Cloud-based Mash-up Authoring Tools for e-Learning

Ivan Madjarov
Aix-Marseille Université, CNRS LSIS UMR 7296,
Domaine universitaire de Saint Jérôme, Avenue Escadrille Normandie Niemen, 13397, Marseille, France

Keywords: Cloud Computing, Authoring Tools, e-Learning, m-Learning, SaaS, Web Services.

Abstract: The objective of an e-Learning authoring tool is the creation of an accessible and reusable pedagogical content that is conforms to the existing e-Learning standards. We argue that this objective can be extended when the learning content is created on a multimodal SaaS authoring and publishing platform with an XML-based language for content and metadata description to satisfy instructional designers for a responsive interactivity coupled with rich and informative substance. This position paper presents a Cloud-services based solution for building a virtual and personal learning environment that combines a wide range of technology and tools for learning-content mash-up authoring and diffusion. The proposed service-based framework is intended to support lifelong learning content creation and to enable mash-up of various learning services and applications by adapting learning objects on desktop and small handheld devices.

1 INTRODUCTION

The current e-learning applications required large investments in infrastructure systems and professional staff to maintain and upgrade systems. It is envisioned that, in the near future, cloud computing will have a significant impact on the educational and learning environment, enabling their own users to perform their tasks effectively with less cost by utilizing the available applications offered by the cloud service providers (Hossain, 2012). Interest in cloud computing in the e-learning area is growing due to potential greater cost savings from scalable architectures and open source products, and the possibility of higher learning outcomes (Fernández, 2012).

Software-as-a-Service (SaaS) is a cloud service which provides software functionality through Internet and can help efficiently the management of web-based applications and pedagogical data in an e-learning and m-learning context. Recently, more and more e-learning tools are migrating to the cloud. For instance, e-learning developers access the content authoring tools over the Internet via a secure hosted system without thinking about IT configurations, software set-ups and licenses. Cloud-based pedagogical content authoring is an e-learning process that is free from the constraints of typical desktop solutions and offers many advantages. In this position paper, we introduce the concept of *e-Learning-Software-as-a-Service* for e-learning and m-learning content-editing, storing, content-adaptation and diffusion. All aspects of an e-learning or m-learning solution can be delivered using the Cloud-SaaS model, including Learning Course Management Systems (LCMS), authoring tools, and collaboration tools like webcasting and white boarding (Basal, 2010).

In this position paper, we present our research in an ongoing project organized as follows: in Section 2 cloud-based content authoring tools and evaluation criteria are discussed. Section 3 presents the developed cloud-based e-learning semantic authoring suite. Finally, Section 4 ends the paper with conclusion and future work.

2 CLOUD-BASED COURSE DEVELOPMENT TOOLS

The objective of an e-learning authoring tool is the creation of an accessible and reusable pedagogical content that is conforms to the existing e-learning standards. We argue that this objective can be done if the learning content is created on a multimodal SaaS authoring and publishing platform with an XML-based language for content and metadata description to satisfy instructional designers for a responsive interactivity coupled with rich and in-

Madjarov I..
Cloud-based Mash-up Authoring Tools for e-Learning.
DOI: 10.5220/0004957105260531
In *Proceedings of the 6th International Conference on Computer Supported Education* (CSEDU-2014), pages 526-531
ISBN: 978-989-758-020-8

formative substance.

Authoring process is an essential part of learning process and is supported by a suite of authoring tools for text, hypertext, graphics, mathematics, chart, and questionnaire content creation; i.e., Learning Objects (LOs). The purpose of an LCMS is to define the learning process by using and reusing created and existing LOs. This is a design process that includes: determination of learning objectives; planning how to assess learner comprehension, and identifying content needed so the learner can meet objectives.

We advocate for an approach where the learning process and the creation process are managed separately and independently of each other, i.e.: (1) a LOs authoring process managed by a suite of authoring tools; (2) a pedagogical process of course building, assessment and content adaptation managed by a LCMS.

2.1 e-Learning Authoring

The course development authoring tools are complex programs (e.g. Lectora (Lectora, 2013), Captivate (Adobe, 2014), ToolBook (SumTotal, 2013), SmartBuilder (SmartBuilder, 2014), StoryLine (Articulate, 2014), etc.) that allows the integration of media objects and the creation of interactive learning materials. So many authors started e-learning content creation with *off-the-shelf* software e.g. word processors, spread-sheets, desktop publishing packages, graphics packages etc.

