

PAPER • OPEN ACCESS

Supporting E-Learning Materials Development Cycle Through Office Application Add-in

To cite this article: A A G Y Paramartha and I G P Sindu 2019 *J. Phys.: Conf. Ser.* **1165** 012019

View the [article online](#) for updates and enhancements.

You may also like

- [Comparing the effectiveness of learning formats in radiation protection](#)
L van Puyvelde, T Clarijs, N Belmans et al.
- [The Impacts of Demographic Variables on Technological and Contextual Challenges of E-learning Implementation](#)
Hanan Aldowah, Samar Ghazal, Irfan Naufal Umar et al.
- [A Strategy using Social Media to Increase Students Engagement in E-Learning for the Equivalency Education Program \(E-LEEP\)](#)
Mesra Betty Yel, Sfenrianto Sfenrianto, Gunawan Wang et al.



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

Supporting E-Learning Materials Development Cycle Through Office Application Add-in

A A G Y Paramartha^{1,a} and I G P Sindu²

¹Informatics Management Department, Universitas Pendidikan Ganesha, Singaraja, Indonesia

²Informatics Engineering Education Department, Universitas Pendidikan Ganesha, Singaraja, Indonesia

Email: ^ayudhi.paramartha@undiksha.ac.id

Abstract. The development of e-learning material is not only done once, but is carried out with repeated cycles. The cycle can simply be divided into 3 parts, including making, publishing, and reusing existing material. Various systems have been developed to support this cycle. Learning Management System (LMS) is the most common system used in e-learning. However, LMS alone cannot fully support the cycles mentioned earlier. Therefore, many other systems / applications are used to support the cycle. For example, a document processing application is used at the stage of making the learning material, then published to the LMS, search by using search engine and reused with specific authoring tools. In this article, we will discuss how to integrate several applications that support the development cycle of e-learning materials through an office application add-in. The add-in was used to integrate LMS and Learning Material Reconstruction System (LRMS) so that the entire development cycle of e-learning materials development can be done in one place.

1. Introduction

The development of e-learning material is not only done once, but it is done with a repetitive cycle. The cycle can simply be divided into 3 parts, including making, publishing, and reusing pre-existing material (Figure 1). In general, each stage uses several tools to produce quality learning material. For example, to make material in the form of text, presentations, spreadsheets, and so on we used to use office applications. At this stage, learning materials should be reused for further use as a new learning material. New learning materials could be built based on parts of other learning materials that can be reused. It should also support interoperability standards so that several objects can be combined to become new material [1]. Furthermore, the learning material that has been made could be published to the Learning Management System (LMS) so that it could be used for the learning process. LMS is a useful tool in improving the learning process. Most educational institutions have LMS used by teachers, students, and faculty staff [2]. Moodle is one of the most widely used open source LMS in the world. Moodle has many features that can be used to facilitate learning process. One of the useful features is the ability to publish learning material in various formats. Unfortunately, Moodle has limited feature for editing learning material [3].

Published material could be searched, downloaded, and reused as needed. Material that has been published in general could be reused to create new learning material. So that there is a search process before reusing of existing material. The search process is done through search engines, where users enter keywords and search results are displayed. All material search activities are usually carried out in a web browser. Then, the search results are downloaded then taken into parts to be added to the material being



made. In the end, this whole process will be repeated over and over so that the material are always evolves to makes quality learning material.

Based on the process above, each stage uses its own application or tools that are not integrated with each other. This makes users have to learn all the applications one by one. This article discusses how to integrate the development cycle of electronic learning material from the stages of making, publishing, to reusing existing material to create new material. This can be achieved by creating an add-in on the office application. The add-in serves as a bridge that integrates the LMRS (which includes search engines) and LMS. With this add-in, users can carry out all the processes of developing electronic learning materials only through one application.



