

# Towards a New Generation of Learning Management Systems

Hubert Vogten<sup>1</sup> and Rob Koper<sup>2</sup>

<sup>1</sup>*Faculty of Psychology and Educational Sciences, Open University of the Netherlands,  
Valkenburgerweg 177, Heerlen, The Netherlands*

<sup>2</sup>*Open University of the Netherlands, Heerlen, The Netherlands*

Keywords: LMS, Learning Network, Distance Education, Liferay, Portal.

Abstract: In this paper we argue that a centrally governed Learning Management System (LMS) still has ample legitimacy in an information society that is ever more adopting cloud computing services in daily life. We argued that control over services and produced data is essential from the perspective of an educational institute for reasons of accountability, quality control, legislation, privacy and reliability. However, the current generation learning management systems are primarily geared to provide 'additional' online learning. In 'real' online learning teachers and students almost never need to meet face-to-face. All instruction, tests, communication and collaboration is organised using internet and mobile technologies. We propose a paradigm shift for the next generation LMS, discarding the course as key concept in favour of the social learning network concept. We argue that a generic social collaborative portal platform is a good foundation for the development of this next generation LMS. We support our arguments by presenting a real world case and we conclude that we can reuse 80% of the standard code.

## 1 INTRODUCTION

So, what to do when you are seriously planning to provide 'real' online learning for students you will never meet or see? Are the traditional LMSs that are currently locally implemented as an add-on to campus-based education usable for these purposes or is it better to implement and use another solution?

The uptake of 'real' online learning has grown massively in the past years driven by the public attention and uptake of Massive Online Open Courses (MOOCs). The term 'real' in the previous sentence is contrasted to the term 'additional' online learning, i.e. additional to campus-based teaching and learning. In real online learning teachers and students almost never need to meet face-to-face. All instruction, tests, communication and collaboration is organised using internet and mobile technologies. In the last decade many universities and schools have implemented a dedicated Learning Management System (LMS) such as Blackboard or Moodle to be used in addition to the regular campus-based teaching and learning. Its use is mostly restricted to the sharing of PowerPoint slides, information about grades, examinations, classroom changes, providing access to online books, papers

and other resources. The real work and instruction is mostly done outside the LMS. Given that the main users of LMSs are using the LMS as an addition, it is likely that the requirements for the development of these platforms are mostly driven by this extended classroom paradigm and not by requirements stemming from 'real' online learning. In practice most universities that are using MOOCs use different (or adapted) platforms, like Udacity, Coursera or Futurelearn.

Using these MOOC platforms may be satisfactory from the learner's perspective; it is not necessary desirable from the educational institution's perspective, for whom the online platform is an essential, integral part of the core business. Such educational organisation is obliged to offer these services at an agreed level of quality and availability. Furthermore, educational institutions want to have access to all learning performance data so the learning and teaching can be improved by applying learning analytics technologies. Also legislation, internal quality control and privacy issues imply that an educational institute is responsible and accountable for these data. An educational institution simply cannot take accountability for services in the cloud that are not

under their control. Hence, an integrated, centrally governed infrastructure for online learning has still got its legitimacy. However, such an infrastructure requires a paradigm shift from a limited view on online course support towards a more inclusive view on 'real' online learning. This raises a question: what is the best technical foundation for implementing such paradigm shift. This question will be addressed in the next sections by discussing an implementation of such an infrastructure at the Open University of the Netherlands.

## 2 THE LMS BEYOND THE COURSE

In 1984 the Open University of the Netherlands (OUN) was established as an institute for open higher distance education for adult learners. Its major goals were (1) to offer adult learners a second chance to higher education and (2) to provide an alternative route to higher education in order to reduce the load on costly traditional, face to face education. The Open University started in 2010 with the development of the infrastructure for their next generation LMS, which is called OpenU.

