

# Can e-book technology be enough to support e-learning?

Martin Takev, Elena Somova  
Dep. "Computer Science"  
University of Plovdiv "Paisii  
Hilendarski" Plovdiv, Bulgaria  
{takevm,esomova}@uni-plovdiv.bg

Miguel Rodriguez Artacho  
Dep. Lenguajes y Sistemas  
Informaticos  
UNED Madrid, Spain  
miguel@lsi.uned.es

Manuel Castro, Sergio Martín  
DIEECTQAI Department  
UNED Madrid, Spain  
{mcastro,smartin}@lsi.uned.es

**Abstract**—In this paper a model to track learning experiences from an interactive e-book is presented. For this a digital book coherent with the EPUB3 standards has been made by exporting an existing course in a learning management system and replicating its interactive behavior. Defining trigger points throughout the book and amongst the interactions made by the students, that send data to a learning record store, enrich the e-book and provide the possibility to analyze and preview statistical data for students' commitment to learning the material and their actions over the whole course. The xAPI specification helps collect educational data and store it in a database, which can be used for creating a further set of reports anytime in the future.

**Keywords**—EPUB format, xAPI, learning analytics, learning experience, Educational Data Mining

## I. INTRODUCTION

Technology in education is a useful tool when it acts as an extension of our capabilities, but becomes an uncomfortable straitjacket when it is used for greater glory of itself. One of the roles of educational technologists is to find unnecessary technology and evaluate how many of the artefacts available are in fact a real useful tool or an expendable one. Making education more human can be related to AI-based tailored needs and multi-platform interaction, which is also a consequence of ubiquitous and distance learning, but also can be related to make it simpler [1].

In this sense we focus on e-books and in creating ways to provide more simple technology for educational processes. We describe a model to track learning experiences from an interactive e-book from sources like learning platforms. For this a digital book coherent with the EPUB3 [2] standard has been made by exporting an existing course in a learning management system – Moodle and replicating the interactive behaviour of the LMS using e-book interactions.

On one side the content is exported to e-book format, keeping its sequencing and structure properties. On the other side, we can export interactions by defining trigger points throughout these interactions, that send data to a learning record store (LRS), enrich the e-book and provide the possibility to analyse and preview statistical data for students' commitment to learning the material and their actions over the whole course. Finally, experience API specification (xAPI) can be used to collect educational data and store it in an LRS, closing the circle of educational data from the same student, sourced from different technologies.

One of these examples can be e-book readers, where learning interactions can be traced and stored, providing context to other educational components. On the other hand, educational resources are now IoT elements, providing its own business model and resource optimizations. A good example of resources optimization for education is the

integration of remote laboratories as a key component of training students with real world interaction and behaviour.

Students' performance can be measured just like any other statistics. There always will be students that perform better on given tasks and others that don't. Recording the steps that lead to a student's success or failure is the main object of this paper. Analyzing as much of their actions gives a precise view of what needs to be changed for increased learning and bigger accomplishments.

The purpose of the paper is to present the research on usage of interactive e-books in the field of e-learning. Section 2 provides brief state of the art in this domain. Section 3 shows the methodology of study. The research approach and achieved results are explained in Section 4. The conclusion describes authors' main contributions and possible extension of the work.

## II. STATE OF THE ART

### A. E-books and education

Digital books have become a widely used tool in educational institutions [3]. There are many definitions of what an e-book is, given by [4], [5], [6] and [7], and in summary it can be conceived as an electronic form of a printed book, which includes digital data and can be opened by an electronic reading device [8]. Regardless if its simplicity, e-books can provide interactive elements for tests, feedback, conditional materials and enhance the reading experience, motivating students more than printed lecture notes [9].

We focus on the possibilities given by a digital book coherent with the EPUB3 standard. EPUB3 is an abbreviation of electronic publication and points to the third revision of the standard. It is commonly used for digital books where scripts and reflowable content is intended [2]. We also explore the e-book's ability of handling the equivalent display of an existing course in a learning management system (Moodle) and replicating the interactive behavior of the learning management system (LMS), using EPUB and e-book interactions.

