

CREATING INTERACTIVE AND TRACEABLE EPUB LEARNING CONTENT FROM MOODLE COURSES

¹⁾ Martin Takev, ²⁾ Miguel Rodríguez-Artacho, ³⁾ Elena Somova
¹⁾³⁾ University of Plovdiv "Paisii Hilendarski" – Plovdiv (Bulgaria)
²⁾ UNED University – Madrid (Spain)

Abstract. Technology enhanced learning is shifting from a centralized platform environment to a variety of elements to support and enrich interactions between learners and educational material. In this context, the paper addresses the creation of eBooks in EPUB format based on the existing content in a Moodle environment. The main purpose is to create enriched eBooks capable of supporting and tracking student activity usually reserved to educational environments. The work describes a plug-in developed to support Moodle course translation to EPUB and details potential user tracking formalization using xAPI.

Keywords: Learning Analytics; xAPI; educational content modelling; educational authoring; EPUB; e-learning environment

1. Introduction

One of the main objectives in the field of technology enhanced learning is to enrich educational material with tracking mechanisms. Since the creation of e-learning standards, this effort has been carried out by several implementations focused on structuring the learning content and providing mechanisms to track interactions in web-based learning environments (Artacho & Verdejo, 2002; Caswell, Henson, Jensen, & Wiley, 2008).

The widespread use of web based learning environments, however, does not provide an effective way to achieve ubiquitous learning, nor a seamless way to interact with educational media. Usually students tend to print or make handouts in paper format in order to have access to a more user-friendly material. However, in the evolution of e-learning standards, there is a transition of paradigms as process of education is not any longer web-centred, as it has diversified to multiple technologies, including eBooks. In the times of SCORM, the interaction is supported by the Runtime environment, with a heavy implementation in the server-side, which implies that we have a local tracking and semantically non interoperable with other educational systems. The creators of SCORM have evolved to a cloud-based tracking called xAPI which achieves semantic interoperability by means of a set of statements and verbs linked to a given namespace which can be shared by multiple users (Saliah-Hassane, 2014).

This paper reports on the possibilities of EPUB format when implemented in a tool that translates Moodle courses. The main goal is creating interactive and traceable EPUB learning content from standard Moodle e-learning courses. Current existing approaches do not cover the whole Moodle course, they simply translate content and do not consider interaction (Bozkurt & Bozkaya, 2015).

Section 2 describes EPUB format and reveals the possibilities for interaction. Section 3 starts with description of the Moodle model for structuring courses and existing software tools for conversion in EPUB format. Section 3 continues with presentation of our Moodle plug-in creating EPUB course from traditional Moodle e-course. In Section 4 is shown how xAPI is used in our EPUB course to handle interactions.

2. Why EPUB? Content and interactivity

The EPUB format was adopted and standardized by IDPF in 2007²⁾ as an open format for eBooks. According to (Arenas & Barr, *The Digital Book in Higher Education: Beyond the Horseless Carriage*, 2013) eBooks in general have emerged as a mass-market commercial success within the past few years. To date, eBooks only replicate the static content of printed books in a digital medium, but EPUB3 introduces mechanisms to provide interactivity and multimedia integration by using JavaScript and HTML5. With these characteristics the EPUB3 can provide similar tracking to web-based learning environments, but with a more ubiquitous device, moving forward anytime-anywhere education. As a result, the EPUB3 format offers a complete solution for portable, interactive, connected content, and it is relatively simple to map the requirements for an interactive learning activity onto baseline EPUB capabilities.

2.1. EPUB structure

The EPUB format is basically a compressed (*.zip) file containing a structure as shown in Fig. 1.

Main component is the manifest file that structures and references the rest of the elements. EPUB content is structured by the *.opf file, which determines the index, and the *.html files which describes the book content and formatting (using *.css files).

EPUBs can be downloaded and read on different devices like smartphones, tablets, PCs or e-book readers. All modern devices have e-book reading software. Thus, the EPUB resources can be used by students or teachers both off-line and on-line with the user's favourite contemporary device and reading software. Nowadays, it continues its evolution as recommended standard from the Book Industry Study Group for packaging content in book publishing industry. EPUB is rapidly gaining momentum, because it is the most widely supported vendor-independent XML-based e-book format. Because it is supported by the largest number of hardware readers, it is easily used.