The most famous and readily available of these is PowerPoint, which is powerful and complex software for slides and multimedia presentations but with some limitations concerning different versions and their use in the e-learning domain. For instance, PowerPoint's interface is somewhat cumbersome when creating pedagogical sequences that integrate multimedia components (e.g. text, images, audio and video). A course is designed to be integrated in a LCMS. To be imported or exported into a LCMS an external interface-program (plug-in) should manage the PowerPoint non ASCII file format which is proprietary and different from one version to another (e.g. Microsoft Office, Open Office, Documents To Go, Kingsoft Office, etc.). This lack of internal format of PowerPoint does not contribute to the mobility and interoperability of pedagogical content. In this case, the learning object discovery process (LODP) is hampered by the lack of metadata. The reuse of a course or selected parts of a course by others is practically compromised because this authoring tool is not SCORM or LOs standards compliant.

Except PowerPoint, for course development, there is a list (Tools, 2012) of proprietary, relatively efficient with several multimedia functionalities but very complex and expensive cloud-based content authoring software tools. We comment a few, non-exhaustive examples:

- *Lectora* (Lectora, 2013) offers a suite of online web-based, collaborative authoring tools for e-learning course creation and publishing in a proprietary format with the possibility to export in Adobe-Flash and SCORM compliant format.
- *Easygenerator* is an online free authoring tool. This Microsoft Windows application stores all pedagogical content in the cloud. With the free version author can have courses in an online workspace and can import existing Power-Point presentations for course build of.
- *QuickLessons* (QuickLessons, 2013) is a SaaS-based content authoring tool, requiring the user to only have access to a web browser. This online collaborative platform includes also evaluation engine for assessments, digital repository to manage media files and LOs. The author creates courses using libraries of templates, so it does not need any programming and design skills. Authors can include existing PowerPoint presentations or export a course to Adobe-Flash format or in HTML5 files. Multiple export options are supported for offline, online, mobile and LCMS use including SCORM and AICC compliance.
- *ZebraZapps* (Allen, 2014) is a SaaS-based authoring and publishing platform. This authoring tool allows developers and non-programmers alike to create interactive applications quickly, as well as share, publish, and sell objects or entire applications.

Some activities in these software tools seem "easy-to-use" by limiting development options. For example, they might provide a set of predefined interaction templates for developers to fill with content. This is really useful if those templates reflect exactly the interactions that the author want to create, but if the author want to customize the output these tools appear very restrictive. In addition, these programs have a common drawback. The authors edit a full course in a proprietary format whose components are not metadata described and hard to discover thereafter. The publication in SCORM format in a LCMS dissolves completely the created content in the learning process.

In this paper, we propose a solution with twofold purpose: on the one hand, it aims at presenting an XML-based language for semantic description of a course. On the other hand, it proposes a cloud-based authoring tool to build learning resources applying the defined language. The proposed solution stores away natively courses and facilitates the reuse of parts of a course in other courses because these parts can be easy discovered by an XML-based search method.

2.2 Authoring Tools Evaluation Criteria

The main function of an e-learning authoring tool is to integrate different media and create interactivities required in a learning program. As mentioned above, a number of e-learning authoring tools have been developed with a wide range of functionalities: from simple (e.g. template-based) to very complex with its own programming language and multimodal user interface. These software programs are too complex to be used by an author who is not an IT or an e-learning specialist. Subsequently a large number of users expressed a real difficulty in using these tools.

It becomes clear that e-learning authors need simpler but powerful authoring tools that could lower the skill barrier and allow more actors (i.e. teachers, academic authors, e-learning system administrators) to participate in the development and customization process. Moreover, to be really useful these tools should be able to reduce development time, effort, and cost, by allowing the reuse, enrichment and customization of available learning contents (Capuano, 2009). This necessity has motivated research and development of MOOCs (Massive Open Online Course) making use of cloud-based learning tools and online tools as well as learning support specifically for LOs creation and sharing (Rizzardini, 2013). MOOC is an online course aimed at unlimited participation and access via the web.

In this paper we introduce some evaluation criteria essential for the appropriate operation of an e-learning authoring tool:

- The authoring tool should be easy to learn and use because, in general, the author is not a professional e-learning developer and his experience in multimedia application is limited.
- The authoring tool should have the capacity to integrate media objects in various formats in the learning content with maximum flexibility.
- For educational subjects such as mathematics, informatics and science in general, some of the contents involve mathematics equations, graphs and diagrams. The authoring tool should provide the functionalities for such content creation.
- The authoring tool should provide the possibility to create quizzes for learners' assessment.
- An authoring tool should not require authors any programming skills.
- The authoring tool should provide deployment method for a LCMS with SCORM compliance integration.
- The authoring tool should also provide many alternative deployment methods for standalone (XML, HML or PDF), or for the web publication without any content reproduction during the deployment stage.
- The authoring tool should avoid the proprietary format for e-learning content for a maximum interoperability and reusability of LOs. Otherwise the LOs discovery and composition process are doomed to failure.
- The authoring tool should be accessible as service with standardized web browser.