Hermans, Kalz, & Koper (Hermans et al., 2013) distinct three types of online environments in which adult learners act:

- The LMS, built around the course concept and intended for formal instruction;
- The Personal Learning Environment (PLE), governed by the learner;
- Social network sites and learning networks (LN) for social and informal learning.

OpenU has the ambition to support all three online environments through the same infrastructure. Therefore, OpenU should be able to support various target groups in their formal and informal learning needs in a distance education setting. These target groups include master, bachelor and PhD students, but also professionals wanting to keep up with latest developments and trends in their area of expertise. But also researchers should be enabled to showcase and discuss the state of art in their topic of research, i.e. to increase the impact of their research. OpenU should allow learning networks to be established for all target audiences, but it should also be possible to cross these boundaries and bring students, professionals and researchers into contact with each other. Informed by a social constructivism view on learning, emphasizing that the development of knowledge and skills require intensive social

interactions (Schunk, 2012) OpenU should provide ample social tools. Finally users should be allowed to construct their own personal learning environment and the self-directed learner should be encouraged to explore formal and informal learning opportunities. Therefore a substantial part, about 10%, of all learning materials will be offered as Open Educational Resource (OER). Students and professionals should be able to receive credits for their participations in these MOOCs, which can be used in either their curriculum or for their professional development. From these use cases we derived a set of high level requirements for OpenU.

### *Social and Collaborative Requirements*

- The system should allow grouping and participation of users into communities. Each community should have a virtual presence on the web;
- It should be possible to set fine grained access rights to these communities, based on the role of a user in such a community;
- A user should be able to define an online identity;
- Various social tools should provide the social cohesion in the system and should allow and promote collaboration.

### *Content Management/Publishing Requirements*

- Users should be able to collaboratively create and manage digital content. This content can take various formats, such as web pages, Wiki pages, blogs and files;
- It should be possible to set detailed access rights to this content, controlling who can create, edit, delete and view the content;
- It should be possible to create web pages for the created content including possibilities to control layout and navigation.

### *Portal Requirements*

- Users should have a single access point for their learning. The system should provide an integrated and consistent user experience.
- It should be possible to combine functionality in a flexible manner to construct the different types of learning networks as each type of learning network has slightly different requirements.
- The system should be open allowing the seamless integration of various external services.

### *LMS Requirements*

- Learners should be able to keep track of their study progress while performing the activities of a learning design.
- It should be possible to upload, discuss and review assignments.

- The system must support the creation and delivery of self-assessments. It should be possible to incorporate these self-assessments into a course design.
- It should be possible to monitor study progress by learners and tutors alike, including self-assessment outcomes.
- Users should have their own portfolio and it should be possible to create showcases based of the data in the portfolio and traces of their activities within the system.
- It should be possible to implement various pedagogical approaches and course designs for the different target audiences.

### 3 THE LIFERAY PORTAL

When reviewing the global requirements one can argue that there are several types of systems that provide a good foundation to achieve the desired functionality. Content Management Systems (CMS), Social Networking Systems (SNS), Web Publishing Systems, Portals and traditional LMSs all provide parts of the required functionality. However, no single of these systems will meet all requirements.

Given that no system meets all of our requirements, the following question arises; what is the best foundation for developing the next generation LMS assuming that it makes no sense to start from scratch? One obvious strategy would be starting with a traditional LMS, such as Moodle or Blackboard, as starting point. However, the focus on the extended classroom and course paradigm probably would hinder a swift integration of social and informal learning objectives in OpenU. An alternative strategy would be using a more generic system that is free from any pedagogical paradigms. We argue that the latter approach is the better, more flexible approach towards the next generation LMS. In the next section we will discuss the OpenU case and show how we have configured and extended the Liferay portal environment (Liferay Portal, 2013) to meet our requirements.

Liferay provides a major part of the required functionality by integrating several relevant subsystems into a single framework. Some characteristics of Liferay are:

- Liferay is a full JSR-286 (Hepper, 2005) portal. It will allow the creation of a singular user experience through portlet technology.
- Liferay has an integrated CMS, complete with workflow control. It supports various content types.

