E-learning for the Web 2.0 Generation

Introduction

Primarily, we identify the main characteristics of the coming Web 2.0, and, concurrently, we will try to judge how legacy e-learning platforms fulfill the current needs. Consequently, we will present our vision how e-learning should be shaped (re-thought) to fit into the coming Web 2.0 era.

The Ideas of Web 2.0

Tim O'Reilly in [tim] outlined the core characteristics of a company (or service) that can wear the label "Web 2.0":

- Services, not packaged software, with cost-effective scalability
- Control over unique, hard-to-recreate data sources that get richer as more people use them
- Trusting users as co-developers
- Harnessing collective intelligence
- Leveraging the long tail through customer self-service
- Software above the level of a single device
- Lightweight user interfaces, development models, AND business models

Services, not Systems

Thus, nobody relies on a single, stable, system but rather on services he/she needs – wherever they are and whoever provides them.

Syndication, not Stickiness

Web Services? Yes, but not of any kind!

How Legacy LMSs Meet Current E-learning Needs?

Primarily, we will show that present Learning Management Systems (LMS) – even the open-source ones – belong predominantly to the Web 1.0 Generation – and, what is more important, do not meet current e-learning needs any more.

Integration in the World of Internet

LMS systems do not exist in an isolated space, in their own universe. No matter how good content they provide, how sophisticated tools they offer, the users at both sites – teachers as providers, students as consumers – live and work in much richer world They use resources out of the LMS, generally accessible materials such as glossaries, Wikipedia pages, or just external web sites. They may need to discuss things in larger, open communities via chat and not to be limited to the closed world of one LMS. They share documents not just for learning, but also in their personal life. They wish to store and publish their photos…

Sure, LMS systems use Internet as the communication platform, they (at least to some extent) run on open platforms such as Linux, Apache and/or Java and are developed in a collaborative manner, but…

Current LMS – Products, not Services

Even the open-source development results in a *product* that needs to be installed, maintained, patched, updated, upgraded – not in a service that is just used... Too liberal policy in contributing to an open-source project leads often to chaotic incorporation of many features that may be unstable, not mature enough or not well documented. Maintenance of such system might become a nightmare – especially for smaller educational institutions without a strong infrastructure and experienced technical staff.

The management and supporting staff is then more concentrated to solving technical and administrative issues than to providing good *services* to customers – students.

Authoring Too Centralized

The are centralized – they resemble more Britannica Online than Wikipedia. On the other hand, current experiences in e-learning clearly show that collaborative authoring systems such as Wikipedia offer better learning content than any closed team of authors – no matter how large and experienced the authoring team is. It is especially true in areas of common interest, in fields that are taught and studied practically the same way all over the world – such as the computer science, or science in general.

The liberal, collaborative authoring produces high-quality content much faster and the content is more easily accessible – just by pointing to the right wiki page...

Publishing Not Dynamic Enough

E-learning is not about creating and publishing stable content in one step. It is about continuous improvement, modifications and quick publishing. With legacy LMS or CMS, publishing a resource often means creating it in an external authoring tool, uploading it into the LMS/CMS and associating necessary metadata with it. Any change in the learning material is cumbersome – it requires non-creative, repetitive work. As a consequence, the teacher often resigns and leaves the "stable" material untouched, including errors. Even publishing to personal web sites is not optimal. It is still too static, and requires specific knowledge of HTML and associated technologies.

Is Full Content Management Really Needed?

Content Management aims at providing tools for collaborative authoring, publishing, organizing, maintaining and delivering digital content.

Fully-featured business-oriented CMS systems offer additional functionality, such as in *Transactional content management* (T-CMS) for managing e-commerce transactions, in *Integrated content management systems* (I-CMS) that assist in managing enterprise documents and content [wikipedia-cms].

Complex solutions of CM represented by *Enterprise content management systems* (E-CMS) with functionality ranging from document-flow management to business process management are both expensive and difficult to configure and maintain.

