Chapter 2

.LRN: E-LEARNING INSIDE AND OUTSIDE THE CLASSROOM

Supporting Collaborative Learning Communities using a Web Application Toolkit

Carl Robert Blesius[1], Pablo Moreno-Ger[1], Gustaf Neumann[2], Emmanuelle Raffenne[3], Jesús González Boticario[3], Carlos Delgado Kloos[4]
[1]Harvard Medical School - Massachusetts General Hospital, Laboratory of Computer Science, 50 Staniford St. MA 02114 (USA); [2]Vienna University of Economics and Business Administration (Austria); [3]Universidad Nacional de Educación a Distancia-UNED (Spain); [4]Universidad Carlos III de Madrid (Spain)

Abstract: .LRN is an Open source Web portal and Web application toolkit designed to support both large and small communities of practice and learning inside and outside of educational institutions and the enterprise. .LRN has the features of a complete Course Management System, but is focused on supporting collaborative online learning communities that often go beyond the typical institutional or course based setting. This chapter describes the main features of .LRN, gives an overview of its modular and adaptable technical architecture, includes case studies from projects the authors are directly involved with (which help highlight the benefits of the platforms open source nature and its high performance capabilities), and concludes by giving a quick summary of the future direction of .LRN and the not-for-profit corporation that has formed around the software.

Key words: Collaborative Learning, .LRN, OpenACS, LMS, CMS, Web Portals, Software Consortia, Open Source, E-Learning.

1. WHAT IS .LRN?

.LRN is a fully internationalized open source portal and application framework built to support online collaborative learning communities and blended learning environments. .LRN is based on the tenet that learning is a

B. Fernández-Manjón et al. (eds.), Computers and Education: E-learning, From Theory to Practice, 13–25.
© 2007 *Springer.*

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

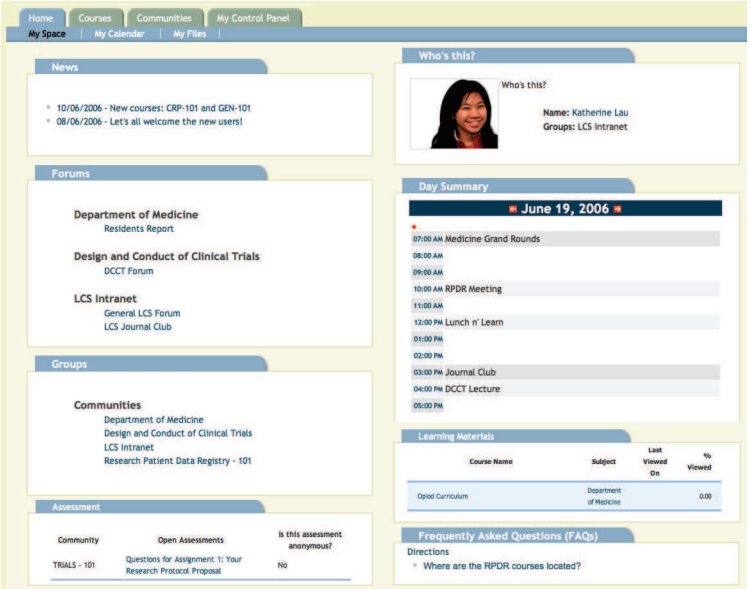


Figure 2-1. A screenshot of the .LRN personal portal page. It aggregates information from community and class portal pages. End users can customize this page as they please

Au: Please provide citation for Fig. 2.1 in text.

social experience (Wegerif 1998), that effective learning usually takes place in the context of communities (Alavi 1994; Wenger 1998), and that administration of these communities should be distributed. Each learning community has its own stakeholders and needs to define its own unique set of interactions, so the software is designed to be flexible and allow delegation of administrative roles as close to the learners as possible. While some e-learning environments are built around a course catalog (course management systems, CMS) and other are built around a content management system (learning content management systems, LCMS) .LRN focuses on online communities (learning community system, LCS), with course management and content management applications as an added value. This contrasts with the more content-centric approaches of most of the key systems in the e-learning arena, like the commercial WebCT (Goldberg and Salari 1997) or the Sakai (Farmer and Dolphin 2005) opensource project and taking the community oriented approach of Moodle (Dougiamas and Taylor 2003) to the extreme.