Figure 1. E-Learning Materials Development Cycle

2. Research Method

The Office Add-in was developed by evaluating the existing system that is currently available. There are 2 main software, including the Moodle-based LMS, and the previously developed Learning Material Reconstruction System (LMRS). LMS serves as a place to publish data and data sources for e-learning material. Moodle has provided web services that can be used directly for various purposes. In this study, the web services used were web services related to the course and file manipulation. In addition, the web services also provide a mechanism for logging into LMS, so that only users who have access rights can access the materials through developed add-in. We used Moodle as LMS in this study. We were choosing Moodle because it is one of the most popular open source LMS in the world.

The material search process required a search engine which includes data collection, information extraction, indexing, and retrieval processes [4]. LMRS provides all these features so that in this study no new system is needed [5]. LMRS provides search engines that can be used to search for existing materials. The search engine owned by LMRS is able to search for materials at the sentence level, so that it can be used as a search engine on the add-in. LRMS actually provides several features that can be used to reuse material, but for the purposes of this research, only the search engine will be used. LMRS does not yet have a web service, so a web service needs to be developed so that material searching can be done outside the LMRS.

After evaluating the existing system, the next step was the system design stage. The add-in runs on Microsoft Office. In this case the selected application was Microsoft Office PowerPoint 2016. The add-in is designed with regard to the system's ability to use the web services. Basically, there were 2 choices in implementing Microsoft Office add-in, namely by using Visual Studio Tools for Office (VSTO) and Microsoft Office Web Add-ins [6]. Some studies used VSTO to extend Microsoft Office's ability to support the learning process [7]–[10]. These extensions are placed in the Ribbon section of the Microsoft

Office user interface. However, VSTO can only be run on desktop-based host applications. While the Web Add-in besides being able to run on desktop-based Microsoft Office can also run on web-based Microsoft Office. The Web Add-in also has several advantages on the developer side. The Web Add-in uses technologies that are used to develop web applications such as HTML, JavaScript, CSS, and supports Document Object Model (DOM) and provides various JavaScript-based APIs for manipulating documents supported by Microsoft Office. We chose the Web Add-in as the main development tool, so that the design developed was also based on web technology.

The next stage was the implementation of the system. System implementation was done by activating the web service on LMS, and creating a web service on LRMS. Both web services were integrated with add-ins using the API provided by the Web Add-in. The result of the implementation is an Add-in that can be used to search for material and access (both download and upload) e-learning material provided by LMS. Making material can be done in Microsoft Office PowerPoint as usual. Search for additional material can be done using the web service from LRMS.

3. System Architecture

The add-in integrates LRMS and LMS through web services. Web services from LMS are used by add-in to authenticate users so that only users who have accounts on LMS get access to the add-in. The web services are also used to download material from LMS for later editing in PowerPoint. When users edit the material, users can search for existing material provided by LRMS. The add-in uses the LRMS web service where users enter the desired keywords. LRMS will respond by giving search results in the form of sentences that match those keywords. Furthermore, if the user has finished editing the material, the user can upload the material to the LMS by utilizing the web service provided by the LMS. In general, the system workflow can be seen in Figure 2.