Content can be exported to e-book format, while keeping most of its sequencing and structure properties. Additionally, we can export interactions and prepare to add tracking by defining trigger points throughout these interactions. The trigger points enrich the e-book by sending the data obtained from students to a learning record store and ensure the afterwards overview and analysis of that statistical data (measuring students' commitment in learning the material and checking what were their actions over the whole course). Finally, xAPI can be used to collect and store educational data in a learning records store, thus closing the circle of educational data from the same student, sourced from different technologies.

## B. Learning activities

All actions performed by the students in an educational environment can be assumed as both formal and informal learning activities (whether they have an assessment score written, view a presentation or do an experiment, it all counts as an experience). Lack of activities may lead to passive students not engaged in the learning process. On the other hand, interactive activities, promoting participation lead to development of skills and knowledge [10].

A model to record these learning experiences with xAPI specification is presented. Interactive e-books can be created from different learning resources (content from educational systems, modules in LMS, tests). In this case a course module has been exported to an EPUB file with all of its activities and materials, each providing different interaction possibilities to track.

After collecting the learning experiences' data with xAPI statements, it can be processed to extract meaningful statistics and charts. Analyzing them give a better understanding of the learning process and it can be modified to increase learners' performance and scores. A basic categorization of the available reports and some of the conclusions that can be attained from them are presented further down in the paper.

Tracking and collecting educational data is one of the ways to improve learning experience and create learning content that is exhaustive and straightforward [11]. Gathering this data can help pinpoint parts of the material, problematic for the students to apprehend. Its various entity requires a standard for interoperability and storage and currently the experience API is one of the widely used ones in e-learning.

## C. What is experience API?

xAPI is a well-known specification for standardizing and collecting formal and informal educational data in the form of statements [12]. Each statement describes a learning experience – a user action and all of the interconnected objects to it in a given context, and so enables monitoring of the learning process [13]. The statements are formatted in JSON and can be easily read both by a human or a machine. They consist of the following elements: actors – user or a group performing the activity; verbs – the performed action (e.g. started, answered, completed, etc.), object – the object of the activity (it can be one of the types activity, actor or statement) and context properties – describing the context in which the action took place. There aren't any limitations about where the experience has occurred (it is not restricted to a specific learning environment or application). Those statements are logged in a learning record store and later on they can be previewed and further analysed. An LRS represents a repository for storing learning records and is a requirement for the xAPI standard to work. It records the learning activity statements, generated by users' behavior [14].

## D. Analyzing data

Educational data mining is a modern and evolving discipline, concerned with the development of methods for exploring the unique data types coming from educational context [15]. It generally deals with the discovery of useful knowledge and mining patterns from educational systems like online learning management systems (Moodle,

Blackboard, etc.), content management systems, registration and admissions systems.

The actual data can be taken from university databases, e-learning systems, LRS and then analyzed to discover meaningful rules and patterns to provoke students to increase their performance or give institutes an overview of their learning process [16].

## III. METHODOLOGY

The main goal of our research is to design and develop a system for creation, support and tracking of interactive learning activities in the e-learning. We use the following methodology: review of previous works, creation of an approach for development, support and tracking interactive learning activities in an independent mobile environment, design of a software (of plug-in type) for converting learning activities from an e-learning environment into a traceable interactive EPUB learning content, development of the software prototype, integration of the plug-in in a particular e-learning environment, and testing the software with real learners.

The used working approach is to integrate a system (plug-in), that converts learning activities into interactive EPUB learning content, suitable for usage and tracking on any mobile device, within a e-learning environment. The paper presents all steps of our methodology, however, only summary of some results are shown, since the authors have already published on this topic. The special attention is done of the automatically generated reports from the students' activities during usage of the interactive e-books.

