

Innovative Authoring Tools for Online-courses with Assignments

Integrating Heterogeneous Tools of e-Learning Platforms into Hybrid Application Solutions

Matthias Then¹, Benjamin Wallenborn¹, Birgit R. Ianniello² and Matthias Hemmje²

¹Zentrum fuer Medien und IT (ZMI), FernUniversitaet in Hagen, Universitaetsstr. 21, 58084 Hagen, Germany

²Lehrgebiet Multimedia und Internetanwendungen, FernUniversitaet in Hagen, Universitaetsstr. 1, 58084 Hagen, Germany

1 MOTIVATION AND RESEARCH PROBLEM

For Higher-Education Institutions (HEIs) Learning Management Systems (LMS) as well as Learning Content Management Systems (LCMS) are nowadays essential elements of educational infrastructures. Driven by this trend, innovative learning scenarios emerge, triggering the evolution of this kind of e-learning software systems and corresponding overall academic teaching and training infrastructures. At the same time, fresh features in such educational software systems can inspire ambitious ideas for further developing teaching- and learning concepts or organization corresponding software application scenarios, so that the state of development of e-learning-software constantly progresses. In this way the needs for cognitive efficient and at the same time technically interoperable, transparently usable tools to support effective learning content creation and consumption approaches of their users are growing.

This means that rapid improvement of these technologies is appreciated, but it is coming with a price-tag: so called naïve users, i.e., users without computer science knowledge, education and training have to keep learning the usage of new software, because formerly accustomed workflows are constantly changing. Therefore, tutors, teachers and professors - especially when they are not working in the field of information technologies - need to invest a distinctively high amount of time to get familiar with every piece of unknown software in their scholastic infrastructure. Even if the recent developments might help to work more efficiently and effectively - in the end, there are still many hours to invest in learning new skills: these hours are lost for educational processes.

Besides this, modern learning scenarios do not only require teachers to get familiar with - also students need to learn about available technologies

and services; teachers are constantly faced with students' questions about new platforms, features and add-ons, so that a certain amount of their teaching time goes into explaining the bits and pieces of most recent learning tool developments to their students. Especially in adult education and if students are not 'digital natives' they often fail to understand the use of new platforms quickly enough to allow for a seamless learning experience.

In summary: while new tools for the development and evolution of e-learning infrastructures and learning environments are needed, students and teachers should not only be embraced by the change but feel inclined to try out new features without being confronted with having to learn every single feature of bulky platforms and software-packages, such as the LMS and LCMS Moodle (Moodle.org, 2015) which is in the focus of our work described in this paper.

In addition, depending on the organization of HEIs, teaching assistants are employed to ease the way into new technologies for students and to help with the process of studying. However, as teaching assistants eventually are recruited from the experienced student-corpus, they typically stay only for a limited amount of time in this job; again a towering amount of paid working time is lost when they leave from their position. The problem of such high fluctuation of costly trained and specialized workforce emerges frequently and adds to the work load of the permanent staff.

As a consequence, for example, an easy to learn and use authoring tool for producing content for LMSs would not only help to save time and trouble for the teachers, but would also help HEIs to save monetary resources in training teaching assistants.

This paper will outline our initial approach towards helping to solve the identified problems by developing an effective and at the same time flexible and powerful learning content authoring-tool which can integrate different technologies seamlessly while

remaining comfortable and cognitively efficient to handle for non-computer scientists.

2 OBJECTIVES AND APPROACH

Since Moodle is widely known and used as an LMS and LCMS and other than its proprietary competitor *Blackboard* (Blackboard.com, 2015) is a so-called open source solution it is the platform of choice for our experimental scenario of course-authoring and execution support for a distance university. Moodle as an LMS and LCMS in combination with several other tools is considered a powerful and at the same time open and flexible base system for our research.

In a first step, we have started to interview teachers, staff and students at the University of Hagen (German Fern-Universitaet of Hagen: the largest distance-education university of the German-speaking countries, in short FernUni) in order to identify the need for and requirements towards additional tools to be integrated with the already existing Moodle tool suite.