How does EPUB works?

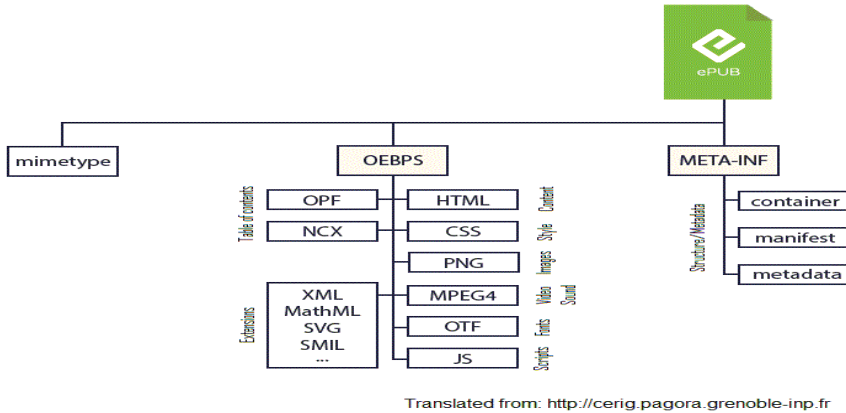


Fig. 1. Structure of an EPUB (Source: edrLAB)

2.2. Interactivity in EPUB using xAPI standard

As pointed out by (Arenas, Richards, & Barr, The IEEE Actionable Data Book: A platform for inclusive education, 2013) the evolution of eBook readers' devices leads to be considered as the new runtime environment where learner interactions are tracked and learner behaviour is structured using xAPI capabilities to formalize user interactions.

In this work, EPUB has been selected precisely because it allows to use JavaScript embedded code to describe the user actions by means of xAPI statements. Surprisingly, a traditionally passive eBook can be turned into a rich interactive educational platform tracking learner activity by means of the Learning Record Store (LRS) that holds the structured interactions in a cloud-based model.

In the specification Experience API[®] (or xAPI) is described as a next generation e-learning standard, which combined with SCORM allows e-learning to use modern technologies in an interoperable way. The xAPI is based on specification Activity Streams, which capture the activity on social networks, created by companies like Google, Facebook, Microsoft, IBM, etc. It will basically obtain the learning traces of the students by tracking and storing their learning experiences (Nouira, Cheniti-Belcadhi, & Braham, 2018).

A couple of the most important characteristics of the xAPI are:

- Operates with disconnected or occasionally connected environments;
- Provides the learning content outside of the Learning Management Systems (LMS);

- Provides data, which can be used to design better interactive and adaptive learning experiences;
- Delivers opportunities to create multi-modal learning experiences by analysing the provided data;
- Reports data from team-based training;
- Reports data from multiple tests (scores) and many test attempts.

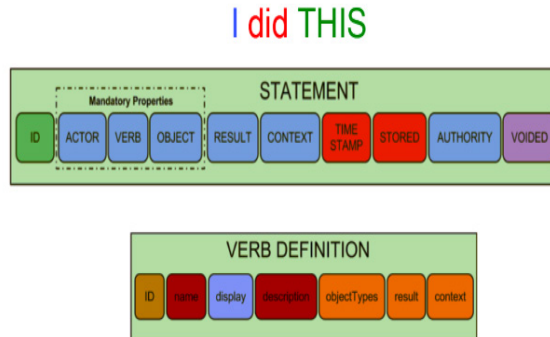


Fig. 2. xAPI statements (Source: ADL¹)

The xAPI collects data about the wide range of experiences a person has done online or offline in statements in the form of “Noun, verb, object” or “I did this” (see Fig. 2). The xAPI gets data in a consistent format about a person or group’s activities from many technologies. In this way very different systems could securely communicate by this stream of activities.

Interaction is achieved by a system that not only presents static data to students, but takes into account the actions they perform on it and their progress throughout its content (Amrieh, Hamtini, & Aljarah, 2015). User activities are saved as statements in the LRS. Further reports and data mining techniques can be applied to them at a later stage (Chakravarthy & Raman, 2014) or the LRS can share these statements with other LRSs. The LRS can exist on its own, or inside an LMS.