## IV. RESULTS

### A. Approach

Undeniable fact is that students prefer using mobile devices instead of personal computers, that is why the more commonly used, complex and heavy e-learning environments have to find some alternatives. In order to create, support and track interactive learning activities on any mobile device we decided to create a system of type "plug-in", that will export our learning content (course) from the university's e-learning environment Moodle to an e-book. The plug-in converts all Moodle learning activities into interactive EPUB learning content, where student's learning activities can be noted and collected in order to be tracked by the teachers.

After exporting the course from Moodle and getting the enriched e-book, the following approach is taken to perform student tracking (Fig. 1). There will be a separate e-book created for every student downloading the course, so all interactions on the e-book performed by them are sent to a learning record store with xAPI statements, specifically tailored to reflect their account data. Optionally the same LRS can be connected to other systems, integrating xAPI, and actions performed in them can also be collected in the same place with the same xAPI profile statement templates. This ensures that all of the students learning, without restrictions of the place where it happened, will be recorded (no matter if the users are logged in Moodle and going thru the course modules or have downloaded the e-book and complete the activities there, statements will be sent and their actions recorded).

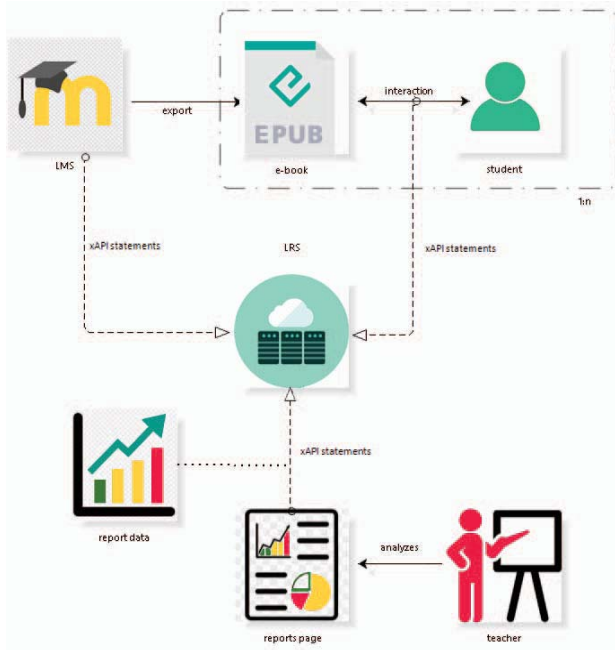


Fig. 1. The process of exporting a course to an e-book and gathering xAPI statements for reporting

Once the xAPI statements are in the LMS, they can be retrieved and processed for different purposes. The information, coming with the statements, can be limited and grouped based on the current report to reflect its specific requirements. The model for tracking students' performance suggests that a report page in Moodle is present, that will request data from the LRS and build a set of reports, which will be then analyzed by the teacher.

### B. Collecting Data

A Moodle course usually consists of lessons, which on their hand include resources (reading materials, links, images, etc.), activities (tests, assignments, etc.) and social elements, encouraging interaction and collaboration between students [17]. Users accessing and going thru different parts of the course and then working on the according modules defines the users learning behavior. Whether a lesson will be read for an hour or all of the assignments will be completed, this all reflects the students' motivation for learning and shows if the final assessment score is deserved or not.

Recording learners' behavior is done thru a variety of xAPI statements, sent to the LRS. Determining which user actions will be documented and what data will be sent while interacting with the lesson is crucial to provide enough information for complete and sensible reports. Five methods are recommended in [18] to improve those statements by storing achievement types, the duration of the learning experience, experience source information, domain learning and forgetting and assessment scores. All of these statements and their corresponding verbs, extensions and contexts are documented into an xAPI profile [19]. Using authored profiles guarantees a semantically interoperable development and successful storage of data that can be used multiple times for different use cases.

One way of implementing the functionality to send statements from an e-book is by using a javascript file, called xAPIWrapper [20]. It simplifies the connection and communication to LRS and let the developers focus on the

actual data. Combining this with another set of javascript classes – xAPIDashboard [21] gives you all the tools to create charts and tables from the stored statements, serving different purposes.