At FernUni in Hagen, several online tools have been developed during the last decades, which are individually tailored to the needs of faculty, staff and students. Most of these initially used in-house software solutions are nowadays to some degree outdated and have already been replaced by now but some of them are still in use today, for example, the assignment tool *WebAssign*, which is described at (Campussource.de, 2015).

Originally developed at FernUni's chair of *Software Engineering* as a project of the so-called Campus Source initiative, it is until today widely accepted by faculty and staff because it supports some particular FernUni-specific workflows. Many teachers worked with *WebAssign* for years and spent a lot of time developing their assignment scenarios and teaching environments with it. Therefore, they would not accept its replacement, but *WebAssign* has to be kept as a legacy tool; it should become integrated with and ideally embedded into Moodle.

A first step to find a solution is to explore the possibilities and features of the *Learning Tools Interoperability (LTI)* standard. According to (McFall, et al., 2014; Developers.imslobal.org, 2015a), LTI specifies a way for integrating learning applications into e-learning platforms. Each university or school which uses a proprietary learning environment or specifically developed tools to foster learning processes can benefit from LTI. The Moodle open source software development

community has identified this need and created a plug-in, offering a way to integrate LTI-compatible tools. Therefore, if *WebAssign* is to be integrated with Moodle, first of all it has to be made LTI-compatible.

In summary this means that the overall goal and initial objective of the first step of our research work is to develop an authoring tool for Moodle courses, which will make it easier for teachers to intuitively create new learning scenarios in Moodle, aiming for a genuine user-interface supporting the WYSIWYG-paradigm while at the same time supporting the integration of assignments into such Moodle courses on the basis of a *WebAssign* integration.

Within its course model, Moodle supports many so-called *activities*, offering features like assignments, forums, chatrooms or wikis. All these activities are implemented as Moodle plug-ins. They are sufficient for most requirements but they do not offer the same comfort and variety of functionality like specialized software focusing on a single aspect of e-learning support, and so it is necessary to connect external tools and platforms to Moodle to be enabled to use the advantages of Moodle and of already established external legacy tools and platforms.

Therefore and in addition to our first objective, a resulting requirements analysis and system integration specification for embedding *WebAssign* (Campussource.de, 2015) to Moodle will be introduced in the remainder of this paper.

For the integration of *WebAssign* with Moodle, parts of the Moodle source code have to be altered and suitable integration interfaces have to be developed. The process of bridging tools through such interfaces in compliance with the LTI standard needs to be supported and so a next objective - and at the same time our first milestone - is to derive a suitable integration architecture and a corresponding software-engineering model for the resulting development project.

In addition to courses concerning regular study paths, FernUni supports advanced training programs. Sometimes, these programs run in cooperation with external institutions, which in a few cases have their own e-learning platforms. In our case, one of the affiliated institutions is heading towards supporting so-called *competence-based learning scenarios* and pushes the integration of competence management features into Moodle. For further information about competence-based learning see (Tencompetence.org, 2015a; 2015b).

Supporting competence-based learning is not the main topic of this paper although it does affect and

constrain the way to integrate external tools like WebAssign into Moodle; additional information related to competence management support has to be inserted, shared, managed and exchanged which is currently not covered by LTI, and a solution going beyond these limitations of LTI has to be found too.

As a summary, it can be stated that our project consists of three parts:

- Integration of external e-learning tools into Moodle via LTI;
- Providing a way to exchange the necessary meta-information supporting competence based learning between Moodle and external tools.
- Developing an authoring tool which offers a user-friendly way to create and modify Moodle courses.

This paper focusses on the first part of our above-mentioned project. As a prototype, the connection between Moodle and WebAssign will be designed and implemented. In LTI jargon, Moodle is the *tool consumer* because it accesses resources from the external tool. WebAssign provides these resources; hence it is the so-called *tool provider*. The Moodle-side of this connection will be less problematic since Moodle contains a consumer-plug-in by default. Turning WebAssign into a tool provider is the greater challenge as it is not currently supporting LTI.

As far as possible, the connection between Moodle and WebAssign should obtain the look and feel of a Moodle plug-in: more precisely an activity. When launching the activity, ideally users should not even notice that they are working outside of Moodle now. To avoid conflicts with our aim to create a solution which can be transferred on other e-learning tools this point is assigned a lower priority.

