. This user model describes a user from a learner, teacher or tutor point of view.
- The model *TEL_Activity* links the two previous models by describing interactions between users and systems/resources. In other words, this model represents the activities performed by a user on a system or resource.

Each of these models presents a high abstraction level and offers the opportunity of defining specific models according to specific objectives. Since our ambition is to establish an overview of the IEM computational environment during the operational phase, the authoring tools are not taken into account within our modeling of systems; indeed, these tools are used during the preparation phase in order to design the learning resources, but they are not involved in the exploitation process. Therefore, in order to meet the requirements of the IEM environment, we specialized the above generic models:

- Three system classes respectively describe learning object repositories (the property *metadata* has been introduced), learning management systems (two specific properties related to their conformance to the *SCORM* and *IMS-LD* standards have been introduced) and content management systems.
- Three classes represent the resources manipulated by the three above types of systems: learning objects, courses (characterized by the property *category*), and exams.

There was no need to extend the user model, as it was exhaustive enough to integrate the user profiles defined within the IEM learning systems. However, depending on the systems, information that is available to describe users may differ:

- Within the learning object repository, a user is defined according to the OpenID standard [15]. Thus, the following information is available: *fullname*, *nickname*, *timezone*, *language* and *gender*.
- The user profile specified in XOOPS includes some basic information such as the *first* and *last names*, *postal* and *electronic addresses* and *country*, but also defines other interesting information from a learning point of view such as the *list of interests*, the *personal web page*, or some *social networks identifiers* that can be further reused to build recommendation and social learning services [6][7].
- The user model specified within INES and MOODLE are different: INES comprises the same basic information as XOOPS, but MOODLE also defines the others interesting (social) learning information together with the *blog entries* a user has produced and the *courses* he/she is enrolled in. The *role* of the user within a course is also gathered.

Finally, the activity model has been specialized to take into account various activities that can be processed on the supervised resources through the learning tools; the whole set of activities we defined for the IEM learning context is listed in Table I. Thus, each action appearing in Table I matches with a class of the specific IEM activity model derived from the generic *TEL_Activity* model, and establishes a link between a user and a resource integrated into a system.

TABLE I. SUPERVIZED ACTIVITIES WITHIN THE IEM FRAMEWORK

	Learning Tool		
	CMS	***LOR***	***MOODLE and INES***
Activities	- Upload and Deletion of an exam - Update of the user profile	- Search for learning objects - Consultation of metadata - Download of content - Indexation of a new LO	- All LOR activities - Creation of a course - Deployment, Consultation, Download and Annotation of a SCORM package

To support this data model, we set up a distributed architecture conform to the WBEM standard and depicted in the next section.

C. A Distributed Architecture

The aim of this architecture is to ensure production, exchange and storage of the tracking information. Therefore, three distinct types of components in our architecture are respectively responsible for processing the above-mentioned tasks. The resulting framework is illustrated in Figure 2 and comprises:

- The learning context from which tracking information is collected; in the IEM environment, it comprises the CMS, the LOR and several MOODLE and INES servers. These tools embed direct access sensors responsible for capturing the values of the information defined within our tracking models; sensors correspond to the data-gathering layer of the six-layered architecture of [9].
- The tracking environment represents both the storage/management and the query layers. Indeed, the tracking repository is an object-oriented database storing tracking information (it contains classes and instances of the models), whereas the tracking manager is able to manipulate the data stored into

the repository using the CIM Query Language (CQL).

- The intermediate layer (reception layer) acts as a proxy between the learning and the tracking environments to offer an easy access to the tracking repository: the tracking service is able to receive tracking information sent by learning tools, and to search for information stored into the tracking repository.

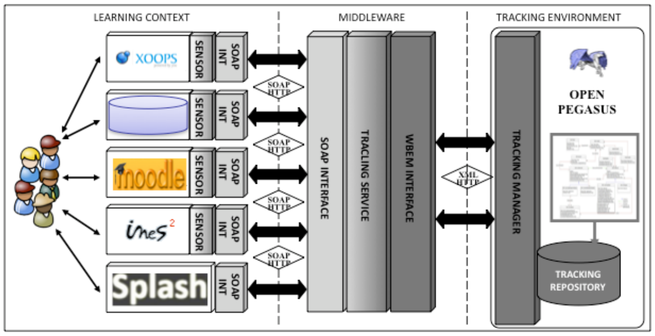


Figure 2. The global tracking architecture.

To make the role of each component of the architecture clearer, the next section details interaction between these entities for a specific use case.