Table I shows a combined collection of identifiers from a set of xAPI profiles (activity-streams, adl, tincan [22]), used to describe all of the performed students' interactions on the e-book. In other words, the profiles describe the learning activities (activity type), the additional information (extensions), needed later for generating reports, and the performed actions (verbs) by students on the e-book.

They can be further classified into three smaller groups – general identifiers (can be used in all of the course modules, capturing behavior such as viewed a resource, completing a resource, etc.), quiz identifiers (used in capturing interactions with the quiz module like answering a question, scoring a question and passing or failing the assessment) and lesson identifiers (used to record students going thru lesson pages and answering to question pages).

For example, a student answering on a multichoice question can be represented with xAPI in the following way (Fig. 2). The statement consists of id, version, timestamp, authority, stored, verb, actor, result and context sections. The id is its unique identifier.

TABLE I. XAPI PROFILE FOR MOODLE COURSE INTERACTIONS

Name	Kind	URI
assessment	activity type	http://adlnet.gov/expapi/activities/assessment
cmi.interactions	activity type	http://adlnet.gov/expapi/activities/cmi.interaction
course	activity type	http://adlnet.gov/expapi/activities/course
lesson	activity type	http://adlnet.gov/expapi/activities/lesson
lms	activity type	http://id.tincanapi.com/activitytype/lms
page	activity type	http://activitystrea.ms/schema/1.0/page
duration	extension	http://id.tincanapi.com/extension/duration
answered	verb	http://adlnet.gov/expapi/verbs/answered
completed	verb	http://adlnet.gov/expapi/verbs/completed
experienced	verb	http://adlnet.gov/expapi/verbs/experienced
started	verb	http://activitystrea.ms/schema/1.0/start
viewed	verb	http://id.tincanapi.com/verb/viewed

The timestamp and stored fields have the time when the learning activity took place and when it was stored on the LRS. The verb is defined by a unique id - http://adlnet.gov/expapi/verbs/answered and a more readable display text - answered. The actor has a type - Agent and can be identified in this case solely by the mail address - mbox. The object field has object type (objectType: Activity), activity type (type: http://adlnet.gov/expapi/activities/cmi.interaction) and interaction type of the question (interactionType: matching). The pairs of elements to be matched are in the source and target fields accordingly. The result section contains data about the given response to the question and if it's been completed or not. The LMS, the course and the quiz from which the question is a part of can be found in the context section (as grouping and parent).



```

{
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/answered",
    "display": {
      "en-US": "answered"
    }
  },
  "version": "1.0.0",
  "timestamp": "2019-10-28T08:33:07.475820+00:00",
  "object": {
    "definition": {
      "source": [
        {
          "id": "4",
          "description": {
            "en-US": "student"
          }
        },
        {
          "id": "6",
          "description": {
            "en-US": "grade"
          }
        },
        {
          "id": "5",
          "description": {
            "en-US": "person"
          }
        }
      ],
      "interactionType": "matching",
      "type": "http://adlnet.gov/expapi/activities/emi.interaction",
      "name": {
        "en-US": "Match the following objects with their types:"
      },
      "target": [
        {
          "id": "4",
          "description": {
            "en-US": "class"
          }
        },
        {
          "id": "6",
          "description": {
            "en-US": "property"
          }
        },
        {
          "id": "2",
          "description": {
            "en-US": "parent class"
          }
        }
      ],
      "id": "http://moodle.slims.eu/question/question?cmid=3&id=7",
      "objectType": "Activity"
    },
    "actor": {
      "mbox": "mailto:user1_email_address",
      "objectType": "Agent"
    },
    "stored": "2019-10-28T08:33:07.475820+00:00",
    "result": {
      "duration": "PT44S",
      "completion": true,
      "response": "{\"4\":\"4\",\"6\":\"5\",\"5\":\"6\"}"
    },
    "context": {
      "platform": "e-book",
      "contextActivities": {
        "parent": [
          {
            "definition": {
              "type": "http://adlnet.gov/expapi/activities/assessment",
              "name": {
                "en-US": "Exam - Inheritance"
              },
              "description": {
                "en-US": "This test will determine the student's knowledge about object inheritance in programming."
              }
            },
            "id": "http://moodle.slims.eu/mod/quiz/view.php?id=3",
            "objectType": "Activity"
          }
        ]
      },
      "grouping": [
        {
          "definition": {
            "type": "http://adlnet.gov/expapi/activities/course",
            "name": {
              "en": "Inheritance"
            }
          },
          "id": "http://moodle.slims.eu/course/view?id=3"
        }
      ],
      "definition": {
        "type": "http://id.tincanapi.com/activitytype/lms",
        "name": {
          "en": "FMI Moodle"
        }
      },
      "id": "http://moodle.slims.eu"
    }
  },
  "id": "c8ce054e-3ced-460f-b6a7-1086d43a7b37",
  "authority": {
    "mbox": "mailto:user1_email_address",
    "name": "user1_name",
    "objectType": "Agent"
  }
}