From an e-learning perspective, some interesting samples of .LRN applications are the Learning Object Repository System (LORS) that supports the IMS Content Packaging Specification (allowing packaging, organization, and import/export of learning objects), the Assessment tool (tests, quizzes, surveys, evaluations, etc.) that supports the IMS QTI import

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

Table 2-1. A rough order of magnitude estimation of the .LRN development cost based on lines of source code

Total Physical Source Lines of Code (SLOC)	487,182
Development Effort Estimate, Person-Years (Person-Months)❶	132.77 (1,593.26)
Schedule Estimate, Years (Months) ❷	3.43 (41.19)
Estimated Average Number of Developers (Effort/Schedule)	38.68
Total Estimated Cost to Develop (average salary = $56,286/year, overhead = 2.40).	**$17,935,662**
❶Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05)	
❷Basic COCOMO model, Months = 2.5 * (person-months**0.38)	
Data generated using David A. Wheeler's 'SLOCCount'	

and export, Gradebook, Homework (an assignment drop box for course facilitators), User Tracking (allows administrators to track what users have or have not seen), the ePortfolio package (allows individuals to keep track of their personal learning progress and share their interests and accomplishments with others) and, optionally, E-Commerce support (e.g. credit card based course registration).

.LRN is about learning communities and these are supported by applications such as the Forums and News applications (with email and RSS subscription options), a WYSIWYG Wiki system (which can be used in the context of the .LRN Learning Object Repository or as a simple content management system), Photo Album, group and individual calendaring (with Outlook synchronization), Weblogger, and Web-based group and personal file storage with WebDAV support.

The first version of what would become .LRN was created by ArsDigita Corporation (which has been sold to RedHat) for the Sloan School of Management (MIT's business school) and was based on software ArsDigita built for corporate knowledge management, intranets, and dynamic database backed websites. The system that was created for Sloan was called the *"ArsDigita Community and Educational Solution (ACES)"* (Meeks and Mangel 2000) and was released under the GNU Public License. The system was built to run on Oracle, but in 2001 a group of volunteers started the work required to support the PostgreSQL open source database.

Simplistic development cost calculations (Table 2-1) give a very rough estimation of the amount of development (~133 person years) and investment (~$18,000,000) that has gone into the software.

2. TECHNICAL BACKGROUND

.LRN is built using OpenACS (Calvo and Peterson 2002; Hernández 2005), a mature enterprise toolkit for building scalable web applications. OpenACS follows a multi-tier software design pattern (Web server,

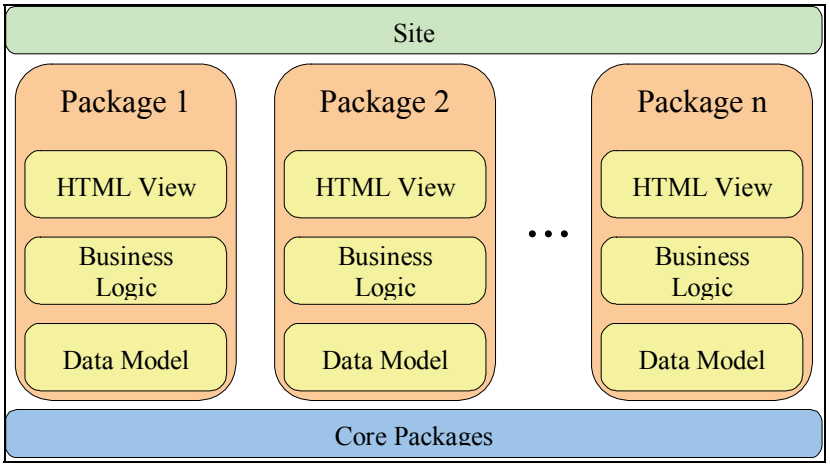
Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

application server, and database) and depends on AOLserver (Reuven 2002) (an application/Web server), which was built from the ground up as a tool for building highly scalable database-backed web applications.

2.1 General architecture

Based on almost 10 years of experience, the .LRN platform has evolved into a modular web application, with the goal of providing a sustainable and rich architecture, which promotes extensibility and adaptability without sacrificing maintainability. This is achieved with an organization in which functionality is divided into *packages* that can be designed, maintained, and distributed independently.