3 TECHNICAL STATE OF THE ART AND INITIAL ARCHITECTURAL EXPERIMENTS

The idea of integrating external e-learning tools with Moodle is already supported: the Moodle plug-in *external tool*, described in (Docs.moodle.org, 2015), provides the necessary interface technology. This tool enables Moodle users having admin or teacher privileges to integrate protected resources from external applications into Moodle courses via LTI. The user does not need to have any further specific knowledge to use it, but some basic understanding of

LTI and how it works is needed.

Creating such a connection can be quite simple, as long as the external tool supports LTI; on (Developers.imsglobal.org, 2015b) a list of certified tools is available. Some universities, for example, the University Duisburg/Essen, adapted their internal developed assignment software for this purpose. However, their tool *Jack*, described at (S3.uni-duisburg-essen.de, 2015), is catered to the needs of the University Duisburg/Essen and their way to implement LTI support does not suit with our requirements.

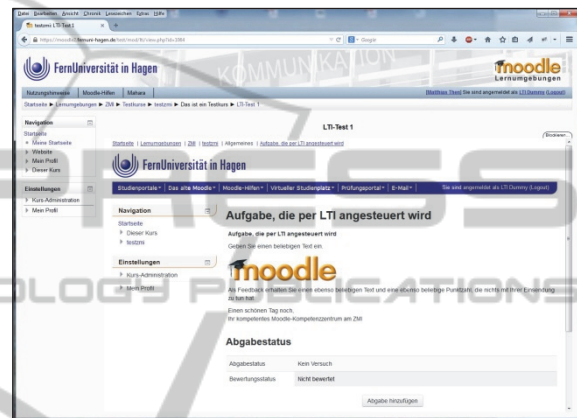


Figure 1: Two Moodles, connected via LTI.

Moodle can also be used as a tool provider; that just requires the installation of a plug-in provided by J. Leyva (Leyva, 2015), which is not included by default.

To get a first impression of working with LTI, we have built up a test environment containing two separate Moodles: Moodle A acts as tool consumer, Moodle B as a provider. Figure 1 displays an example, where an assignment in Moodle B is used by a course in Moodle A. It probably would make sense to suppress header and blocks of Moodle B. However, here both Moodle instances are displayed for demonstrational purposes only. When launching the assignment, the active user in Moodle A is automatically logged into Moodle B. The login-procedure itself is handled completely by the provider.

Moodle's consumer plug-in offers services for receiving information such as scores. The provider plug-in periodically requests these services by a *cronjob*, which identifies new results and transfers them back to the consumer. Finally, the information exists at the provider and at the Moodle course grade book.

The consumer plug-in is a very important feature

to realize the project but a closer look at the provider plug-in is necessary as well because it grants a first impression of upcoming development for extending WebAssign to support LTI.

4 TECHNICAL INTEGRATION METHODOLOGY

The key topic of the following sections is the inspection of *LTI* and *OAuth* with a view to our first milestone - embedding external e-learning tools into Moodle. A resulting prototype will be described later. This model will bridge the interfaces of Moodle and *WebAssign* by using the *LTI* standard.

4.1 IMS LTI

LTI specifies a standard way for integrating learning applications such as forums, chat rooms, wikis, assignment tools and video-streaming into platforms such as LMS. In LTI-context, tools are called *tool providers*, and platforms are *tool consumers*.

The Moodle community saw the potential of LTI and developed those two plug-ins described in section 3. The latest Moodle release, version 2.8, contains an improved version of the consumer plug-in, which supports LTI v2.0. Earlier versions support LTI v1.1.

4.1.1 Versions and Terms

These versions exist:

- LTI v1.0 (McFall, et al., 2010);
- LTI v1.1 (McFall et al., 2012a);
- LTI v1.1.1 (McFall, et al., 2012b);
- LTI v2.0 (McFall, et al., 2014);
- LTI v1.2 (McFall et al., 2015);

The following features are essential for the development of our prototype:

- Launching external resources (since v1.0);
- Returning outcomes to the consumer (since v1.1).