3. Building a Moodle plug-in for conversion of e-course to EPUB book

The main goal of this attempt is to create a plug-in for Moodle⁴, which converts e-learning courses into EPUB files. The major challenge is the conversion to interactive EPUB course.

The module is used to create learning resources, suitable for mobile devices where students can study learning content in a different format, answer questions/ assignments, get feedback from a teacher or the system and in overall tracking and storing their learning traces (Nouira, Cheniti-Belcadhi, & Braham, 2018).

3.1. Learning resources and activities in Moodle

Moodle is a free online LMS, which is widely used. Moodle can meet needs of different users (teachers, students, authors and administrators), customizing its core features. Moodle proposes a variety of resources and activities to construct one dynamical learning course, using any exciting methodology:

- *Resources* – file, folder, URL, label, page, book, etc.;
- *Activities* – lesson, quiz, assignment, workshop, database, wiki, glossary, chat, forum, external tool, etc.

Activities require sometimes students to interact with other actors in the learning process and like in (Amrieh, Hamtini, & Aljarah, 2015) the system needs to store and analyse the performed actions. Such formal learning in Moodle can be executed with the help of a plugin⁵⁾ that emits events from the system logs and sends them to a LRS as xAPI statements.

3.2. eLearning platforms and EPUB conversion

There are only a couple of adds-on⁶⁾ for the e-learning platforms, that realize some conversion from/to EPUB: Lucimoo, custom plugins for Moodle and SHEilds.

The EPUB format is not supported by Moodle. Lucimoo³⁾ consists of two plugins for the Moodle book module. One plug-in that provides functionality to import material from EPUB e-books into a book in the Moodle book module, and another plug-in that provides functionality to export books from the Moodle book module as EPUB e-books. The import adds-on, in addition to importing an entire book, can import chapters as individual books. The EPUBs can be imported from files and from URLs. The export adds-on can export a book even with imported EPUB e-books in the content.

EPUB book export tool for Moodle⁶⁾ is a plug-in that adds the possibility to export a book module inside a Moodle course. It adds a link inside the book settings which will download the generated EPUB file. There is also an option to include an image to the book.

The plug-in, used by SHEilds eLearning platform⁷⁾ allows lesson modules to be exported to EPUB documents. The plug-in is capable to export document pages and question pages (although some answers are not shown, such as short answer questions). This SHEilds plug-in is presented in Moodle⁶⁾ as Lesson Export EPUB tool.

Wiki export plug-in for Moodle⁶⁾ adds the ability to export Moodle wikis as either PDFs or EPUB documents. Originally plug-in was commissioned by the Judicial Institute for Scotland for their Totara LMS. The plug-in allows to order the pages in the export.

The drawbacks in Lucimoo come with the scope of exportable content. The option to only export and import book modules greatly restricts the data that can be used in a generated EPUB as Moodle courses have in times several different types of modules. There are courses in Moodle that consists only of book module, but

they are a small excerpt. Book modules contain only static content and cannot incorporate interaction into the created EPUB file, thus making the content not so appealing to the learners.

The main drawback of SHEilds is that it wasn't made specifically for Moodle and its usage is not optimal. Exporting a whole lesson module, as well as question pages provides the level of completeness of the EPUB file and the interactive part of it. But there are still areas to improve like range of exported modules and display of questions.

3.3. Exporting e-course to EPUB book

In order to create an easily accessible and usable module, a plug-in is done to create a new block. This block could be then added on every Moodle page. It lists all the courses available for exporting and contains links to download them in EPUB format.

In the current, first version of our plug-in (Fig. 3), we succeeded to transform the following Moodle components: file, folder, URL, label, page, book, lesson and quiz with certain types of questions. Conversion of file, folder, URL, page, label and book gives the static part of the EPUB course.