A default installation of the platform includes the core packages that provide the infrastructure of the system and the basic set of .LRN packages necessary to support a typical E-Learning scenario, however there are dozens of official packages that can be added to the default installation. In addition, a mature package development process allows numerous non-official packages to be created by third parties. This allows institutions to create their own additions to the basic OpenACS/.LRN installation and encourages them to share their creations. This is similar to the Building Blocks program around Blackboard's open architecture (Blackboard Inc 2006), although the distribution of the packages as open source applications is more common in the OpenACS/.LRN community.



Au: please provide citation for Fig. 2:2. in text.

Figure 2-2. General architecture of the OpenACS/.LRN platform. The "site" layer is a common entry point for all the packages, where the look and feel of the site is defined.

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

Generally packages are designed following a Model-View-Controller pattern that separates the view (HTML syntax), the business logic (Tcl scripts), and the data model. Typically the resulting packages are web-based software applications that take advantage of base functionality provided by core packages and can be distributed independently.

2.2 Separation of the View and Business Logic: The Templating system

Decoupling the view from the functionality itself provides a number of advantages in the development of web applications, most of them derived from the fact that neither web programming nor HTML design are trivial tasks and that both require a different set of skills. In fact, the first benefit obtained from this separation is the definition of two separate roles: The web *programmer* and the web *designer*.

The .LRN platform provides such a separation by means of the Templating System. Every request that the server receives is handled in two steps. First, a Tcl script (Ousterhout 1994) is executed. This contains the business required to attend to the request, like checking for permissions, accessing a database, modifying some data or preparing some data for display. The actual rendering of the page is postponed until the second step, the execution of an ADP page (AOLserver Dynamic Page), which is a simple document containing HTML markup and special tags. The key idea is that creating an ADP page does not require special programming skills (other than HTML).

2.3 Implementation of the Business Logic: The Application Server and the Web Application Programming Interface

The evolution of the .LRN platform and its backing software (namely, OpenACS and AOLserver) provides a number of programming APIs that facilitate the implementation of the business logic layer with different level of abstraction. At the core of the system, AOLserver provides a highly efficient multithreaded back-end, which is based on resource pooling and a caching mechanism, with a throughput capable of sustaining sites such as www.aol.com and www.mapquest.com. AOLserver provides a fully featured API for application programmers that can be called using Tcl.

However, it is not common to have to deal with this low-level API, as .LRN provides a rich API that facilitates development with high-level

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

constructs for database access and management, creation of data structures, and flow management. The power of this API is an important factor in .LRN platform development, because it simplifies the development and maintenance of packages by third parties.

2.4 The Data Model and the Object System

The .LRN architecture is enhanced by an abstraction in the underlying data model: the Object System. Initially, all the data objects defined by the different packages are stored in their respective package-specific data model which contains the tables and views necessary to properly store and manage those objects. But the .LRN implementation suggests the registration of these objects in a centralized listing. This centralized system stores references to all the objects and keeps track of the packages responsible for their management.

One of the most obvious advantages of this approach is exemplified by the Permissions System. The management of a healthy learning community often requires moderation and permission based contribution that may require a complex and fine-grained authorization mechanism. Instead of re-writing code to support varied and complex permissions a developer can simply register the objects needed in their applications (e.g. a blog post) in the centralized repository. The .LRN Permissions System will then detect any calls to that content (regardless of the package) and act consequently. This approach also facilitates the integration with other services or applications offered by the platform (e.g. adding support for comments to the blog post by integrating with the comments package), versioning (a core feature), and categories (another package/application).

Another benefit of registering all objects in an Object System is the ability to display content objects in multiple contexts. An example of this can be found in the .LRN Learning Object Repository System (LORS) designed for the maintenance of content in the form of Learning Objects (Koper 2003) and compatible with the specifications proposed by IMS (IMS Global Consortium 2004). LORS takes advantage of both Wiki Package content objects and .LRN IMS QTI (IMS Global Consortium 2004) compatible Assessment Package objects to allow end users to create sequenced learning modules (e.g. a self test made up of sequenced content and questions). As long as all objects are properly registered in the central repository various combinations of these objects can be used in different contexts with relative ease and limited development overhead.

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

3. SUCCESS STORIES AND SELECTED USE CASES

The high performance implementation and the open source nature of the platform has been a key factor in the adoption of .LRN by a number of institutions in different countries. These are some examples.

3.1 E-Lane

E-LANE is one of the projects funded by the European Commission under the @LIS Programme (Alliance for the Information Society 2006). @LIS is a European Commission project aimed at reinforcing the partnership between the European Union and Latin America in the field of the Information Society. E-LANE (E-LANE 2006) is one of the projects focused on e-Learning. Its objective is the deployment of courses for digital literacy and life-long learning in partner countries in Latin America, namely Mexico, Guatemala, Colombia, Chile, and Brazil.