To these *technical* criteria, we add a key *technological* one: *cloud service-based e-learning authoring process*. The cloud authoring is free from the constraints of typical desktop solutions. Advances in internet technology make appear web-based tools with collaborative capabilities, but also with some limitations than advantages. Some web-based tools are cumbersome with an input process using templates and forms, which results in a limited creativity and flexibility. The pivotal advantage of cloud-based content authoring is not the number of connected users but saving time, reducing rework, and sharing learning content across all projects. Authors access authoring software over the Internet via a secure, affordable hosted system, and regardless of location they can make updates and reusing content to complete a course. The adaptation of a cloud-based approach brings many benefits and relieves the author from embarrassing task (Fernández, 2012).

We argue that an effective solution for a successful cloud-based authoring process passes by: (1) the choice of a common format for the totality of the course content, and (2) the choice of an accessible and reachable storage format for the course. We suggest a non-proprietary format for better LOs reuse and to facilitate the LOs discovery and assembly, i.e. XML-based. Once created, a course can be saved in a single XML document or as a collection of XML documents. According to the evolution of the course, authors may need to modify its content. Therefore, the correct operation of a collaborative authoring system imposes the storage of learning collections, possibly in an appropriate database, for a better reuse and diffusion of these documents. We suggest the choice of a native database (NXD) which allows the storage of XML documents in their native format. This choice, in opposition to that of a relational database, is explained by the nature of pedagogical documents which are of narrative type, i.e. *document-centric* and not *data-centric* that is typical for relational tables. NXD and RDB follow two very different approaches to managing data. The concept of XML documents is to keep all attributes for a given LO together as one entity so that it can easily retrieve or insert all the relevant information for an LO in one shot. This is mostly useful for complex LOs whose attributes would otherwise be spread over many relational tables in the RDB model. XML is also suitable for LO whose set of attributes can evolve over time, since XML offers schema flexibility. Although relational database products today provide built-in XML document and query support, NXDs are arguably the best choice for learning content and metadata storage. As far as the query language is concerned, XQuery is recommended. Interoperability and data integration with existing e-learning systems and LCMS can be done by the means of cloud-based services. This solution simplifies the LOs reuse and the process of LOs discovery. As side effect of this approach, we can see a marked improvement of interoperability between e-learning systems.

3 E-LEARNING AUTHORING AS A SERVICE

This section presents a solution for building a virtual and personalized learning environment which utilizes a cloud-based technology to create a service-oriented model for e-learning and m-learning application service providers and learners. The concept of *e-Learning Services-as-a-Software* is introduced as software distribution model in which applications are hosted by a service provider and distributed via the web. In this environment we can easily combine semi-structured data, stored in a native XML database (NXD), with structured data stored in a relational database (RDB) through web services (WS). The objective is to provide direct data and application integration, located at distributed sites in order to improve the achievement of learning outcomes, i.e. integrate authoring process managed by a *cloud-service management system* (CSMS) (Fernández, 2012) with pedagogical process managed by a LCMS. This approach promotes a device-independent gateway between different units and the huge number of learning resources available on a plethora of LCMSs. A semantic content adaptation service is plugged for content standardization. It becomes possible by combining the *web-based open semantic editor suite* (WOSES) (Madjarov, 2011) with a set of additional cloud-services to allow different mobile units a direct access to LOs customarily designed for desktop web browsers. An alternative service is available for a speech solution, which allows learners to turn any written text into natural speech files, when using standard voices. This approach allows the generation and the progressive downloading of text and audio based learning material dynamically for m-learning and ubiquitous access. As result, we cover the essential part of introduced criteria in section 2.2. Details are presented in the following section.

3.1 e-Learning Authoring Tools

The tools we developed are inspired by mashing-up principles allowing an easy and fast integration, over defined schemas, of local and remote resources.