```

Fig. 2. Example xAPI statement for test interaction

When creating the e-book, alongside with the recreated Moodle content using HTML and CSS, JavaScript code is added whenever an interaction needs to be described. Additionally, JS is used for trigger points for xAPI statements. Same technology is used for the report page. The JavaScript classes xAPIWrapper and xAPIDashboard are added to simplify the connection to the LRS and to help gathering and displaying its data in a visually appealing manner – in the form of graphs, bar charts and line charts. The queries retrieving the statements are similar to SQL and xAPIDashboard provides data manipulation before and after the requests are sent over to the storage server, making the extraction only of necessary information for the current chart.

### C. Reports for teachers

The experiment was conducted with real users – 21 students in the course “Object-oriented design and programming” at Faculty of Mathematics and Informatics of the University of Plovdiv “Paisii Hilendarski”, Bulgaria. The reports presented below are shortened and with altered student's personal information, in order to simply showcase the available report types.

xAPI statements by themselves do not lead to big understanding or knowledge when read by the teacher. Converting them to information that is meaningful and can be easily analysed is what provides tools to monitor students' behaviour and find similar behaviour amidst the groups of students [23]. The stored learning activities statements in the LRS can be used to create charts and user metrics, thus different reports can be made. They each show different parts of user interaction with the learning content and overall commitment of the users' participation and willingness to study the material.

Several report categories can be differentiated in accordance to the data displayed:

- **course** reports: they can show the ratio between the count of people who have started a course, completed a course or left it unfinished. Also, another metric included in the graphs is the users' average total time spent on a course (Fig. 3). The report is good to determine courses that are too extensive and should be split into several smaller ones or courses, which started to completed ratio is low – meaning they couldn't keep the students interest at all.

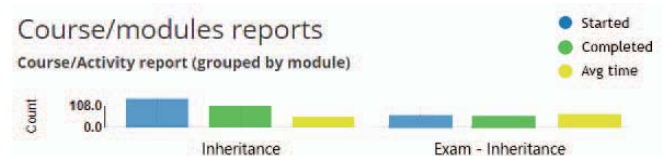


Fig. 3. Preview of a course report

- **quiz** reports: divided reports for every question constituting the quiz in the form of how many users answered the same (e.g. choosing answer A on question 3 from the test) or the average time it took the users to progress through a question. Another possible report is to group the users by their quiz score. That way the number of users with an excellent, middle or low score are visible.

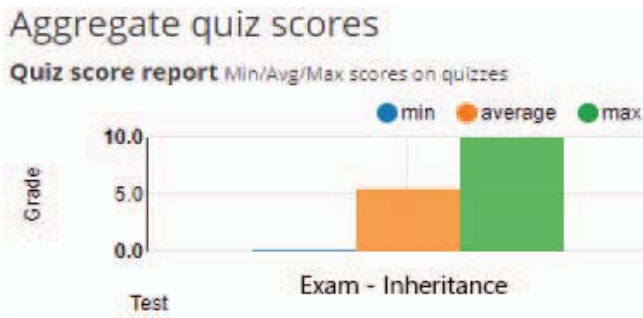


Fig. 4. Graph with the min/avg/max score of quizzes

Fig. 5 displays a graph with all of the students' scores, divided in ranges (five ranges are available – from scores 1 to 3, 3 to 5, 5 to 7, 7 to 9 and 9 to 10). This is useful to quickly check how students have scored on the test and what is its difficulty.