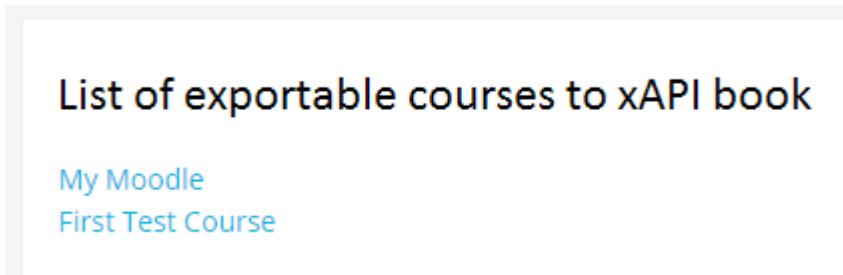


Fig. 3. View of the developed plug-in for export of Moodle courses to EPUB

The plug-in takes the data directly from the corresponding tables in the Moodle database and creates the HTML view, showing all available exportable courses. The courses are exported to an e-book, that lists all the course activities one after another and in the end of the document there is a table of contents (ToC).

Lesson and quiz create the more complicated interactive part of the EPUB course, including both lessons' document and question pages. The quiz contains a great assortment of questions/assignments; therefore, the assessment of the student can be made as said in (Nouira, Cheniti-Belcadhi, & Braham, 2018) with a limited number of common question types: multiple choice question, multiple answer question, essay, matching, short answer, true/false question and cloze test/gap fill.

The processing of quiz questions is done, using JavaScript. The correct answer is hidden in the file content and then compared with the answer given by the student. Besides being evaluated, the answers are at the same time captured and send as xAPI activity statements, so the quiz data can be interpreted (like in (Chakravarthy & Raman, 2014)) at a later point in time. The lesson module consists of parts that are shown conditionally. The user may see one learning content or another, depending on the actions he/she takes during the interaction with the lesson content. The workflow in the document is accomplished again with the use of JavaScript. Specific parts of it are hidden, so to be later made visible and the other way around. Additionally, the student's progress thru the learning material is stored with another one of xAPI's 'endpoints' – the State API, which describes the connection between a user and a material, saved in a given "state". This is especially helpful when the learning material is accessed thru different devices or if the device does not have its own internal storage⁹).

4. Model for recording learning activities using xAPI

An xAPI profile¹⁰ offers documentation of specific vocabulary and statement templates for implementing xAPI. Profiles follow practices, keeping them discoverable and reusable. It is common to see different profiles meant for reflecting the same activity, thus complicating the use of the data mined. Other complications, presented in (Corbi & Burgos, 2014), occur with the missing abilities of xAPI to record consistent and interoperable data. Institutions adopt xAPI to continue tracking students learning process beyond the normal course, where they are physically present.

Finally, we are going to propose a xAPI profile and define some verbs. The xAPI profile for an interactive EPUB learning content could consist of the following actions: content oriented and interaction oriented.

4.1. Content oriented actions

Tracking a user as he/she is accessing and viewing the content can be useful for checking whether enough time is being spent on the learning content:

- *User has accessed page x;*
- *User has left page x;*
- *User has stayed x minutes on page y.*

Furthermore, implementation of statements that track users highlighting specific parts of the learning content or adding comments to specific sentences and phrases, can lead to insight on which are the most valuable parts of the study content or parts not written well, that aggregate into misunderstood content, misleading students learning process or simply generate more questions in students.

There are visible markers that show if a student had interest in a learning material like actions, performed on the learning content:

- *User highlighted text in document;*
- *User added comment to text in document.*

Knowing the assessment score of the student, several conclusions can be made using the data stored on how long he/she has spent on the educational materials:

- *User knew the content from before* – If there is little or no time spent on the material and the assessment score is high, then the student must have known the material and has studied it before;
- *User has learned the content* – If the assessment score is high and there's sufficient time spent on the material present, then the student must have spared time on the material and learned it;
- *User did not study the content* – No time spent on the material and a low score means student did not have any interest in learning the content neither in studying for the assessment;
- *User had difficulty learning the content* – Student having a low score with a considerable time spent on the material means that there were problems comprehending it and maybe rewriting the part of the material that's being assessed or giving some support to the student is appropriate.