4.1.2 Launching External Resources

In a Moodle course, a connection to an LTI-tool-provider is specified by an activity-instance of the type *external tool*. Its configuration contains authorization parameters as well as essential information like the URI of the resource to connect to. The resource could be, for example, an assignment. When a user enters a Moodle course,

the connection is represented by a link to the activity-instance; via mouse click a request for launching the resource is sent.

To secure the access, LTI uses the standard *OAuth* for authorization (see section 4.2). The verification of a request works as follows: On the consumer side, a request has been signed with a token, generated from the *consumer key* and the *shared secret*. The provider uses the *shared secret* to decrypt the token. In case the result is the consumer key, the sender is identified as Moodle respectively a Moodle resource, namely an activity of type external tool.

Once the sender is authenticated, further parameters - such as user, context and role - will be identified. Until this stage, the request is only known as generated by a Moodle activity. The provider will handle the decision upon granting access to any demanded resource. If approved, the user will be logged in automatically, meaning that the provider will create a user session. The provider tool then can be launched in Moodle - for example, in an iframe or a separate window.

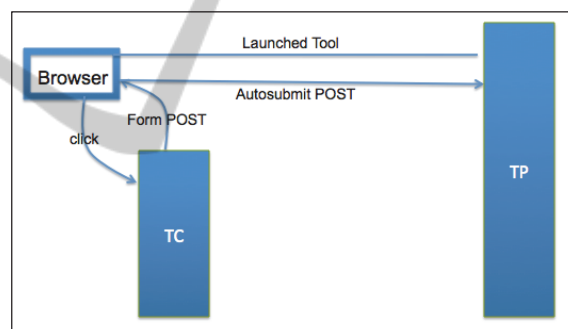


Figure 2: LTI v1.0 (Developers.imsglobal.org, 2015a).

Figure 2 shows the process of launching a tool. A Moodle user starts this process by clicking on any link to an activity-instance of the type *external tool*, representing a connection between Moodle and a protected resource at the provider-side. In this example, the tool consumer 'TC' prepares the launch parameters, signs them via OAuth and then sends a message back to the browser from where it was originally posted. From there it is automatically - this feature is likely to be developed in JavaScript - forwarded to the provider 'TP'. The provider verifies the consumer via OAuth, and in the case of success, it checks further submitted parameters. If access is approved, the external resource can be launched in Moodle.

4.1.3 Returning Outcomes to the Consumer

This feature is part of LTI since version v1.1. Its purpose is enabling learning platforms to receive assessment information like scores and comments from the external tools. Moodle stores this information in the students' grade books.

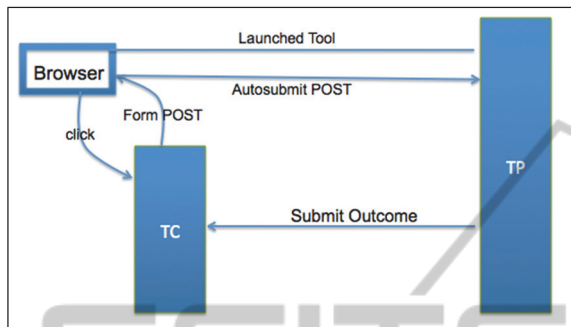


Figure 1: LTI v1.1 (Developers.imsglobal.org, 2015a).

The message, which contains the information for the consumer, is completely asynchronous with the tool launch. The provider-plug-in in Moodle works with a *cronjob*, which is, depending on the configuration of the system - executed every few minutes.

4.1.4 New Features in V2.0 and V1.2

Returning outcomes to the consumer has been the first feature in LTI including web services. Version 2.0 introduces additional services, offering according to its developers (McFall, et al., 2014) the following:

- A wider range of information and not just a score can be returned as the so-called outcome;
- Providers are enabled to inspect consumer-side resources like courses and course sections;
- Providers may check enrolments in consumer-courses;
- Providers can be configured to store data on the consumer-side.

LTI v2.0 not only defines its own functionalities, it offers an interface for importing additional services - in other words: LTI-extensions can be developed in-house and later be integrated. The LTI core is not affected by that, because those services are treated as plug-ins. Especially when keeping in mind a further milestone, beyond the objectives of this current paper, the option of importing in-house services could be the key to exchange information for *competence based learning*.

