Experience with Open Source for e-Learning

Firoz Kaderali, Olaf Ehlert
FernUniversität Hagen / Germany
firoz.kaderali@fernuni-hagen.de, olaf.ehlert@fernuni-hagen.de

Abstract

A learning management system is an elementary requirement for teaching in a communication network using new media. Resulting claims for the introduction and long-term operation of an electronic platform are flexibility, adaptability, and scalability. Modular software architectures can satisfy these demands. Establishing good working modular standards is an error-prone task, which is best done in a community. CampusSource is an open source initiative, which makes available tools for teaching and learning with new media and guides the development of modular architectures.

1. Introduction

An electronic operation platform (commonly referred to as learning management system) is an elementary requirement for teaching in a communication network using new media. Many e-learning providers (e.g., universities, authors, and tutors) do not realise, that they need supplementary tools to handle their multimedia content, administrate students, support staff, and manage tutoring and communication. For the majority of institutions the use of a management platform is a strategic issue and will be critical for their current and future business (in terms of a potential service portfolio, ongoing operations, rationalization of work processes, and relating dependencies).

Planning, acquirement or development and deployment of such a platform is a difficult and time and effort consuming process. At present there are over 200 commercial and open source products to choose from and more than several hundred relevant criteria for the choice. Following a “standard” approach one has to identify ones own individual relevant criteria and their importance and then evaluate available products accordingly [1],[2]. Such a complex process always leads to a platform solution, which is suboptimal, since one never has a free choice of all required features, mostly caused by interdependencies. Moreover a learning management system has to cover a multitude of relevant areas with totally different requirements. In practice this turns out to be a serious restriction. Furthermore the operation of an electronic platform is a dynamic process, not all needed features even for one's own platform are known in advance (during initial stages, e.g., planning or development), in most cases the demand for certain features is detected during deployment and operation and additionally the demands and needed features will change over time.

Consequently major claims for an e-learning platform are flexibility, adaptability, and scalability. Current software engineering practice can provide the technical basics to implement applications fulfilling these requirements (e.g., modular component software). However the core problems are intrinsic to the market for e-learning tools and the associated development processes. Many e-learning providers develop their own environment and invent the wheel over and over again, wasting a lot of effort. On the other side commercial platforms have numerous features, many of which are usually not required for ones own teaching, and the functions considered indispensable are often not available. Since source code of commercial platforms is not available, one has low flexibility and little extensibility. In addition the interworking with existing IT-infrastructure and legacy systems is more difficult.

The proprietary character of commercial software determines the potential business models and restricts corporate identity. The actual features implemented in a particular platform are dependent on the business case intended by the platform providers (in most cases the developers). For example commercial providers tend to offer monolithic (closed) systems with a large choice of features and even integrate specific authoring tools in their platforms. On the other hand open source providers concentrate on the realisation of their specific business case and the resulting feature set, often missing important customer needs,

but in contrast they usually try to implement open standards to be compatible to other products.

2. Vision of future systems

As such, the future development of e-learning tools and platforms should be guided by a modular architecture based on open standards, which would enable a free and individual choice of the desired features. The ultimate vision being that e-learning providers can model their business case by some formal description, an enumeration of required features is automatically deduced, following the feature list the providers can compose pre-build and own developed software modules to produce an appropriate e-learning environment. This would be fully flexible and highly scaleable.
A modular architecture will even optimise operation and maintenance by means of feature adaptation of a platform, e.g., substitution of modules, if business processes change over time.

The vision as such is a long-term goal, it is primary intended as a guideline for research and development. Definition and fostering of such architecture is one of the core tasks of the open source initiative CampusSource. CampusSource proposes a co-operative approach based upon open source. The fundamental idea behind open source initiatives is the common use and mutual development of software. Open source software can be downloaded and used free of charge. The documentation, usually including installation guide, and the source code are freely available. The software can as such be modified, improved, adapted for different applications and generally enhanced - the only requirement being that a copy of the altered software has to be made available for common usage. Usually a community of users and developers emerges, which supports the further development of the software, its de facto standardization, and influences the direction of future development. The conditions of usage and modifications of the software are fixed in an appropriate license agreement. The best-known license today is the GNU General Public License (GNU GPL) [3], which allows a commercial utilization of the software and is also used for most parts of Linux.