For the selection of the e-learning platform, it became clear from the very beginning that it had to be one based on an open source license. With .LRN, E-LANE found a solid foundation on which to build. In the context of the project, the platform was extended to support detailed learner tracking and e-learning specifications, such as IMS-QTI (IMS Global Consortium 2005) as mentioned previously and IMS-LD (IMS Global Consortium 2005).

Pilots have been rolled out with more than 1000 students, in scenarios such as basic and graduate courses in Guatemala, digital literacy courses in Mexico (in collaboration with the local government), courses in schools in Chile, and basic courses for indigenous people in their own languages in Colombia.

3.2 Vienna University of Economics and Business Administration: Learn@WU

The Vienna University of Economics and Business Administration (Wirtschaftsuniversität Wien, abbreviated WU) is one of the largest business universities worldwide, with more than 22,000 students and ~2,000 courses offered each semester. The Austrian university system offers unlimited access to all students who pay a tuition fee of 350 €. This admission policy leads to high enrollment. Up to 4,000 freshmen enroll in a business-related study program per year.

High interest in the programs caused severe resource problems at the university. Therefore, in 2001 the university started Learn@WU (Alberer,

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

Alberer et al. 2003) as a content project with an initial 2 year budget of 3,4 Mio € to provide full coverage of the first-year courses and to provide an E-Learning alternative to classical university style lectures.

Some requirements of the university did not fit well the standard feature portfolio of other course management systems evaluated:

- A student should be able to visualize the topics that have been already covered within a "concept space map", and see how his or her scores compare with other students.
- The system must be able to represent a very detailed organizational structure for permission management. Since up to 50 parallel classes are offered per semester per course, multiple institutes host the same classes, and external lecturers teach many courses. The requirements for the rights management system were quite complex.
- The system should be integrated with the existing Kerberos authentication databases.
- More than 40,000 exams are evaluated per year for the first year of study only. The system should provide support for the integration of a mark reader and exam feedback system for students.

None of the evaluated environments were able to address these needs, nor was it feasible to modify them appropriately. The decision to use the framework turned out to be very effective, and the university used it as the base on which to develop a highly tailored system according to these requirements, continuously working with the collaboration and support of the .LRN community.

Learn@WU has been highly accepted by the students and faculties, with more than 30,000 learning resources online available to students and more than 25,000 registered users. It delivers up to 35 GB of data per day and handles up to 4.3 Mio requests (hits) per day. Students solve up to 350,000 interactive exercises per day during preparations for exams. Even at peak times, when more than 1,200 users are concurrently active, the response-time of the system (an IBM power5+ server running 64-bit Linux) is below 0.4 seconds. At the university the system is regarded as a very successful project, having halved the number of beginner classes. Most of all, it helped the university provide a high quality learning experience with limited personnel and room resources. "Without Learn@WU, the operations of our university would not have been possible", says Christoph Badelt, president of WU. Triggered by Learn@WU, E-Learning became a strategic goal of the university. As of summer 2006, there are 25,474 .LRN users, 405 open courses in 135 subjects, and 820 communities.

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

3.3 National University for Distance Education in Spain

The National University for Distance Education in Spain (UNED) is the biggest governmental Spanish university for distance teaching education with national and international coverage. From its beginnings in 1972, it has proved its quality and solidity both as an educational and as a research institution. As one of the ten largest Universities in the world by number of students, UNED is providing IT services for over 180.000 students, 1500 faculty staff, 5000 tutors, 1,200 administrative personnel, and 120 study centers. UNED has more than five years of experience in delivering online courses for over 140,000 students per year, supporting a task-based learning approach, where students no longer deal with the question-and-answer model and loosely coupled contents, and they choose .LRN for its robust architecture, scalability, and modularity.

The flexibility and open source nature of .LRN allowed UNED to create a customized solution, called aLF (Boticario, Gaudioso et al. 2001), which provides a wide set of tools for supporting courses and collaboration communities of varied nature (departments, faculties, administrative units, research projects, user groups, study communities, etc.). The system has evolved over the last four years to integrate E-Learning and collaboration utilities with the rest of the IT services provided at UNED. As of summer 2006, there are 42,000 .LRN users, 130 courses, and 160 communities.