From the viewpoint of security and privacy,

some of the new possibilities can be regarded as critical. Therefore, LTI offers a standardized way to define individual *interface contracts* between the involved applications. Such a contract, called *ToolProxy*, determines which services shall be used by the participants.

LTI v1.2 has been developed at the same time and is limited to a subset of version 2.0, for example, it does not include ToolProxies; a detailed comparison is shown in (Developers.imsglobal.org, 2015a).

4.1.5 Benefits for Institutions

The involved groups benefit from LTI in different ways: A developer of external tools does not need to have a deeper understanding of the providing platform itself, and might handle it as a black box. Furthermore, she can choose her preferred language and IDE for the development process. Probably the most important advantage is that a single user interface is suitable for any consumer.

The administration of the learning platform, usually a learning management system, profits from LTI as well. It is no longer required to install and test different integrations for each tool-provider. Besides, security and performance of the LMS are not affected, because the connected tools are running in separate environments. Furthermore, updates of the LMS are less risky, as long as the underlying LTI plug-in is not corrupted by the update. Another helpful feature consists of the possibility to authorize teachers so that they also can add LTI connections.

Last but not least teachers and learners benefit from LTI, because it minimizes the efforts necessary for using of external applications, and the choice of tools is only limited by the institution, encouraging the creation and development of further applicable e-learning tools. This process nourishes an environment for improving the quality of modern education.

4.2 Short Introduction to OAuth

OAuth is a standard for authorization, which has been developed from the *Internet Engineering Task Force (IETF)*. LTI uses OAuth v1.0a, for detailed information see (Hammer, 2009; Hammer-Lahav, 2010). The newer version v2.0, which has been released in October 2012, is controversial because of security issues. As it is not used in LTI, we focus on v1.0a.

OAuth specifies a secure process, which enables a consumer to access a protected resource provider-

sidewise. At first, the resource owner authorizes the consumer to open the resource; the outcomes of this process are the two parameters *consumer key* and *shared secret*. If successfully authorized, the consumer may access the resource by submitting an access-token, which is generated from the consumer key and the shared secret; the resource cannot be used unless the authentication process approved the request. Without the access-token, the resource owner would have to identify herself directly with credentials as username and password, which cause communications to be prone to man-in-the-middle attacks.

4.2.1 A User-centered Integration Scenario

The following example assumes LTI support both on Moodle- and on WebAssign-side. The user *teachersmith* has owner privileges on a protected resource *assignment1* in WebAssign. She also has the role of a teacher in a Moodle course *course99* and wants to integrate *assignment1* into this course.

The user logs into Moodle and creates an activity-instance *activity13* of the type *external tool* in *course99* to assign the launching-URL for *assignment1* to *activity13*. After that she logs into WebAssign and authorizes *activity13* for accessing *assignment1*. The outcome of that process is the login-information for *activity13*, consisting of a key and a secret - in LTI-jargon *consumer key* respectively *shared secret*. In a final step, our user adds these parameters to the configuration properties of *activity13*, and *assignment1* can be launched by clicking on the link to *activity13* in Moodle.

The verification of a launching request works the following way: On Moodle-side, the request has been signed with a token, which has been generated from consumer key and shared secret. The provider uses the shared secret to decrypt the token. If the result is the consumer key, the access is approved.

4.3 Concept of Integration

Since Moodle v2.8, the consumer plug-in supports LTI-version v2.0, which is the latest LTI-version at that time. Modification of this plug-in will not be necessary until we start to work with competences. The consumer plug-in offers two features, which are essential for our project:

- Authentication via OAuth v1.0a;
- Receiving outcomes from WebAssign and entering them into the Moodle grade book.

4.3.1 WebAssign as LTI-provider

On the provider-side, there is a lot more to do, because WebAssign is not supporting LTI at the moment. First of all, a requirements specification is needed; that means some decisions about the capability range of the Moodle-WebAssign-integration have to be made.

4.3.2 Requirements and Decisions

A typical course at the FernUni is divided into four to seven blocks with each block consisting of an exercise booklet. These booklets serve as containers for assignments. Since this structure was part of the requirements for the development of WebAssign at that time, it is perfectly integrated. Unfortunately, WebAssign limited in this way of organizing assignments. Concerning our prototype, we have to decide, what exactly has to be integrated into Moodle – the assignments, the booklets, or both?