- Liferay provides a social, collaborative environment via a range of social portlets.
- Liferay is based on an extendable open service architecture that allows bidirectional exchange of data through well-defined and standardized interfaces such as web service and JSON/REST.

However, Liferay is lacking specific LMS functionality which therefore needs to be added. In the next three sections we will describe how we used and configured Liferay to meet our requirements and we will describe what components we added.

### 3.1 Setup of Liferay as LMS

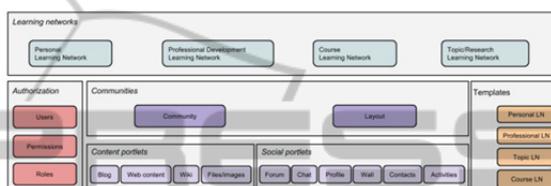


Figure 1: Basic Liferay functionality.

Figure 1 depicts a high level, layered functional decomposition of the OpenU LMS, based on an out of the box Liferay portal. We will explain how we decided to map the main Liferay concepts to build our learning networks. The top level represents the four learning network types that OpenU currently supports:

- The Personal Learning Network is owned and largely controlled by the user. In effect this is the OpenU implementation for a PLE. All users have their own personal learning network which they control;
- The Course Learning Network is intended for delivery of formal, designed courses for the bachelor and master curricula. Most members will be course students, but prospective students and professionals are allowed to participate as well, although they will not receive support by any Open University staff.
- The Professional Development Network focuses on latest trends and developments that are especially of interest to professionals desiring to keep up with latest developments in their profession. Although these networks are intended to address continuing professional development, regular university students can participate in them as well. Both professional and students can receive credits for their participation.
- Topic/research networks focus on the state of the art on main research topics. The intended audiences are researchers, students and

professionals alike.

A learning network is implemented via the Liferay ‘community’ concept. A Liferay community consists of a number of web pages and each of these web pages can contain portlets in a specific layout. A portlet occupies a part of the screen estate and its functionality can range from something very simple such as displaying a piece of web content, to a full fledged interactive group wall application. One can compare these portlets with Lego bricks. Each type of Lego brick provides a basic building block, but only the combination certain bricks result in a desired model. Similar, only after combining a particular set of portlets this will result in the desired learning network type.

Besides determining which portlets should be combined on a page, we also must define who is member of a learning network and what rights each user should have within the network. In Liferay this authorization is handled through a role based permission system. A standard role, for example, is the community member role. But it is possible to define as many roles as required. The actual permissions that can be set vary per portlet. Typical permissions are view and edit permissions.

With these standard Liferay concepts it is possible to construct a basic learning. However, configuring each learning network from scratch would be impractically at best. Liferay provides a template mechanism for automating this process. We defined a template for each type of learning network.

The bottom layer of Figure 1 is formed by the Liferay portlets, the equivalents of the Lego bricks. This includes content management portlets as well as social and collaborative portlets. With this approach and the standard portlets it is only possible to create very basic learning networks. Advanced learning networks will require additional LMS functionality that is not available out of the box.

### 3.2 Extending Liferay

Liferay provides an open API and SDK for developing new portlets which can be either build from scratch or can reuse services from existing portlets. We extended Liferay with a number of educational portlets that are required to turn Liferay into a LMS using both aforementioned portlet development strategies.

Figure 2 depicts the new high level functional architecture of Liferay including these extensions.