Finally, tracking opening files and links inside the EPUB file helps determine whether the user has gone through all of the content. Only students actively engaged in the study material search for additional learning content besides the one that's obligatory:

- *User opened file;*
- *User accessed link.*

4.2. Interaction oriented actions

Interaction activities in the xAPI Specification¹²⁾ define how to process structures for interactions and assessments in traditional e-learning. They are intended to provide a simple and familiar utility for recording interaction data. Listed below are different kinds of interactions that may aid in the assessment of the learners and understanding their behavior:

- *User selected answer x* – Marking an answer without then submitting it could either mean that this happened by mistake or in some cases – that the user is unsure of the correct answer and is hesitating between answers. Then after analysing the users' behaviour, if there is data that several people have been in fact hesitating between the same answers, then it can be concluded that the learning content does not provide good instructions, the question is poorly written or just too complicated. There may be questions that are purposely written this way to require more effort from the learners in answering them.

- *User completed question/test x in y minutes* – data for question/test completion time gives great understanding of material complicity level and whether learners found it easy or hard to complete it. In relation to the time spent on the learning content, several conclusions can be made like how much time is needed to assimilate the new material.
- *User answered correct/wrong* – influences the flow in which the learning material is shown to the learner (show additional learning resources if users have scored high on tests/questions and vice versa).

The following standardized *verbs* can be used for the above-mentioned actions¹¹:

- *attempted* – indicates the actor made an effort to access the object. An attempt statement without additional activities could be considered incomplete in some cases;
- *commented* – indicates the actor provided digital or written annotations on or about an object;
- *completed* – indicates the actor has finished or concluded the activity normally;
- *exited* – indicates the actor intentionally departed from the activity or object;
- *failed* – indicates the actor did not successfully pass an activity to a level of predetermined satisfaction;
- *initialized* – indicates the activity provider has determined that the actor successfully started an activity;
- *interacted* – indicates the actor engaged with a physical or virtual object;
- *launched* – indicates the actor attempted to start an activity;
- *passed* – indicates the actor successfully passed an activity to a level of predetermined satisfaction;
- *progressed* – indicates a value of how much of an actor has advanced or moved through an activity;
- *scored* – indicates a numerical value related to an actor's performance on an activity.

The following standardized *activities* can be used for the above-mentioned actions¹¹:

- *course* – represents the learning material of an entire “content package”. The largest level of granularity. Unless flat, a course consists of multiple modules;
- *module* – represents any “content aggregation” at least one level below the course level. Modules can exist for layering purposes. Modules are not content. Modules are one level up from all content;
- *question* – part of an assessment and requires a response from the learner, a response that is then evaluated for correctness.

```
{
  "actor": {
    "objectType": "Agent",
    "name": "John Doe",
    "account": {
      "homePage": "https://www.example.com/users/"
      "name": "5"
    }
  }
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/scored"
    "display": {
      "en-GB": "completed"
    }
  }
  "object": {
    "objectType": "Question"
    "id": "http://example.moodle.com/mod/quiz/view.php?id=9"
  }
}
```

Fig. 4. Example statement, sent to xAPI, when a user answers a question in a test

On the base of these verbs and activities are written statements, which are sent to xAPI to show user activity. Fig. 4 shows an example statement that will be sent to xAPI when a user answers a question in a test.

5. Conclusions

In this paper we have shown that xAPI is a promising architecture to carry out an educational experience and to implement student tracking. We have also presented the use of a tool to map Moodle courses into EPUB3 format using xAPI as the runtime environment.

EPUB capabilities have not been fully explored yet. We plan to implement a more abstract modelling of the educational content, not only using Moodle structure of courses, but using an educational modelling language to explore the authoring process in EPUB, as we have already done with web based environments in the past.

Current effort of existing special interest groups (SIG) in ADL and IEEE to develop interactive EPUBs are also a promising reference, and will be helpful in the achievement of these goals. This work illustrates a little piece in the overall puzzle of learning devices and interactions. Our further work will try to provide a more structured authoring and enriched modelling of statements and interactions.

Acknowledgements

We thank eMadrid project 2013-ICE-2715 and Comunidad de Madrid for the support in this development, as long as EU Erasmus+ and UNED mobility program.