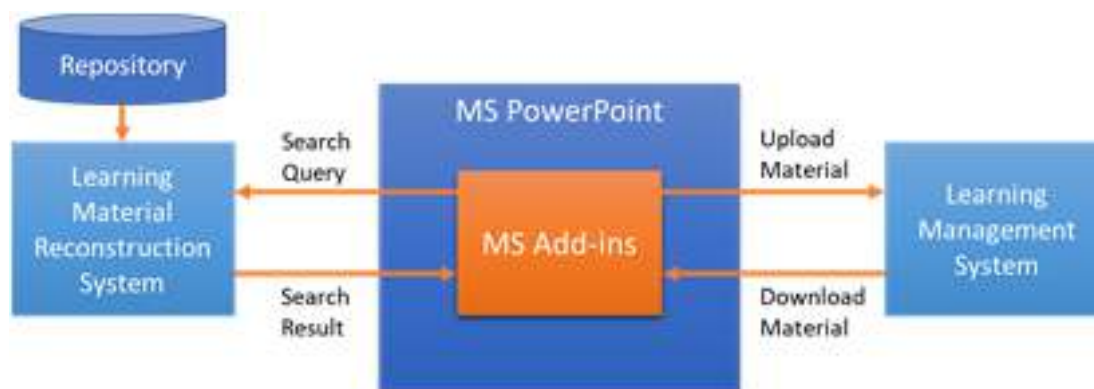


Figure 2. System architecture

4. System Implementation and Results

Office add-ins has a strict security method of accessing web resources. One security method that must be met is that the web resources must meet the HTTPS protocol. This causes all applications to be integrated with add-ins must use a valid HTTPS protocol. For that purposes we have implemented HTTPS on the LMS and LRMS. Before the LMS can be integrated, the web service on the LMS must be activated first. As far as Moodle-based LMS uses the HTTPS protocol and activates the web service, the add-ins can be integrated with it. LRMS does not yet have a web service, so the implementation of web services was done before it can be integrated with the add-in. LRMS provides several modules such as data gathering, feature extraction, material indexing, material reconstruction and material searching. In this study, only the material searching module was used.

Add-ins were implemented using Microsoft Office Web Add-ins in Visual Studio 2017. The programming language used was JavaScript with HTML and CSS as interfaces. There were 3 main modules on the add-in, namely: 1) LMS authentication; 2) download and upload material module; and 3) material search module from LRMS. The results of the development of add-in can be seen in Figure 3.

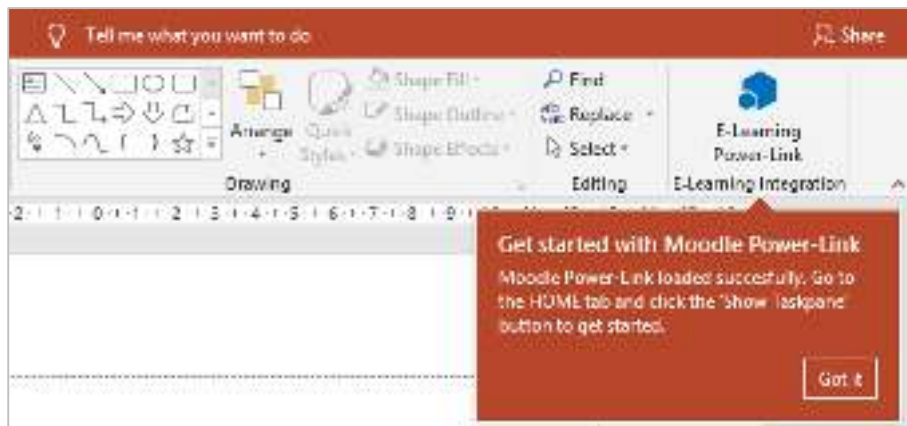


Figure 3. Office Add-in

If the add-in is executed, the user must first log in to the LMS via the login page. The user must enter a username, a password, and LMS-destination URL. As explained earlier, users can use LMS from anywhere as long as the LMS is based on Moodle with a web service that has been activated and uses the HTTPS protocol. If the user successfully logs in, there will be several actions that can be performed, namely: 1) Download document, 2) upload documents, 3) search content. The add-in will also display all courses owned by the user as shown in Figure 4.