The core feature of the connection is to launch a WebAssign-resource from within a Moodle course. Up to now it is not cleared, which types of resources and which workflows shall be opened for Moodle:

- Students working on their assignments;
- Teachers creating and editing assignments;
- Teachers using the correction workflow.

Since students should not realize WebAssign as a separate software, the first workflow is essential.

Some assignment types require manual marking and commenting; such assignments can, for example, be handed in as *pdf* files. WebAssign offers a capable workflow to manage the marking process - including features like automatic distribution of assignments between teachers and teaching assistants. The distribution modes can be individually configured, which would be an advantageous feature in Moodle, making the usability of external tools even more desirable.

WebAssign offers an interface, which enables teachers to employ additional tools for auto-correcting assignments. In the department for computer science teachers used this feature to integrate their own modules for analyzing program code written in Pascal, Java or C++. These additional tools will probably not affect the development of connections between Moodle and WebAssign directly, but they should be kept in mind for further improvements.

4.3.3 Considerations and Technical Aspects

Resource owners in WebAssign need an interface to

authorize and withdraw connections. For example, the user *teachersmith* from section 4.2.1 must be able to authorize *activity13* for launching the resource *assignment1*.

LTI requires OAuth-support, which is currently not available in WebAssign, so an appropriate extension has to be discovered and installed. Successfully OAuth-verified requests have to pass some further checks:

- Sender: The sender needs to be identified, and data must be compared to the list of verified WebAssign-users. Furthermore, it has to be checked, if the sender is authorized to access the resource.
- Requested resource: An approval of the request is needed, because several resources only have a granted access during limited time periods.

The sender has to be logged in automatically with the *LDAP* account, if the access is permitted. After that, a secure session must be initialized. Next step is to generate a browser-readable presentation of the requested resource.

Last but not least a workflow for returning information such as outcomes to Moodle has to be developed. As this process will be asynchronous, a possible solution might be a cronjob.

5 EXPECTED OUTCOME

Key issue of this paper is the first milestone in our project: a concept for integrating external e-learning tools into Moodle via LTI. Later, this concept will be extended to support *competence based learning*. As this is not covered by LTI, a solution to exchange competence-specific information has to be found.

To gain the necessary technical experience, and data for this research project, an LTI-connection prototype will be designed and implemented from scratch, implying the extension of the tool provider for LTI-support. Working with LTI-ready tools is not sufficient in this scenario, because later the connection must be extended for the use of competences; therefore knowledge of a deeper level is inevitable. The prototype will connect Moodle and WebAssign; latter was the tool of choice, because its Moodle-integration is demanded at the FernUni.

Once the prototype is created, the next agenda item can be tackled: extending the prototype for the use of competences. Most likely not just the consumer- and provider-protocol have to be modified, but also the connection itself. As mentioned in section 2, FernUni is cooperating with an external institute, and planning to merge

competence based learning with its concept. At this stage of the project, the scheduled course authoring-tool is becoming significant, because it is part of this cooperation. Ideally, this approach will have the potential to be generalized or be treated as a kind of best practice.

6 STATE OF OUR RESEARCH

We are currently working on the requirements specification of our prototype. To identify the needs of users, we are leading interviews with teachers, teaching assistants and students. In section 4.3.2, a list of upcoming decisions concerning the range of WebAssign's LTI-support is shown; furthermore some technical aspects are described.

To gain technical expertise with LTI, we first analysed Moodle and its possibilities for integrating external tools, and so we experimented with the plug-ins described in section 3, to create a sample connection between two separate Moodles as shown in Figure 1. After that we developed an LTI-provider.

As soon as the requirements' specification for the Moodle-WebAssign-integration is complete, we are ready to start designing and implementing our prototype.