Xesop is an open source authoring software (Madjarov, 2012) that provides a flexible XML-based suite of tools for author customization, editing, storage and publication of LOs compliant with existing e-learning standards. The content aggregation specification comprises two models: a metadata model specifying the metadata elements of learning resources, and a content packaging model representing content structure. Both are hierarchical, which is convenient for representing data consisting of many elements and sub-elements. XML is perfectly suited for representing hierarchical models, as exemplified by the LOM and content packaging XML binding specifications published by IMS, both of which are adopted in SCORM.

In Figure 1 the structure of the created course with the Xesop project's semantic tools is presented. The original LOM structure is extended with new elements that seem important for more homogeneous and identifiable pedagogical content. For encoding textual information and content assembly, an XML semantic editor is developed (Figure 2) and a tree structure of a generic learning document is generated, while a validation grammar of XML schema type is used. Depending on course specificity, (mathematics or informatics course and science in general), the author can represent texts, diagrams, mathematical formulas or data in tables. A MathML editor was created for mathematical expressions, a SVG editor for vector graphics creation (Figure 3), a QTI editor for student's progression evaluation, a schema for table generation and a chart editor for data presentation. In this case, XML is used for encoding non-textual information such as vector graphics and mathematical expressions.

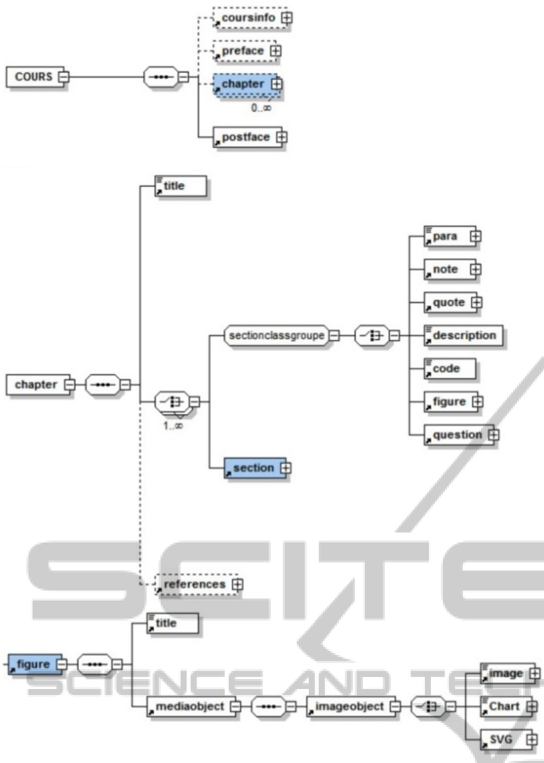
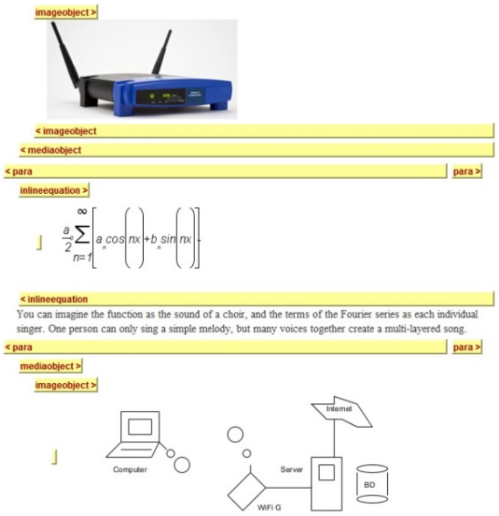


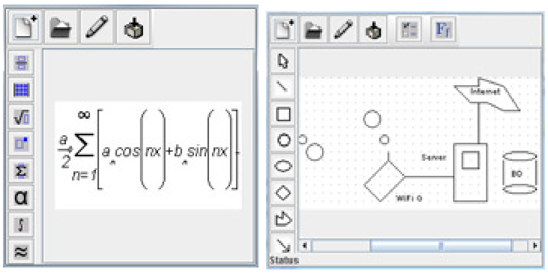
Figure 1: An extract of the Xesop course semantic.



$$\frac{a_0}{2}\sum_{n=1}^{\infty}\left[a_n\cos\left(nx\right)+b_n\sin\left(nx\right)\right]$$