The following additional portlets have been developed:

- *Assignment portlet*: this portlet allows the sharing

and discussion/rating of assignments with tutors;

- *Assessment portlet*: this portlet allows the creation of assessments and provisions these assessments to learners;
- *Sequencing portlet*: this portlet builds on the Liferay Wiki and allows the construction of learning designs that are very similar to IMS-LD level A. The course author can create and sequence learning activities. On a role basis, the course author can determine who should perform which learning activities;
- *Showcase portlet*: allows users to share evidence of their learning progress via a showcase. The showcase portlet was initially developed as part of the EU lifelong learning programme project TRAILER (Brouns, Vogten, Janssen, & Finders, 2013);
- *Monitor portlet*: portlet that provides tutors with information about the progress of students in the learning network;
- *Groupwall portlet*: a portlet build on top of the Liferay forum resembling the Facebook wall feature, but owned by the network community.

With these additional portlets in place, we were capable of implementing all four learning network types. However, populating these networks with actual users is very labor intensive and therefore also error prone, especially when the number of users increases. Additional software is required to manage these subscriptions.

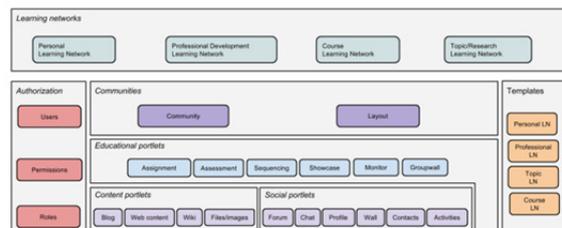


Figure 2: Extended Liferay functionality.

### 3.3 Subscription Management

Figure 3 depicts the functional architecture with an additional subscription management layer. Informed by the ‘separation of concerns’ design principle, this management layer hides all specific Liferay configuration details.

This is achieved through the introduction of a semantic neutral artefact called ‘product’.

The UML class diagram of Figure 4 represents this product artefact and its relationship with the Liferay concepts. The standard Liferay artefacts all have a grey background color. A product represents

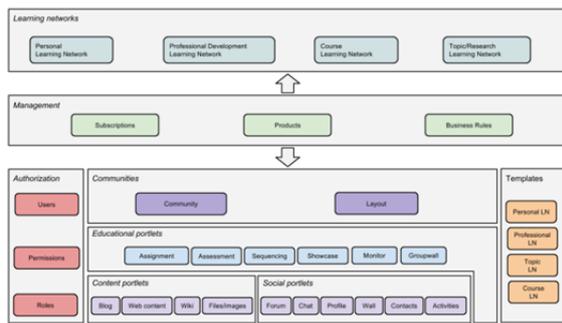


Figure 3: Subscription management layer.

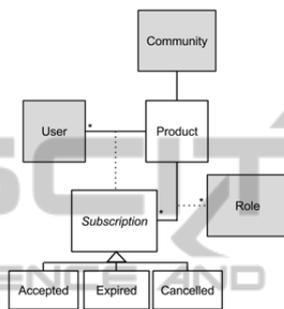


Figure 4: UML class diagram for product artefact.

a learning network via its association with a community. A product also defines which Liferay roles are associated with a subscription instances. A subscription instance has a state of either *accepted*, *expired* or *cancelled*. A user can have a subscription for a product, which is therefore either accepted, expired or cancelled.

Setting or changing a subscription will trigger a recalculation of the associated Liferay role assignments for the subscription user. This guarantees that a user always has the correct Liferay permissions in accordance with the product subscriptions. A product subscription can be set from various sources. For example, in the case of OpenU, a proprietary student administration system acts as source for all subscriptions of the bachelor and master students. Product subscriptions can also be set as a result of a purchase in web shop or simply via as a result of an open registration. The latter is typically when accessing the open educational resources. Regardless what the source of the subscription is, all role assignments will be automatically calculated without any further need for manual intervention and as a consequence the correct Liferay permissions are granted to the user.

Finally, we have implemented some registration business rules that allow the definition of products dependencies. With these business rules it is

possible to define flanking products for a source product, meaning that a user will be automatically subscribed to these flanking products whenever a user is subscribed for the source product.