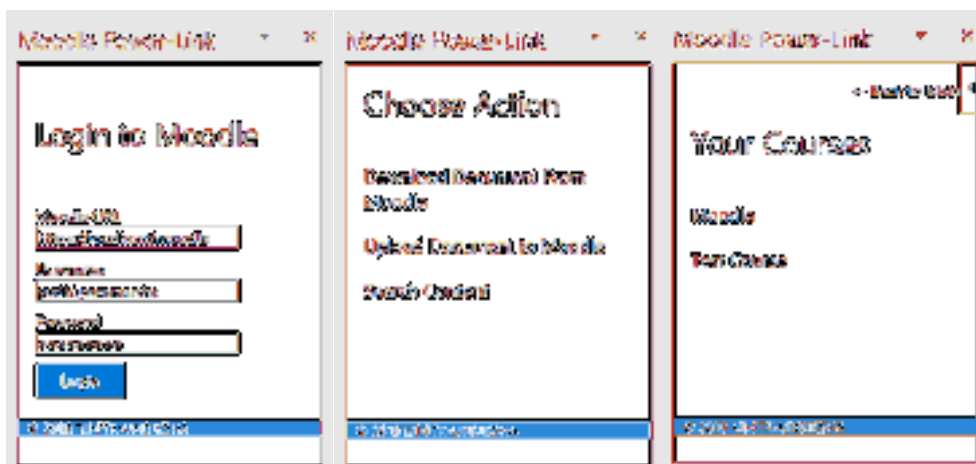


Figure 4. Login, action, and course page

If one course is selected, the course page will be displayed. All materials that uses the * .pptx format are displayed by topic. Users could download the material for editing. Uploading material to LMS is done by clicking on one of the topics and a confirmation page will be displayed. Download and upload features can be seen in Figure 5.

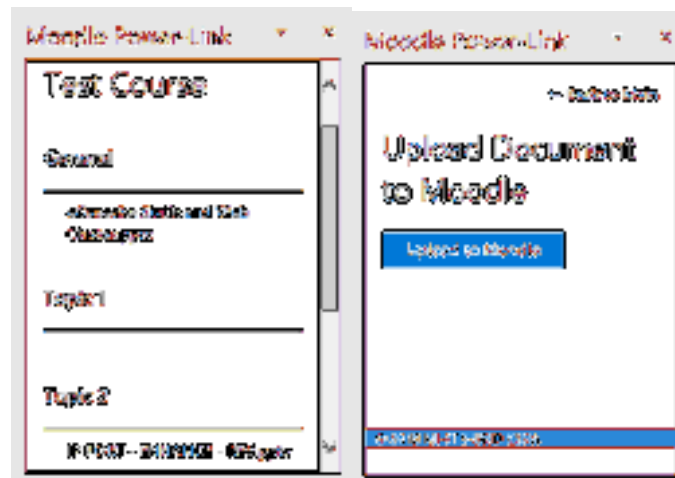


Figure 5. Upload and Download page

The last feature in this add-in is the material search feature from LMRS. These features can be accessed through the "search content" menu after logging in. In this page, there is one textbox that can be given a keyword by the user. If the search button is pressed, a response will be displayed in the form of a list of materials obtained from LMRS. The material is in the form of sentences that have been previously extracted by LMRS from exiting material. By selecting one of the search results, the add-in will automatically insert the material into the presentation slide. Users can do editing as usual on the slide if needed. Examples of searching and entering material into slides can be seen in Figure 6.