3. Acquisition, development and deployment strategies

In the case an educational institution seriously intends to use new media for its teaching, it has the following four basic strategic directions to realise a platform:

- Own development
- Commercial product
- Open source solution
- ASP Application service provider solution.

These directions have intrinsic pros and cons, differences are characterised in the following parts.
A modular architecture can combine several of the different strengths of these approaches by choosing different implementations of modules accordingly.

Customisation

One important step regardless of the chosen strategy is customisation, it is absolutely necessary to reflect the demands of all participants (e.g., administration, teachers, students) and to represent the e-learning provider in the best possible way.
Major parts in the customisation process are:

- Adaptation to business case
- Adaptation to corporate identity
- Adaptation to individual work flow

Customisation is a complex process, mainly caused by the high number of requirements and their dependencies, therefore it is a time consuming step and will cost a major slice of the budget. Since customisation must be done in any case, the comparative high expenses are mostly independent of the implementation strategy, and the provider itself is deeply involved in this process, the reasoning is to recommend a modular approach, this will enable the additional free choice of functionalities.

Own Development

With a proprietary development one can construct a system, which satisfies ones needs precisely. It is possible to implement exactly the intended business case and to integrate existing and legacy infrastructure in the platform design. Since the

source code is available and was developed by oneself, one has a great flexibility and is independent of others – particularly of commercial suppliers and service providers. However this kind of solution is unreasonably expensive, one usually develops features that are already available elsewhere, this means reinventing the wheel and wasting a lot of effort and funding. Of course one also faces the usual problems encountered in software development: shortage of man power and limited know how of the own developers, early aging due to short innovation cycles, time delays, budget overdraws etc. Due to these reasons, this kind of solution is only appropriate for large companies and institutions with high financial and development resources or in the case when very special application requirements have to be fulfilled.

Commercial Solutions

At present commercial solutions are relatively inexpensive to begin with and usually offer extensive features. However ones own corporate identity is restricted and the commercial software basically determines the business model. The solutions offered normally require certain hardware and software infrastructures, so that the integration in ones own IT-infrastructure may turn out to be a problem. Since commercial providers generally do not make source codes available, one is seriously restricted in case extensions or modifications are necessary. Although numerous features are available, commercial providers have a focus on features that are used by many customers. In case one has special uncommon requirements, these are expensive to satisfy. Usually commercial providers bind their customers by the dependency, which is derived from their licensing, versioning and compatibility policies, which lead to a vendor lock – an expensive affair.
The commercial market is in a stage of refinement after which the costs of primary installations and the licenses both will rise. One also has to face the risk that the vendor one chooses might not survive the refinement process.

Open Source Solutions

Open Source solutions are the alternatives with the lowest costs, both in acquiring the software and in operating the system. Since all source codes are available, they are as flexible as a proprietary development as far as extensions and modifications are concerned. Usually open standards are used for the development, so that compatibility to other products is high and one is independent from any commercial provider. Particularly for university applications, many matching solutions are available. However a number of tools offered are not matured enough and neither tested nor appropriately documented. It is recommended to participate in the respective user and developer community, to get an accurate impression of the product considered. Since open source products for e-learning have become available quite recently, commercial support for them is frequently not available.

ASP Application Service Provider Solutions

Today a number of application service providers run commercial or open source learning management systems and offer all services required for teaching in the network to educational institutions. The services offered can be very diverse and range from common portal and management systems up to enhanced dedicated outsourcing. In case of an ASP solution, it is not required to have ones own hardware or service personnel for running the system and one can completely focus on preparing the teaching material and tutoring. With respect to the total cost of ownership, such solutions are very attractive, particularly when there is a shortage of own manpower. On the other hand one is heavily dependent on the service provider, has little flexibility and is restricted in the choice of ones own business model.

4. Modular Architectures

Contrary to the growing use of object-oriented and component-oriented modelling and development in the field of software engineering, the development process in the domain of e-learning is generally less modular and not well documented. Working and practiced standards are mostly missing. Therefore exchange of experience in this domain is highly needed and helpful for the convergence to modular architectures.