REFERENCES

- Blackboard.com, (2015). *Blackboard | Reimagine Education | Education Technology & Services*. [online] Available at: <http://www.blackboard.com> [Accessed 6 Mar. 2015].
- Campussource.de, (2015). *CampusSource Software WebAssign*. [online] Available at: <http://www.campussource.de/software/webassign/index.html> [Accessed 6 Mar. 2015].
- Developers.imsglobal.org, (2015a). *IMS Global Developer*. [online] Available at: <http://developers.imsglobal.org/tutorials.html> [Accessed 6 Mar. 2015].
- Developers.imsglobal.org, (2015b). *IMS Global Developer*. [online] Available at: <http://developers.imsglobal.org/catalog.html> [Accessed 6 Mar. 2015].
- Docs.moodle.org, (2015). *External tool - MoodleDocs*. [online] Available at: https://docs.moodle.org/28/en/External_tool [Accessed 6 Mar. 2015].
- Hammer-Lahav, E., (2010). *RFC 5849 - The OAuth 1.0 Protocol*. [online] Available at: <http://tools.ietf.org/html/rfc5849> [Accessed 6 Mar. 2015].
- Hammer, E., (2009). *OAuth*. [online] Available at: <http://hueniverse.com/oauth/> [Accessed 6 Mar. 2015].
- Leyva, J., (2015). *Moodle plugins directory: LTI Provider*.

- [online] Available at: https://moodle.org/plugins/view.php?plugin=local_ltiprovider [Accessed 6 Mar. 2015].
- McFall, G. et. al. (2015). *IMS Global Learning Tools Interoperability® Implementation Guide Version 1.2*. [online] Available at: <http://www.imsglobal.org/lti/ltiv1p2/ltiIMGv1p2.html> [Accessed 6 Mar. 2015].
- McFall, G. et. al. (2014). *IMS Global Learning Tools Interoperability Implementation Guide Version 2.0 Final*. [online] Available at: <http://www.imsglobal.org/lti/ltiv2p0/ltiIMGv2p0.html> [Accessed 6 Mar. 2015].
- McFall, G. et. al. (2012a). *IMS Global Learning Tools Interoperability Implementation Guide Version 1.1*. [online] Available at: <http://www.imsglobal.org/LTI/v1p1/ltiIMGv1p1.html> [Accessed 6 Mar. 2015].
- McFall, G. et. al., (2012b). *IMS Global Learning Tools Interoperability Implementation Guide Version 1.1.1*. [online] Available at: <http://www.imsglobal.org/LTI/v1p1p1/ltiIMGv1p1p1.html> [Accessed 6 Mar. 2015].
- McFall, G. et. al., (2010). *IMS Global Learning Tools Interoperability Version 1.0*. [online] Available at: <http://www.imsglobal.org/lti/blti/bltiv1p0/ltiBLTIimgv1p0.html> [Accessed 6 Mar. 2015].
- Moodle.org, (2015). *Moodle - Open-source learning platform | Moodle.org*. [online] Available at: <https://moodle.org/> [Accessed 6 Mar. 2015].
- S3.uni-duisburg-essen.de (2015). *JACK – Specification of Software Systems*. [online] Available at <http://www.s3.uni-duisburg-essen.de/forschung/e-learning-and-e-assessment/jack/> [Accessed 6 Mar. 2015].
- TenCompetence.org, (2015a). *TENCompetence Overview | TENCompetence*. [online] Available at: <http://tencompetence-project.bolton.ac.uk/node/181/index.html> [Accessed 6 Mar. 2015].
- TenCompetence.org, (2015b). *The TENCompetence Personal Competence Manager V1.0 | TENCompetence*. [online] Available at: <http://tencompetence-project.bolton.ac.uk/node/96/index.html> [Accessed 6 Mar. 2015].
- Vickers, S. (2013). *LTI saves lives - A case study of how e-Reflect was LTI-enabled*. 1st ed. [ebook] Available at: http://www.celtic-project.org/Project_blog/2013/05/LTI_saves_lives/LTI_saves_lives.pdf [Accessed 6 Mar. 2015].
- Vickers, S. (2012). *IMS Learning Tools Interoperability - A Briefing Paper*. 1st ed. [ebook] CETIS - centre of educational technology and interoperability standards. Available at: <http://publications.cetis.ac.uk/wp-content/uploads/2012/05/LTI-Briefing-Paper.pdf> [Accessed 6 Mar. 2015].