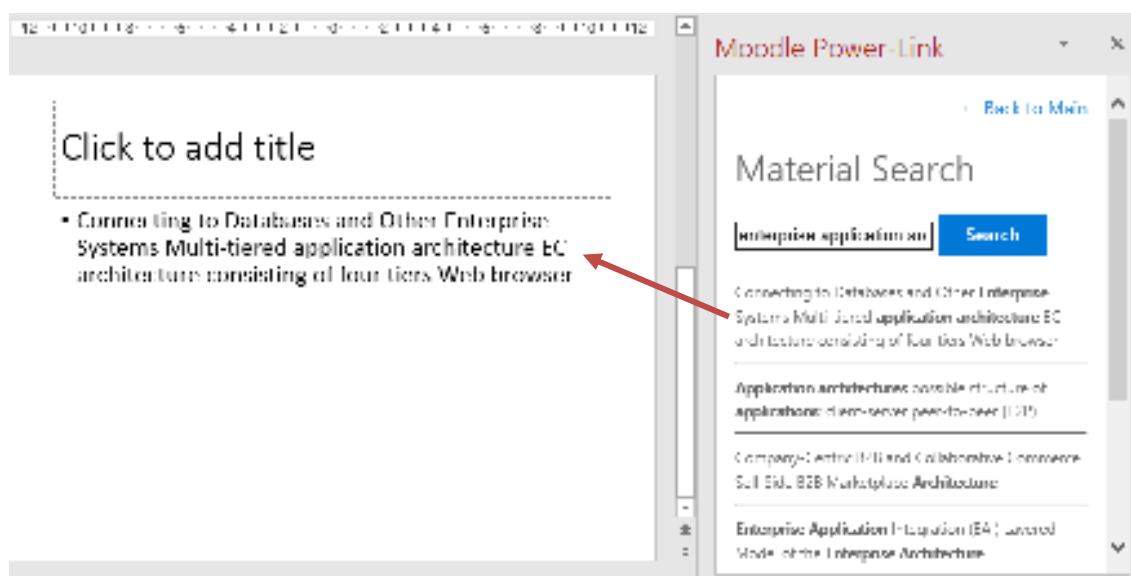


Figure 6. Material searching and insert material to PowerPoint slide

5. Conclusion

This study discusses how to support e-learning material development cycle by integrating LMS and LMRS on an add-in installed in Microsoft Office PowerPoint. Integration was done by activating web services on Moodle based LMS and creating web services in LMRS. The modules used to integrate include authentication, course modules, and uploading/downloading material to and from LMS. In LMRS, a search module was integrated with add-in that was used to search for existing materials. these

existing materials can then be reused for creating new materials. The results of this study are in the form of office add-in installed in Microsoft Office PowerPoint that can be used to create material, publish material to LMS, and search for materials from LRMS to be reused. All these activities are carried out in one place without having to use another application.

References

- [1] M. M. Shaker, A. M. Khidhir, and Y. A. Al-Rizzo, "Design and implementation of a virtual university prototype technology requirements and evaluation," in *Proceedings of 4th International Multiconference on Computer Science and Information Technology*, 2006, pp. 560–568.
- [2] E. Dahlstrom, D. C. Brooks, and J. Bichsel, "The Current Ecosystem of Learning Management Systems in Higher Education: Student, Faculty, and IT Perspectives," 2014.
- [3] A. Iglesias, L. Moreno, P. Martínez, and R. Calvo, "Evaluating the accessibility of three open-source learning content management systems: A comparative study," *Comput. Appl. Eng. Educ.*, vol. 22, no. 2, pp. 320–328, 2014.
- [4] Z. A. Bakar, M. Kassim, M. N. Sahroni, and N. Anuar, "A survey: Framework to develop retrieval algorithms of indexing techniques on learning material," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 2–5, pp. 43–46, 2017.
- [5] A. A. G. Y. Paramartha, I. K. Purnamawan, N. W. Marti, and P. H. Suputra, "Sistem rekonstruksi materi pembelajaran berbasis objek pembelajaran granular," in *Seminar Nasional Vokasi dan Teknologi (SEMNASVOKTEK)*, 2017, pp. 99–104.
- [6] B. Raymond, "Comparing VSTO and Office Web add-ins," *TechGenix*, 2018. [Online]. Available: <http://techgenix.com/comparing-vsto-and-office-web-add-ins-video/>.
- [7] A. PÎRJAN, "Solution for repurposing the default actions and states of the office controls through component object model add-ins," *J. Inf. Syst. Oper. Manag.*, pp. 482–493, 2015.
- [8] D.-M. PETROȘANU and A. PÎRJAN, "Solutions for developing and extending rich graphical user interfaces for office applications," *J. Inf. Syst. Oper. Manag.*, pp. 482–493, 2015.
- [9] L. Xu and W. He, "The Office Software Learning and Examination System Design Based on Fragmented Learning Idea," in *MATEC Web of Conferences*, 2016, vol. 85, p. 1051.
- [10] H. W. Ka, "Real-time Synchronous Slideshow Presentation Access Technology for the Blind," *J. Technol. Pers. with Disabil.*, pp. 91–101, 2016.