3.4 Partners Healthcare Systems (PHS) and Massachusetts General Hospital (MGH)

Partners Healthcare Systems is a not-for-profit corporation with 9 large institutional members and 35,000+ full time employees. It is the major teaching affiliate of Harvard Medical School and is fully dedicated to enhancing patient care, teaching, research, and leading the world in integrated healthcare.

MGH is a member of PHS and is the largest hospital-based research program in the United States, with an annual research budget of more than $450 million. It is the oldest and largest teaching hospital of Harvard Medical School and nearly all of the hospital's active staff physicians are on Harvard Medical School faculty. At MGH ~1.5 million patients are seen each year and teaching and just-in-time knowledge access is recognized as a key part of providing the best possible care.

Within the Harvard system there are multiple in-house E-Learnig solutions available in addition to purchased commercial systems. MGH was interested in a system that would help facilitate clinical research by connecting staff, support online courses for both regulatory requirements and

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

continuing education, and help with the administration and reporting needs of educational units. .LRN was proposed in response to an internal request. After evaluation of .LRN, existing local solutions, and proposals by closed source commercial providers, .LRN was selected. Major factors in the choice were: the modular open source architecture, which allowed MGH to create custom packages, the high speed and low cost of required additions, and available commercial support.

MGH added a number of applications for local use, tying them into the preexisting .LRN data model. These modules included administrative tools to facilitate course registration workflow, attendance tracking, reporting, PDF certificate generation, and improvements to the assessment package. The system ties into the local MS Active Directory user management system (using LDAP) giving all the MGH staff easy system access. MGH also implemented a data collection infrastructure and reporting mechanisms for course registration so that people can be targeted by administration with tailored information and offerings.

The initial pilot/development phase (initially circa 1500 users in the context of clinical research and regulatory training) was successful and the second pilot has begun. This portion of the work includes a focus on just-in-time access to clinically relevant resources (E-Learning in the context of patient care), clinical learning modules, ePortfolios, and integration of assessment with collaboratively created learning content for physicians in training. This allows the residency program to test key knowledge, create a virtual space for exchange, and focus on covering any identified deficiencies in medical training on an individual basis.

4. GOVERNANCE AND FUTURE DIRECTION

4.1 The .LRN Consortium

The .LRN software is backed by the .LRN Consortium, a non-profit corporation founded by users of the software, which is committed to advancing innovation in educational technology through open source principles. Consortium member institutions work together to support each other's deployments and to accelerate and expand the adoption and development of .LRN. The Consortium ensures software quality by certifying components as .LRN-compliant, coordinates software development plans, and maintains ties with OpenACS.

The .LRN Board of Directors, representing Consortium members, sets broad strategy and has ultimate oversight responsibilities for the project, while the .LRN Leadership Team is a group of .LRN experts who are

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

responsible for the evolution and management of the .LRN project. The .LRN Leadership Team decides what goes into releases and works with the OpenACS technical release management to help manage implementation, testing, and documenting new features and bug fixes.

4.2 Future Direction

Currently that OpenACS/.LRN repository contains more than 200 different packages. While this is a great achievement, there is significant organizational and technical overhead involved in maintaining these packages. The code base is developed by more than 100 developers in total, who provide more than 2,000 contributions per quarter. This is substantially more than the average source forge project, which has about 600 contributions per quarter (Koch 2004). From the code management point of view it is desirable to reduce the number of packages on the one hand and to increase the flexibility of the packages on the other hand.

The OpenACS community has voted to base the forthcoming versions of OpenACS on the object oriented scripting language XoTcl (extended Object Tcl), see (Neumann and Zdun 2000). The development of the XoWiki system (Neumann) is a case study to explore some of the functionality in the context of .LRN and OpenACS. XoWiki provides functionality of a wiki-system and of a content management system (revisioning, inclusions, categorization), while providing a range of "Web 2.0" functionality (Weblog, RSS, AJAX, COMET) through a single package.

Another short term goal that has been earmarked by .LRN Leadership Team, (partially funded through a government grant) is to fulfill level 1 and level 2 accessibility success criteria (as defined by the W3C Web Accessibility Initiative) (W3C 2006) over the next 2 release cycles. .LRN is committed to helping provide users with special needs access to E-Learning services and is positioned to achieve this.