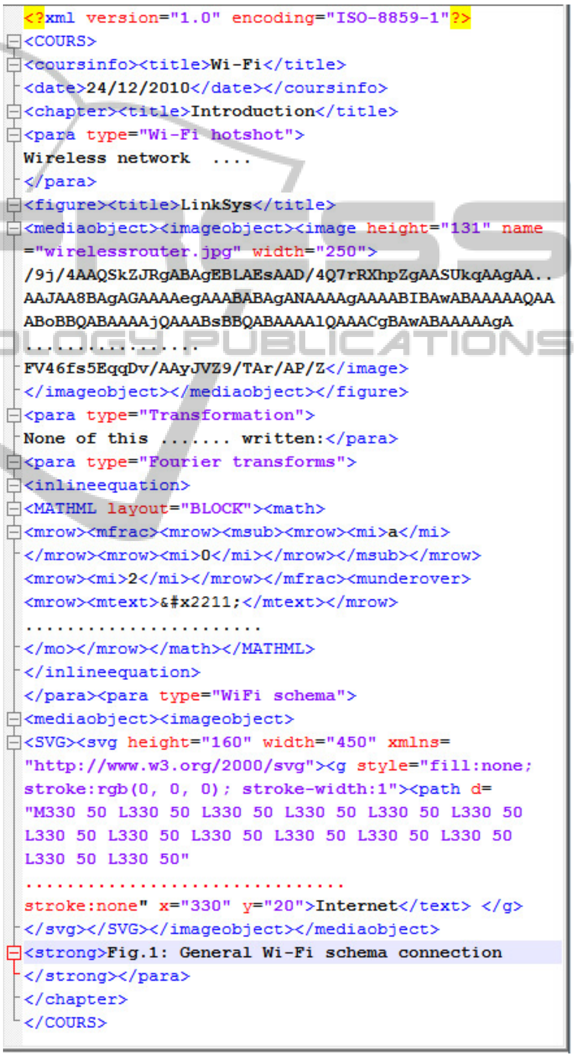
You can imagine the function as the sound of a choir, and the terms of the Fourier series as each individual singer. One person can only sing a simple melody, but many voices together create a multi-layered song.

Figure 2: Xesop XML semantic editors.



$$\frac{a_0}{2}\sum_{n=1}^{\infty}\left[a_n\cos\left(nx\right)+b_n\sin\left(nx\right)\right]$$