D. Use Case: Inegration of a Learning Object into a MOODLE Course

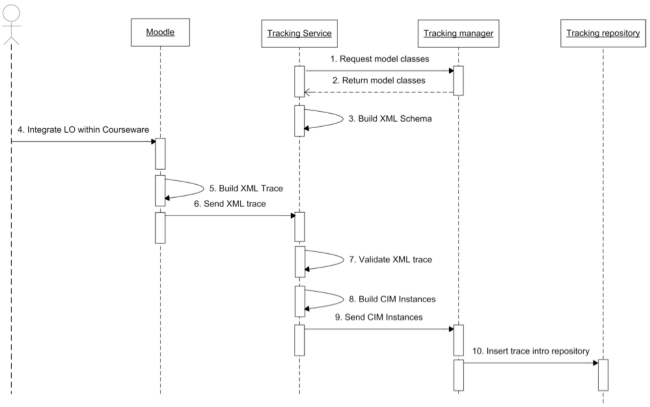


Figure 3. Collecting a trace from a MOODLE server.

Figure 3 represents the UML sequence diagram illustrating exchanges and treatments required to ensure production, exchange and storage of a trace translating the integration of a learning object within a MOODLE course by a teacher. When the tracking service is started, it queries the tracking manager (1) in order to retrieve classes of the tracking model (2), and finally creates the XML schema according to the models' specifications (3). When the teacher integrates the learning resource into the MOODLE course (4), the sensor extracts the tracking information from MOODLE and creates the matching XML trace (5); this trace contains information about the user, the learning object, the course, the learning system and the activity. The sensor then sends a request to the tracking service (6) which validates the XML trace against the XML schema produced in step 3 (7), and finally builds the matching instances of the models (8). These lasts are sent to the tracking manager (9) and stored into the tracking repository (10).

In this architecture, the gathering sensors only are specific to the systems to supervise. The next section details how the sensors have been designed for the IEM context according to the information they have to collect, and shows how a map of the whole E-MI@GE consortium can be established from these data.

IV. SUPERVISING E-MI@GE SITES

A. Designing the gathering sensors

The gathering sensors are responsible for capturing the values of the information defined within our tracking models. According to these models, a certain amount of data summarized in Table II must be located and extracted from the various tools of the IEM framework; to achieve this task, we studied the databases and session variables of each IEM learning system.

TABLE II. TRACKING INFORMATION AVAILABLE WITHIN THE IEM FRAMEWORK

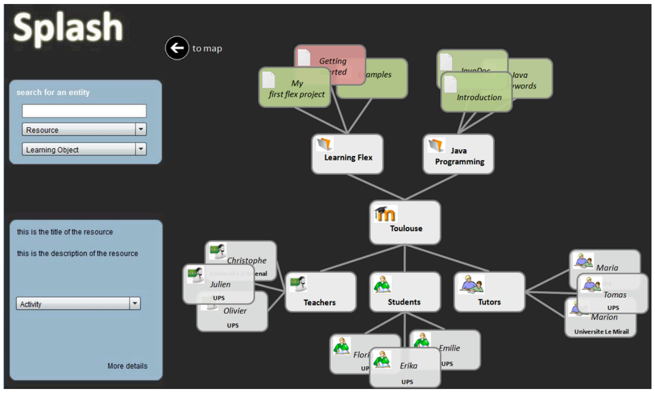
Entity	**Learning Tool**		
	XOOPS	***LOR***	***MOODLE and INES***
Users	First name, last name, postal adress, electronic address, country, list of interests, personal web page, social network identifiers	Full name, nickname, time zone, language, gender	All XOOPS properties, blog entries, courses, role
Systems	Name, location, version, description	All XOOPS properties, metadata	All XOOPS properties, SCORM, IMS-LD
Learning objects		Title, description, location, date of creation, date of modification, date of deletion	Title, description, location, date of creation, date of modification, date of deletion
Courses			Title, description, location, date of creation, date of modification, date of deletion, category
Exams	Title, description, location, date of creation, date of modification, date of deletion		

A gathering sensor is only between 50 and 100 lines of source code, as the main treatments operated on the trace (transformation of the XML trace into the matching instances of our model, and storage of these instances into the tracking repository) are ensured by the tracking service; the communication between the sensor and the tracking service is established using the web services technologies). The process of delegating tasks presents two main advantages: the original behavior of the learning system is not altered (the XML trace is very short string, and the AJAX technologies used to send the request from the sensor to the tracking service reinforce this capacity), and the tracking information is collected in real time as soon as the user performs a supervised activity. Moreover, XOOPS, MOODLE and INES are all PhP-based software, so the integration of the sensors within these tools was facilitated by the reuse and adaptation of the source code.

Let us note that data integrity is ensured within the IEM tracking framework by the models themselves. As an example, even if data describing users are gathered from several applications, they are merged together into the tracking repository so that a user is described by a single profile containing information gathered from XOOPS, MOODLE, INES and the LOR.