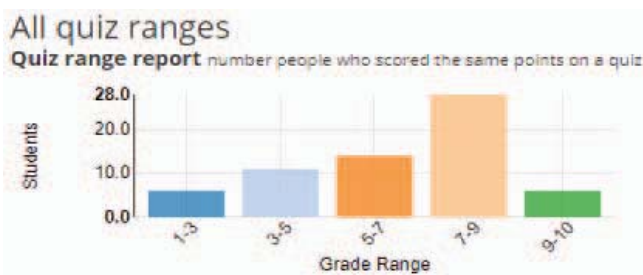


Fig. 5. Graph with the number of students with similar result scores (grouped by ranges)

Every bar corresponds to the number of students having a score in the concrete range. More exhaustive data can be viewed with the next report at Fig. 6, showcasing all students' emails who have scores in a given score range.

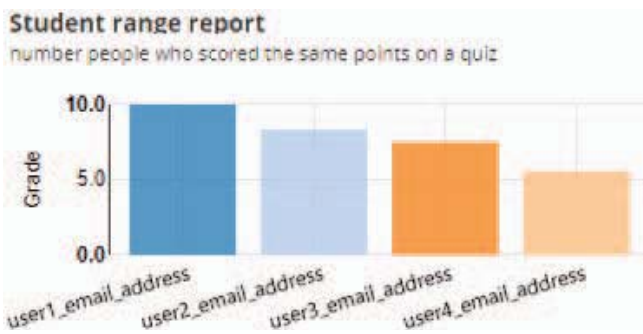


Fig. 6. Graph showing all students in a certain score range (from 5 – 10)

Reports can be made for individual quizzes too. A student's answers for an exact test can be gathered and whether they are correct or not. On Table II the questions are taken from the *object definition* property in the xAPI statement and the correctness of the student's answer from the *result success* property.

- **assessment** reports: these reports show crucial points in the assessment of the students – questions that got the most wrong answers from the students (Fig. 7), questions that took the most time to be answered (Fig. 8), meaning that they are too hard or written not that well.

TABLE II. STUDENTS' ANSWERS FOR ALL QUESTIONS IN A QUIZ

All student answers	
object.definition.name.en-US	result.success
How would you declare in C++ the elements: student, person, grade?	true
All the methods in every inheritance class are predefined?	false
How many properties does the class Motion have?	true
What is the constructor called in class Circle?	false
Which is the correct code for writing a constructor?	true
Match the following objects with their types:	false

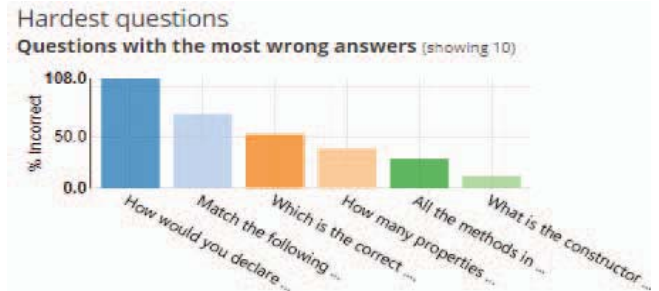


Fig. 7. Graph with the questions with most wrong answers from students

Further reports can be made about which students have answered wrong on a question. On Table III there are two columns: one with the students' email (email can be found under the *actor mbox* property on the xAPI statement) and one with the result of the answer.