NOTES

1. ADL Vocabulary. <http://xapi.vocab.pub/datasets/adl/>. Accessed 10.02.2019
2. International Digital Publishing forum (IDPF). EPUB 3.0.1. <http://idpf.org/epub/301>. Accessed 10.02.2019.
3. LUCIMOO EPUB import/export plugins for Moodle book module. <https://lucidor.org/lucimoo/>. Accessed 10.02.2019.
4. Moodle. <https://moodle.org>. Accessed 10.02.2019.
5. Moodle logstore xapi plugin. https://moodle.org/plugins/logstore_xapi. Accessed 10.02.2019.
6. Moodle Plugins. <https://moodle.org/plugins/>. Accessed 10.02.2019.
7. SHEilds eLearning platform. <https://www.sheilds.org/>. Accessed 10.02.2019.
8. The Advanced Distributed Learning Initiative (ADL). Research & Development Experience xAPI. <https://adlnet.gov/experience-api>. Accessed 10.02.2019.
9. The Learning Technology Manager's Guide to the xAPI, <https://www.ht2labs.com/xapi-guide-download/>. Accessed 10.02.2019.
10. xAPI Browse Concepts. <http://xapi.vocab.pub/browse/index.html>. Accessed 10.02.2019.
11. xAPI Solved and Explained. <https://xapi.com>. Accessed 10.02.2019.
12. xAPI Specification. <https://xapi.com/specification/>. Accessed 10.02.2019.

REFERENCES

- Amrieh, E. A., Hamtini, T. & Aljarah, I. (2015). Preprocessing and analyzing educational data set using X-API for improving students performance. *Applied Electrical Engineering and Computing Technologies (AEECT). IEEE Jordan Conference*, (pp. 1 – 5).
- Arenas, E. & Barr, A. (2013). The Digital Book in Higher Education: Beyond the Horseless Carriage. *30th ascilite Conference 2013 Proceedings*, (pp. 63 – 67). Sydney.
- Arenas, E., Richards, T. & Barr, A. (2013). The IEEE Actionable Data Book: A platform for inclusive education. *IEEE Global Humanitarian Technology Conference (GHTC)*, (pp. 63 – 67). San Jose, CA, USA.
- Artacho, M. & Verdejo, M. F. (2002). High Level design of Web-Based environments for distance education. *Computers and Education in*

- the 21st Century*, Springer, (pp. 275 – 285). doi:10.1007/0-306-47532-4_25
- Bozkurt, A. & Bozkaya, M. (2015). Evaluation criteria for interactive e-books for open and distance learning. *The International Review of Research in Open and Distributed Learning (IRRODL)*, 16(5), 58 – 82.
- Caswell, T., Henson, S., Jensen, M. & Wiley, D. (2008). Open Educational Resources: Enabling universal education. *The International Review of Research in Open and Distributed Learning*, 9(1). doi:1492 – 3831.
- Chakravarthy, S. & Raman, A. C. (2014). Educational Data Mining on Learning Management Systems Using Experience API. *Communication Systems and Network Technologies (CSNT) Fourth International Conference*, (pp. 424 – 427).
- Corbi, A. & Burgos, D. (2014). Review of Current Student-Monitoring Techniques used in eLearning-Focused recommender Systems and Learning analytics. *The Experience API and LIME model Case Study. IJIMAI*, 2(7), 44 – 52.
- Nouira, A., Cheniti-Belcadhi, L. & Braham, R. (2018). An enhanced xAPI data model supporting assessment analytics. *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018*. Belgrade, Serbia.
- Saliah-Hassane, H. (2014). Mobile open online laboratories: A way towards connectionist massive online laboratories with xAPI (c-MOOLs). *IEEE Frontiers in Education Conference*. Madrid, Spain. doi:10.1109/FIE.2014.7044172

✉ **Mr. Martin Takev, PhD. Student**

✉ **Ms. Elena Somova, Assoc. Prof.**

Department “Computer Science”
University of Plovdiv “Paisii Hilendarski”
24, Tzar Assen St.
4000 Plovdiv, Bulgaria
E-mail: takevm@gmail.com
eledel@uni-plovdiv.bg

✉ **Prof. Miguel Rodríguez-Artacho**

Department “Languages and Systems”
UNED University
14, Calle de Juan del Rosal
28040 Madrid, Spain
E-mail: miguel@lsi.uned.es