B. *Visualizing status of the IEM computional environment*

The collection of tracking information from the IEM applications started recently, but the set of data already available within the tracking repository makes it possible to build applications to exploit these data. A first prototype of SPLASH (Secured and Personal Learning dASHboard) has been developed and acts as a client application that queries the tracking service to retrieve and correlate data in order to provide end-users with relevant and interpretable information.



this is the title of the resource

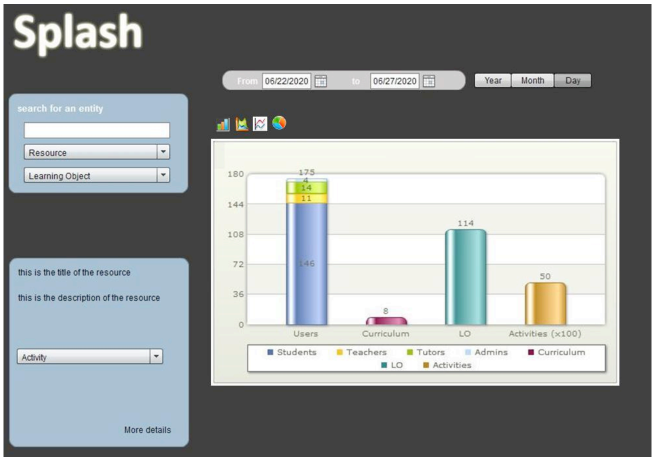
this is the description of the resource

Figure 4. Visualizing the Status of a Site.

The main feature of SPLASH establishes a global overview of the whole IEM computing environment by drawing, on a world map, some points representing the sites (the sites match with the MOODLE and INES instances stored into the tracking repository, and points are drawn according to the location property). When a user activates such a point on the map, several interesting information are available and illustrated on Figure 4:

- The list of courses deployed within the site (*Learning Flex* and *Java programming*). Courses are identified by selecting all associations between the given site instance and the course instances.
- The list of teachers, tutors and students enrolled in the site (a user has to log at least once in MOODLE or INES). Starting from the site and its list of courses, the algorithm selects all associations with user instances.
- The list of SCORM modules deployed within each course (on top of Figure 4); here, the algorithm searches for learning objects instances associated with the given course.

Even if the detailed information appearing in Table II are displayed on the GUI for each entity, the information listed above allows us to get a global point of view on systems, courses, learning objects and people involved into the IEM community only. These data are not precise enough to ensure that all IEM sites are deploying the "correct" SCORM packages. Thus, each learning resource integrated into a course is colored on the GUI: a green color means that the version of the resource matches with the most recent version of the same resource stored into the LOR, whereas the red color denotes a mismatch. SPLASH thus provides the IEM editorial committee with the opportunity of checking the current status of all courses regarding the versioning issue, and to alert the site(s) concerned by a mismatch.



this is the title of the resource

this is the description of the resource

Figure 5. Visualizing Statistical Information about an IEM Site.

Much statistical information can also be computed and displayed with respect to different chart formats (area, pie or stacked column). These statistics can be produced at a raw level (users, systems, resources, activities) or at an abstract level (site, set of sites). The algorithm that computes the raw level statistics implements some basic computing methods, whereas it intelligently correlates the raw level data to build the statistics of a (set of) site. The Figure 5 depicts the statistics of the site of Toulouse as a state column chart: 174 users are registered in the INES platform (146 students, 11 teachers, 14 tutors and 3 administrators), 8 curriculum are currently deployed, 114 learning objects are integrated within these courses, and more than 5,000 activities have been performed. The row level statistics can then be displayed as a pie chart by clicking on the matching column on Figure 5: if the "Activity" column is clicked on, a piece of the pie represents a specific activity, and the weight of a piece is defined according to the number of times the activity has been performed.

As mentioned before, the deployment of the whole IEM tracking framework happened recently. Even if the functionalities currently implemented within SPLASH reveal fundamental information regarding the good health of the computing environment, much more advanced features can be built on top of the tracking information. Thus, several advanced services focusing on the pedagogical area are currently being elaborated into SPLASH. One of these services allows learners to be aware of their colleagues' progress during the learning process, thus increasing their motivation level. Another service encourages the social learning by recommending relevant tools, persons and resources to a specific learner according to his profile, difficulties and context.

V. CONCLUSION AND FUTURE WORKS

We presented in this paper a solution to supervise users' activities within the International E-MI@GE digital campus, a curriculum delivering a French engineering degree. Our proposal stands on a common representation of data collected from the heterogeneous learning tools, together with a distributed architecture managing traces from their creation to their storage and visualization. The content management system, the learning object repository and most of the learning management systems of the IEM computational environment are now integrated into the tracking framework. A visualization application exploits the data collected from these systems to build an up-to-date overview of the whole set of users, systems and resources involved in the project, thus facilitating its management and preventing some important issues to occur.