.LRN provides a highly flexible set of tools that can be customized for individual needs and scaled upward as local usage dictates. .LRN is actively under development, both formally and informally guided by its users through an inclusive governing process, and a flexible foundation to cultivate innovative learning environments.

ACKNOWLEDGEMENTS

Special thanks go out to the <e-UCM> Research Group (www.e-ucm.es) for their support, Katherine Lau from the Lab of Computer Science in Boston, the National Library of Medicine for supporting this work through a Medical

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

Informatics Training Grant (T15-LM-007092), and the Regional Government of Madrid by supporting this work through the FPI Scholarship Program (4155/2005).

REFERENCES

Alavi, M. (1994). "Computer-Mediated Collaborative Learning: An Empirical Evaluation." Management Information Systems Quarterly 18(2): 150-174.

Alberer, G., P. Alberer, et al. (2003). The Learn@WU Learning Environment. 6th International Conference on Business Informatics, Dresden, Germany.

Blackboard Inc. (2006). "Blackboard Building Blocks." Retrieved June 14th, 2006, from http://www.blackboard.com/extend/b2/.

Boticario, J. G., E. Gaudioso, et al. (2001). Towards personalised learning communities on the web. First European Conference on Computer-Supported Collaborative Learning, Maastricht.

Calvo, R. A. and D. Peterson (2002). The OACS web application framework. Ausweb 2002, Australia.

Dougiamas, M. and P. Taylor (2003). Moodle: Using Learning Communities to Create an Open Source Course Management System. World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003, Honolulu, Hawaii, USA, AACE.

E-LANE. (2006). "The E-LANE Project." Retrieved June 14th, 2006, from http:// www.e-lane.org.

Farmer, J. and I. Dolphin (2005). Sakai: eLearning and More. 11th European Univeristy Information Systems (EUNIS 2005), Manchester, UK.

Goldberg, M. W. and S. Salari (1997). An Update on WebCT (World-Wide-Web Course Tools) - a Tool for the Creation of Sophisticated Web-Based Learning Environments. NAUWeb '97 - Current Practices in Web-Based Course Development, Flagstaff, Arizona (United States).

Hernández, R. (2005). OpenACS: robust web development framework. Tcl/Tk 2005 Conference, Portland, Oregon.

IMS Global Consortium. (2004). "IMS Content Packaging Specification." Retrieved June 14th, 2006, from http://www.imsproject.org/content/packaging/.

IMS Global Consortium. (2004). "IMS Content Packaging Specification." Retrieved June 14th, 2005, from http://www.imsproject.org/content/packaging/.

IMS Global Consortium. (2005). "IMS Learning Design Specification." Retrieved June 14th, 2006, from http://www.imsproject.org/learningdesign/index.html.

IMS Global Consortium. (2005). "IMS Question & Test Interoperablity Specification." Retrieved June 14th, 2006, from http://www.imsglobal.org/question/index.html.

Koch, S. (2004). "Profiling an Open Source Project Ecology and Its Programmers." Electronic Markets 10(2): 77-88.

Koper, R. (2003). Combining re-usable learning resources and services to pedagogical purposeful units of learning. Reusing Online Resources: A Sustainable Approach to eLearning. A. Littlejohn. London, Kogan Page: 46-59.

Meeks, C. and R. Mangel (2000). "The Arsdigita Community Education Solution." Arsdigita Community Journal.

Neumann, G. (2006). "XoWiki." Retrieved June 14th, 2006, from http://media.wu-wien.ac.at/download/xowiki-doc/index.html.

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information

Neumann, G. and U. Zdun (2000). XOTcl, an Object-Oriented Scripting Language. Tcl2k: The 7th USENIX Tcl/Tk Conference, Austin, Texas, USA.

Ousterhout, J. K. (1994). Tcl and the Tk Toolkit. Reading, MA, Addison-Wesley

Reuven, M. L. (2002). "At the forge: introducing AOLserver." ACM Linux Journal 2002(101): 12.

W3C. (2006). "Web Accessibility Initiative." Retrieved June 14th, 2006, from http://www.w3.org/WAI/.

Wegerif, R. (1998). "The Social Dimension of Asynchronous Learning Networks." Journal of Asynchronous Learning Networks 2(1): 34-49.

Wenger, E. (1998). Communities of Practice: Learning, Meaning, and Identity. New York, Cambridge University Press.

Draft version: See http://www.e-ucm.es/publications/articles.html for updated citation information