TABLE III. STUDENTS, THAT ANSWERED WRONG ON A GIVEN QUESTION

Students answered wrong on the question on their last attempt	
actor.mbox	result.success
user1_email_address	false
user2_email_address	false

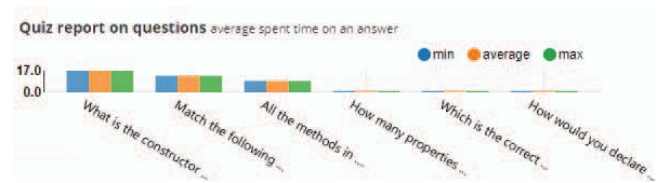


Fig. 8. Graph with the min/avg/max time spent on questions

- **leaderboard** reports: give an overall look of which user performed the best on tests (achieved top scores on a quiz), submitted an assessment for the fastest time or which user had the most logins or spent the most time on a course section;

On Table IV a list with the top 25 of all students and their test scores is presented, sorted in a descending order. The email is again taken from the *actor mbox* property and the total points on the test are taken from the *result score raw* property of the xAPI statements.

TABLE IV. LEADERBOARD DISPLAYING THE STUDENTS WITH THE HIGHEST GRADE ON A QUIZ

Leaderboard scores - table with top scorers on a quiz (agent – score)	
actor.mbox	result.score.raw
user1_email_address	10.00
user2_email_address	8.33
user3_email_address	7.50
user4_email_address	5.56

Leaderboards can show the students who have managed to complete a quiz in the shortest time – Table V. To accomplish this an extension property on the xAPI statement needed to be defined displaying the time a student had taken completing a quiz. This property is already available in the TinCan xAPI Profile: *result\_extensions* (<http://id.tincanapi.com/extension/duration>).

TABLE V. LEADERBOARD WITH THE FASTEST COMPLETION OF A QUIZ

Leaderboard times - table with time of fastest quiz completion	
actor.mbox	result.extensions [http://id.tincanapi.com/extension/duration]
user1_email_address	44.03

- **learner’s path** reports: displays all of the stored xAPI statements for a certain user, thus tracking all of the actions performed by him/her and their sequence, giving a general picture of his/hers exact steps of completing the course;
- **engagement** reports: combine stat measurements like course views, quiz submits and time spent on modules to present how much the users were involved in the course activities;
- **effectiveness** reports: a user is evaluated before taking the course and afterwards. The difference in results of both evaluations shows whether the course is effective;
- **time period** reports: usually a time period gets selected (month, week, etc.) and a specific metric is evaluated (time spent on a course, users finished a quiz, etc.). Some examples include the most frequent xAPI statement verbs (Fig. 9), the total count of xAPI statements collected for a given date (Fig. 10) and the count of activities, performed by the same student, grouped by dates (Fig. 11).



Fig. 9. The most popular action, taken by students in the past month



Fig. 10. The count of students’ activities by days in the past month



Fig. 11. The actors with the most activity in the past month

## CONCLUSIONS

E-books can be found everywhere in current learning processes. Adding interactivity and tracking to them give rich possibilities to learners in terms of interaction and involvement in the learning material. E-books are a rich enough platform to track simple interactions and provide interesting feedback. We have shown how equivalent can be to perform the same activity in a traditional learning environment and in an e-book. If not fully functional as Moodle and other platforms, e-books are simple and ubiquitous, and can get enough interaction and tracking to analyze student performance and provide guidance and help.

This paper observes the ways to track students learning behavior and the possibilities to record it in an easily accessible and usable place. Then a simple categorization of the applicable reports with that data is presented and examples of their implementation in a report page.

The research can be extended by adding to the pool of interactivity elements. Adding comments, collaborative assessment and workshops are further functionalities that have the potential to be implemented. Collecting data from the e-books is a prerequisite to replicate the learners’ paths and create charts that analyzed additionally by the teachers shows the weak points in the learning material and if teachers are willing to change it – make better content for studying.

## ACKNOWLEDGMENT

We thank Madrid Regional Government for the support of the project e-Madrid-CM (P2018/TCS-4307). e-Madrid-CM project is co-financed by the Structural Funds of the European Union (FSE and FEDER).