A short-term perspective consists in tracking the RELOAD authoring tool [13]. Since the IEM pedagogical resources are packaged within SCORM modules using this application, it will be easy to automatically index the whole package into the LOR by extracting the metadata mentioned within it; this process would considerably facilitate the indexation mechanism, since the IEM authors would not have to manually perform this task. Furthermore, we plan to disaggregate the package in order to index learning objects characterized by a smaller granularity level. This would encourage e-learning actors to reuse the IEM content and help promoting the quality of this digital campus: some studies [1] shown that this kind of learning objects are easier to share and reuse by others e-learning actors than resources characterized by a high level of granularity.

Another perspective concerns the user privacy. Indeed, to protect the privacy of the IEM actors, the solution we adopted consists of displaying the users' first name only. But last names are also recorded into the tracking repository, making it easy for a third application to query the tracking service and to retrieve confidential information such as the set of activities performed by a user identified by his first and last names. To tackle this issue, we plan to set up a federated identity mechanism by delegating the management of users' authentication and authorization to a third party; the most common solutions in this area such as the Security Assertion Markup Language (SAML) [11] elaborated by the OASIS consortium or the OpenID protocol [15] have to be considered.

REFERENCES

[1] P. Baumgartner, "The ROI Paradox," Keynote Presentation, Gesellschaft für Medien in den Wissenschaften Conference, Autriche, 2004.

[2] G. Behama, B. Kumpa, T. Leya, and S. Lindstaedt, "Recommending Knowledgeable People in a Work-Integrated Learning System," Procedia Computer Science, vol. 1, issue 2, 2010, pp. 2783-2792, doi:10.1016/j.procs.2010.08.003.

[3] J. Broisin, P. Vidal, D. Marquie, and O. Catteau, "The International E-Miage project: from an Isolated Framework to an Architecture Based on Learning Standards," Proc. Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT-ICIW'06), Feb. 2006, pp. 10, doi: 10.1109/AICT-ICIW.2006.189.

[4] V. Butoianu, P. Vidal, K. Verbert, E. Duval, and J. Broisisn, "User context and personalized learning: a federation of Contextualized Attention Metadata," Journal of Universal Computer Science, John Wiley and Sons, vol. 16, issue 16, 2010, pp. 2252-2271.

[5] G.M. Cochard and D. Marquie, "An E-Learning Version of the French Higher Education Curriculum 'Computer Methods for the Companies Management'", Proc. IFIP International Federation for Information Processing, vol. 156/2004, 2004, pp. 557-572, doi: 10.1007/978-1-4020-8157-6_52

[6] S. Downes, "E-learning 2.0," eLearn Magazine, Association for Computing Machinery, Oct. 2005, available at http://www.elearnmag.org/subpage.cfm?section=articles&article=29-1, consulted on Dec. 2010.

[7] R. Farzan and P. Brusilovski, "AnnotatEd: A social navigation and annotation service for web-based educational resources," New Review of Hypermedia and Multimedia, vol. 14, issue 1, Jan. 2008, pp. 3-32, doi:10.1080/13614560802357172.

[8] V. Henning and J. Reichelt, "Mendeley - A Last.fm For Research?," Proc. Fourth IEEE International Conference on eScience (eScience 08), Dec. 2008, pp. 327–328, doi: 10.1109/eScience.2008.128.

[9] K. Henricksen and J. Iadulska, "Developing context-aware pervasive computing applications: Models and approach," Pervasive and Mobile Computing, vol. 2, issue 1, pp. 37-64, Feb. 2006, doi:10.1016/j.pmcj.2005.07.003

[10] IMS Learner Information Package Best Practice & Implementation Guide, 2001, available at http://www.imsglobal.org/profiles/lipbest01.html, consulted on Dec. 2010.

[11] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen, and T. Scavo, "Security Assertion Markup Language (SAML) V2.0 Technical Overview," OASIS Committee Draft, March 2008, doi:sstc-saml-tech-overview-2.0-cd-02.

[12] T. Ramandalahy, P. Vidal, and J. Broisin, "Opening Learner Profiles across Heterogeneous Applications," in IEEE International Conference on Advanced Learning Technologies (ICALT 2009), July 2009, pp. 504-508, doi:10.1109/ICALT.2009.153.

[13] Reusable eLearning Object Authoring & Delivery (Reload), 2008, available at http://www.reload.ac.uk/, consulted on Dec. 2010.

[14] Systems Management: Common Information Model (1998). Open Group Technical Standard C804, DMTF, ISBN: 1-85912-255-8, 1998.

[15] The OpenID Foundation, "OpenID Authentication 2.0 – Final," Dec 2007, available at http://openid.net/specs/openid-authentication-2_0.html, consulted on Dec. 2010.

[16] Web Based Enterprise Management (WBEM), 1999, available at http://www.dmtf.org/standards/wbem/, consulted on Dec. 2010.