Important building blocks of a modular vision are:

- Modular architecture with a flexible composition mechanism (e.g., component based)
- Formal modelling of business work-flow and feature deduction
- Formal description of requirements and functionality (including interdependencies)

- Separation of layout and content (e.g., based on XML)
- Metadata standards for components and content

Achieving the goal of truly modular systems is limited by practical considerations. The technological environment causes one fundamental problem. At current, different major styles of development processes dominate the development of available and future systems: PHP based development, Java/EJB based development, and .NET based development. These development platforms are not compatible on a low technical level (for instance on the binary layer, library layer, or source layer). For usage in a common modular platform it is necessary to introduce an additional and more abstract layer, like web services.

Further problems lie in missing documentation and documentation standards for modules and features, respectively interfaces, implemented. This causes extra effort in the evaluation of usability of available code. Better formal description mechanisms with enhanced semantics are needed.

Another common problem for the integration of different software into one platform is the "versioning problem". After the integration of a module into one's own platform, usually the developers of the module will publish new versions with new or modified features. Often this will force a larger effort to reintegrate the module into the platform.

Design of a modular architecture is a complex process and calls for experienced developers. Usually only a minority within the greater community is capable of initiating a mature modular process. A good starting point is the common set of basic features that will occur in almost all platforms or applications:

- Administration of students
- Management of content
- Support of authors and tutors and
- Organization of the communication and tutoring.

The implementation of features in applications is driven by the strategic positioning of the developers and mostly by the demands of the application specific user community, because ideas and impulses for new features or solution of problems arise in the communities gathered around a specific application. This is the reason that today many more supplementary features are available in form of simple tools, plug-ins or even full platforms.
Based on the experience in CampusSource the resulting implemented features vary widely. In most cases the developers have no intention to put extra effort in the implementation of modular integration standards. An accompanying awareness campaign should sensitise the communities for possible synergies and added values by following a modular approach.

5. The CampusSource Offer

CampusSource is an open source initiative, which makes tools available for teaching and learning with new media to all interested parties on basis of the GNU GPL License. The starting point of the initiative was the development of various e-learning tools between 1996 and 1999 in different universities in Nordrhein-Westfalen. These were then adapted over the following two years so as to be adequate for open source usage. Since the opening of the open source pool of CampusSource [4] in April 2001, a growing number of tools from other German universities have become available. Today thirteen different tools including six complete electronic platforms are available in CampusSource and an active community supports their development. A number of small and medium companies offer support for these tools on a commercial basis. In the meantime software and documentation, originally in German, is also available in English. The initiative CampusSource is financially supported by the Ministry of Science and Research in Nordrhein-Westfalen.

The software tools offered in CampusSource are addressed to national and international users in educational institutions. All tools available were developed by renowned universities, which also have them in use on a long-term basis and were recommended by two experts, who also checked certain basic requirements, such as sufficient documentation including an installation guide, download features and compliance with open source requirements. However the tools were not tested at check-in, but the community validates and evaluates them.
Participating in CampusSource also gives the chance to take part on discussions on modular

standards, description mechanisms, and influencing current and future development.

Since the CampusSource pool was established in April 2001, more than 1500 users and developers of the software tools offered have registered at CampusSource. The network that has thus emerged is dedicated to the fundamental open source notions and has led to communities around the individual systems, which support their application, modify, enhance and update the software. Also discussion groups on subjects like future architecture of learning management environments, their standardization and compatibility etc. are emerging. The universities that originally developed the systems and a number of commercial small and medium sized enterprises offer support for the systems available under CampusSource. The range of available solutions covered varies from online information services and training over installation aids up to complete installation or outsourcing. For more information on the CampusSource initiative please refer to [4].

6. References

[1] Virtual Learning Community, http://www.virtual-learning.at/evalplattform.htm

[2] Online educational delivery applications: a web tool for comparative analysis, http://www.edutools.info/course/productinfo/index.jsp

[3] http://www.opensource.org/licenses/gpl-license.html

[4] http://www.campussource.de/org/