Figure 3: MathML and SVG plug-ins.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<COURS>
<coursinfo><title>Wi-Fi</title>
<date>24/12/2010</date></coursinfo>
<chapter><title>Introduction</title>
<para type="Wi-Fi hotshot">
Wireless network  ....
</para>
<figure><title>LinkSys</title>
<mediaobject><imageobject><image height="131" name
="wirelessrouter.jpg" width="250">
/9j/4AAQSkZJRgABAgEBLAEsAAD/4Q7rRXhpZgAASUkqAAgAA..
AAJAA8BAgAGAAAAegAAABABAgANAAAAgAAAABIBAwABAAAAAQAA
ABoBBQABAAAAjQAAABsBBQABAAAAlQAAACgBAwABAAAAAgA
.................
FV46fs5EqqDv/AAyJVZ9/TAr/AP/Z</image>
</imageobject></mediaobject></figure>
<para type="Transformation">
None of this ....... written:</para>
<para type="Fourier transforms">
<inlineequation>
<MATHML layout="BLOCK"><math>
<mrow><mfrac><mrow><msub><mrow><mi>a</mi>
</mrow><mrow><mi>0</mi></mrow></msub></mrow>
<mrow><mi>2</mi></mrow></mfrac><munderover>
<mrow><mtext>&#x2211;</mtext></mrow>
........................
</mo></mrow></math></MATHML>
</inlineequation>
</para><para type="WiFi schema">
<mediaobject><imageobject>
<SVG><svg height="160" width="450" xmlns=
"http://www.w3.org/2000/svg"><g style="fill:none;
stroke:rgb(0, 0, 0); stroke-width:1"><path d=
"M330 50 L330 50 L330 50 L330 50 L330 50 L330 50
L330 50 L330 50 L330 50 L330 50 L330 50 L330 50
L330 50 L330 50"
..............................
stroke:none" x="330" y="20">Internet</text> </g>
</svg></SVG></imageobject></mediaobject>
<strong>Fig.1: General Wi-Fi schema connection
</strong></para>
</chapter>
</COURS>
```

Figure 4: Course XML collections.

In our e-learning authoring suite, binary data of multimedia content is embedded directly into XML course content. During the editing process, if the author inserts an image or any binary data into the edited content, the semantic editor will encode it using the Base64 encoding method. As a result, the course collection can be managed easily since all materials relating to the course are stored in a single XML collection (Figure 4). For more flexibility and content reuse, author can separately save different elements of the course as LOs in the NXD as XML collection with multiple files.

A large number of media-rich content can be stored using the abundant set of available XML schemas. By providing proper XSLT transformation files, the XML content can be presented in many forms, such as HTML5 for web-based desktop or mobile users. If necessary, formatted HTML and PDF versions of extracted learning content can be published in a LCMS via web services. Existing and old pedagogical documents can be adapted in the compliant XML format via importing web services as external input in the XML semantic editor. M-Learning pedagogical content can be given in the form of a visual presentation as text, pictures, tables in XML, responsive HTML5 format or as PDF data. Optionally, m-learning content can be given as sound data in the form of an acoustic presentation in an MP3 or WAV format (Madjarov, 2012).

3.2 Implementation Scenario

The implementation scenario highlights the WOSES cloud-based application integration with a LCMS. The interconnection is carried out by a *web services management system* (WSMS). In our scenario, the *learning-centric data* and the *management-centric data* are clearly separated. Pedagogical documents are developed in WOSES framework of the *e-Learning Services-as-a-Software* based *Xesop* system and thereafter are stored in a NXD. The information relevant to learner personal data, learner profiles, course maps, LOs sequencing, data presentation and general user data is stored in the RDB of the LCMS. The publication process of learning content is carried out by WSMS. This allows integration existing LCMS systems via a cloud-based service. In the discussed case, web service-based modules make the bridge in a simple and effective way through Apache Libcloud, an open source library that provides a system-neutral interface to cloud provider APIs. The Java version supports Amazon EC2.

For system deployment, we used AmeTice (Moodle-based) e-Learning system deployed at Aix-Marseille University. Our system is based on the Apache containers suite for data storage and service management. We integrated the PHP-based LCMS interface via web services. For services deployment we used Apache Axis. For storing and managing LOs, we used eXist running in the Apache Tomcat Servlet engine as a web application and invoked via REST-style web services API. To integrate with other e-Learning and/or m-Learning systems we implemented an Apache jUDDI registry.

4 CONCLUSIONS

The developed Cloud-SaaS solution makes possible to create adaptive and responsive e-learning courses. Authors need to follow, in general, four phases of course development. To satisfy the phases of analysis, design, development and updating a free XML authoring language is used. The developed editor suite provides high-level WYSIWYM visual tools and enables a complete course to be designed without writing any programming code. We believe that a future cloud-based e-learning system should consist of a set of independent but cooperating non-monolithic services-based applications that integrate pedagogical data between common LCMS.

REFERENCES

Adobe Captivate, http://www.adobe.com/Captivate, last visited January 2014.

Allen Interactions, ZebraZapps, http://www.alleninteractions.com/products/zebrazapp, visited January 2014.

Articulate, StoryLine, http://www.articulate.com/, last visited January 2014.

Basal A. M., A. L. Steenkamp, A Saas-Based Approach in an E-Learning System, *IJISM*, pp. 27-40, June, 2010.

Capuano Nicola, et al., A mash-up authoring tool for e-learning based on pedagogical templates, *The first ACM international workshop on MTDL*, ACM, New York, USA, 87-94. 2009.

Fernández A., D. Peralta, F. Herrera, J. M. Benítez, An Overview of E-Learning in Cloud Computing, *Workshop on Learning Technology for Education in Cloud, LTEC'12*, pp 35-46, 2012.

Lectora, http://lectora.com/e-learning-software/, v. 2014.

Ivan Madjarov and Omar Boucelma, XESOP: a Content-Adaptive m-Learning Environment, EC-TEL'12, LNCS 7563, pp. 531–536, Springer 2012.

Ivan Madjarov and Omar Boucelma, Multimodality and Context-adaptation for Mobile Learning, Social Media Tools and Platforms in Learning Environments, pp. 257-276, Springer, 2011.

MD. Anwar Hossain Masud, Xiaodi Huang, A Novel Approach for Adopting Cloud-based E-learning System, In: IEEE/ACIS, pp. 37-42, 2012.

Rizzardini R. H., Gutl1 C., Chang V., MOOCs Concept and Design using Cloud-based Tools: Spanish MOOCs Learning Experiences, The Sixth Conference of MIT's LIT Consortium, Cambridge, USA, 2013.

SmartBuilder, http://www.smartbuilder.com/, last visited January 2014.

SumTotal, ToolBook, http://tb.sumtotalsystems.com/, last visited December 2013.

Tools, The Ultimate List of Cloud-Based Authoring Tools, http://elearningindustry.com/the-ultimate-list-of-cloud-based-authoring-tools, 2012.

QuickLessons, http://www.quicklessons.com/, last visited December 2013.