## 4 IMPLEMENTATION

To get an impression about the efforts required to extend Liferay we performed some code metrics analysis. We have used the CLOC 1.6.0 (CLOC, 2014) for this purpose. Table 1 represents the result of running the code metrics on the sources of the standard Liferay 6.0.12 EE product.

Table 1: CLOC statistics for standard Liferay Portal.

Language	files	code
Java	17581	1881276
HTML	4190	409408
JSP	873	68249
XML	227	44102
CSS	39	5809
Javascript	117	34715
XSD	21	18250
SQL	32	5821
XSLT	5	169
<b>SUM</b>	<b>23085</b>	<b>2467799</b>

Table 2 has the code metrics for the extensions of the Liferay code that we have developed. This includes the educational portlets as well as the code for subscription management layer as described in the previous sections. It also includes some very specific code for the integration with the Open University infrastructure, such as the identity management services. However, these specific additions have only a minor impact on the statistics because they are relative small in size.

Although we didn't use all available functionality of the standard Liferay Portal, we used a very large portion of it. Therefore we may conclude that roughly 80% of the code needed to implement OpenU was realized with standard Liferay code and roughly 20% had to be developed from scratch.

The source code of all extensions is available as Open Source through SourceForge at <http://sourceforge.net/projects/openu/>

Table 2: CLOC statistics for the Liferay Portal extensions.

Language	files	code
Java	2291	374844
HTML	7	423
JSP	150	10298
JSF	295	24841
XML	306	18739
CSS	69	15541
Javascript	31	4909
XSD	1	230
SQL	23	1016
XSLT	1	77
<b>SUM</b>	<b>3174</b>	<b>450918</b>

## 5 CONCLUSION AND DISCUSSION

So, what have we learned? Most LMSs have dedicated code for teaching and learning, but this is built on a more generic functionality layer that can also be found in most portal and CMS software platforms. In our situation we found that only 20% of the code is specific to an LMS. Furthermore, the educational model of a distance teaching university like the OUN requires a specific set of teaching and learning modules that are hard to find in a standard LMS. At the pedagogical level there are many choices to make. For instance whether or not to implement informal and social learning as described in this paper. Many users of traditional LMSs need to adapt and configure the platform substantively in order to fit the specific local educational requirements or when delivering ‘real’ online education at substantive scale. In this effort they could be hindered by underlying restrictions in the models applied by the developers of the LMS. For instance, most LMSs embrace the course and extended classroom concepts in their core, which can be difficult to change towards more generic concept like learning networks.

The real issue in selecting and implementing an infrastructure for teaching and learning is to find a suitable platform that fits into (and connects to) the existing infrastructure, is secure, flexible, stable and

scalable, provides generic CMS, communication and collaboration tools out of the box and is extensible and adaptable. But of course, this is in our opinion the best way to proceed when one is serious in implementing ‘real’ online learning.

## REFERENCES

- Brouns, F., Vogten, H., Janssen, J., & Finders, A. (2013). E-portfolios in lifelong learning. In F. J. García-Peñalvo (Ed.), *Proceedings of the 2013 Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 535–540). Salamanca Spain: TEEM2013.
- CLOC. (2014). CLOC. *Website of CLOC*. Retrieved January 09, 2014, from <http://cloc.sourceforge.net/>
- Hepper, S. (2005). JSR 286: Portlet Specification 2.0. *Website Java community process*. Retrieved March 10, 2013, from <http://jcp.org/en/jsr/detail?id=286>.
- Hermans, H., Kalz, M., & Koper, R. (2013). Towards a learner-centred system for adult learning. ... *-Wide Information Systems*. Retrieved from <http://www.emeraldinsight.com/journals.htm?articleid=17098107&show=abstract>.
- Liferay Portal. (2013). *Website of Liferay*. Retrieved March 10, 2013, from <http://www.liferay.com/products/liferay-portal/overview>.
- Schunk, D. (2012). *Learning Theories: An Educational Perspective*. *Learning theories: An educational perspective*. (Sixth Edit.). Pearson Education Inc.