## REFERENCES

- [1] M. Montebello, “e-Learning so Far. In: AI Injected e-Learning. Studies in Computational Intelligence,” *Springer*, vol 745, 2018.
- [2] "EPUB 3 Overview. Final Community Group Specification 08 May 2019," [Online]. Available: <https://www.w3.org/publishing/epub/epub-overview.html>. [Accessed 12 12 2019].
- [3] B. Leporini and C. Meattini, "Personalization in the Interactive EPUB 3 Reading Experience: Accessibility Issues for Screen Reader Users," pp. 1-10, 2019.
- [4] D. Hawkins, "Electronic Books: A Major Publishing Revolution. Part 1: General Considerations and Issues," vol. 24, no. 4, 2000.
- [5] M. Landoni and G. Hanlon, "E-book reading groups: interacting with e-books in public libraries," *The Electronic Library*, vol. 25, no. 5, pp. 599-612, 2007.
- [6] E. Morgan, "Electronic books and related technologies," *Computers in Libraries*, vol. 19, no. 10, pp. 36-39, 1999.
- [7] D. Carvajal, "Racing to convert books to bytes," *The New York Times*, 1999.

- [8] D. Prasetya, A. Wibawa and T. Hirashima, "An interactive digital book for engineering education students," *World Transactions on Engineering and Technology Education*, p. 16, 2018.
- [9] A. Parasuraman, "Technology readiness index (TRI) a multiple-item scale to measure readiness to embrace new technologies," *J. of Service Research*, vol. 2, no. 4, pp. 307-320, 2000.
- [10] R. Vhalery, Nofriansyah, A. Martiah, "Cooperative Learning in the Learning Activity of Students," 2018.
- [11] A. Noura, L. Cheniti-Belcadhi and R. Braham, "An Enhanced xAPI Data Model Supporting Assessment Analytics," *Procedia Computer Science*, vol. 126, pp. 566-575, 2018.
- [12] J. Kevan and P. Ryan, "Experience API: Flexible, Decentralized and Activity-Centric Data Collection," *Tech Know Learn*, vol. 21, no. 143, 2016.
- [13] "Student's Academic Level Evaluation," [Online]. Available: <http://www.lib4dev.in/info/nikhildutta333/student-performance-prediction/184567131>. [Accessed 10 12 2019].
- [14] "Learning Record Store," [Online]. Available: [https://en.wikipedia.org/wiki/Learning\\_Record\\_Store](https://en.wikipedia.org/wiki/Learning_Record_Store). [Accessed 12 12 2019].
- [15] C. Romero, S. Ventura and E. García, "Data mining in course management systems: Moodle case study and tutorial," *Computers & Education*, vol. 51, pp. 368-384, 2008.
- [16] S. Fatima, I. Siddiqui and Q. Ali, "Analyzing Students' Academic Performance through Educational Data Mining," *3C Tecnología. Glosas de innovación aplicadas a la pyme*, 2019.
- [17] W. Rice, "Moodle 2.0 E-learning Course Development: A Complete Guide to Successful Learning Using Moodle," *Packt Publishing Limited*, 2011.
- [18] R. Sottolare, R. Long and B. Goldberg, "Enhancing the Experience Application Program Interface (xAPI) to Improve Domain Competency Modeling for Adaptive Instruction," pp. 265-268, 2017.
- [19] "xAPI profiles," [Online]. Available: <https://github.com/adlnet/xapi-profiles>. [Accessed 12 12 2019].
- [20] "xAPIWrapper," [Online]. Available: <https://github.com/adlnet/xAPIWrapper>. [Accessed 12 12 2019].
- [21] "xAPI-Dashboard," [Online]. Available: <https://github.com/adlnet/xAPI-Dashboard>. [Accessed 12 12 2019].
- [22] "Repo for xAPI authored profiles," [Online]. Available: <https://github.com/adlnet/xapi-authored-profiles>. [Accessed 12 12 2019].
- [23] A. Bibiloni, A. Oliver and J. Molino, "Automatic collection of user behavior in 360° multimedia. Multimedia Tools and Applications," vol. 77, 2017.