Their functionality is not needed in the e-learning area. The same applies for the content organization. The legacy CMS tend to organize documents into hierarchies, pre-designed by a "guru" who must have known the structure in advance. Of course, the basic concepts of the

studied discipline, its outline and terminology, must be set up as a logically sound, organized structure. However, once a course that uses the material starts, the learning community tends to semantically self-re-organize the stuff by tagging it. This is called “folksonomy”, in opposite to (predefined) taxonomy.

From the technical point of view, some of the CMS systems even require a client software or plug-ins into the authoring tools.

User Management – Again, What We Need?

E-learning

Web 2.0 Approach to E-learning

The Services Required

We illustrate the requirements on a case study on e-learning targeted to development of web-based systems in Java EE environment.

The core idea is to let the users – both teachers and students – do their most of the job as ever before, not to force them to simply throw away tools they like and use.

For content creation, storage and delivery

This is probably the most important component in the stack of the e-learning infrastructure.

TODO Mediawiki, Docbook, VCS – Subversion

For timeline-related e-learning content

Legacy LMS usually offer some kind of timeline-oriented tools, such as lecture schedule or deadlines for assignments. Time criteria also generally play role in accessing any kind of resources, including learning content, assignment description, assignment delivery systems, self-testing tools. Access to any resource may be also time-limited.

TODO blogs, RSS

For searching

TODO Google search, Google Scholar

For content syndication

TODO RSS, ROME, Atom

For compatibility and reusability

However, the e-learning content creation and delivery should comply with established standards, especially for content metadata, such as AICC or SCORM. This is not just for backward compatibility and for collaboration with legacy systems. So, we should provide proxy services that can transform the required Web 2.0 content into the e-learning standard form. It is not, in principle, difficult, as there are already tools for creating SCORM-compliant content packages from wikis, blogs or web pages.

TODO: scormpackager, eXe

For identity management

TODO simple user management built as REST service

For communication

TODO Web-based chat, Jabber, archives, on-line discussion forums w. archives, Gmail

For collaboration

TODO Bugzilla, issues tracking tools, versioning/source code management tools, file-sharing tools (BitTorrent)

For multimedia

TODO Flickr, BitTorrent

For adaptivity

As Adaptive XML Includer has a RESTful interface, it may serve as proxy for adapting any markup content (HTML, Docbook, DITA, xdoc, RSS) that can be accessed the REST way via a URI. Other types of content are not a priori excluded from adaptation. There are input filters for plain wiki pages, ordinary web pages, and text files.

TODO AXI

For mobile access

TODO partially: based on AXI content adaptation

For personalization

Providing personalized, person-centered views of any content is the Holy Grail of many current research activities. But again, many of them concentrate on building either centralized systems for maintaining user profiles, usually requiring quite complicated bootstrap-phase until enough information about the users are collected. They also typically need rich semantic background in the field of the personalized content.

However, one can find also “cheaper” solutions.

TODO: peer personalization w. LXP

Transition to Web 2.0

As we saw in the previous sections, the legacy LMS more or less provide the required functionality. However, for virtually each kind of this functionality, there are better, Web 2.0 services or tools readily available.

So, the direction for the future is to:

- Build systems based on services.
- Build them simple, but extensible.
- Integrate the services available, either directly or via proxies.
- Provide own services in specific areas, not covered by general Web 2.0 ones.
- Respect the REST architectural style of services wherever possible.

Shaping the Web 2.0 E-learning Platform

Finally, we will present a vision of a complete Web 2.0 e-learning platform, respecting the previously specified requirements.

References

[tim] O'Reilly, T.: What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. http://www.oreillynet.com/lpt/a/6228

[wikipedia-cms] Content management system. http://en.wikipedia.org/wiki/Content_management_system

[whatis-cms] Content management system. http://searchwebservices.techtarget.com/sDefinition/0,,sid26_gci508916,00.html

[bittorent] http://www.bittorrent.com/introduction.html

[squidoo-web20] Porter, J.: Introduction to Web 2.0. http://www.squidoo.com/introtoweb20