



TEI of Crete
Technological Educational Institute of Crete

UBIQUITOUS LEARNING WITH EXPERIENCE API

by

KORALIA PAPADOKOSTAKI

Bachelor in Informatics, Athens University of Economics and Business, 2003

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

DEPARTMENT OF INFORMATICS ENGINEERING

SCHOOL OF ENGINEERING

TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CRETE

2017

Approved by:

Assistant Professor
Spyros Panagiotakis

Statement of Originality

The work contained in this thesis has not been previously submitted for a degree or diploma at any other higher education institution or any other purpose. To the best of my knowledge, the thesis contains no material previously published or written by another person except as specified in references. I certify that the intellectual content of this thesis is the product of my own work and all the assistance received in preparing this thesis and sources have been acknowledged.

Koralia Papadokostaki

Abstract

It is without doubt that the current major and popular technological achievements- Internet, social media and mobile devices- influence dramatically current trends in education and learning. With the rapid expansion of broadband and Wi-Fi connections and the explosion of Web 2.0, learning has evolved to e-learning and mobile learning (m-learning) and is no longer limited to classrooms or books. Learning can nowadays be achieved through YouTube videos, Facebook posts or Khan Academy, part of a vision called Ubiquitous learning, where learning may happen everywhere, anytime and in any way. However, since the content is distributed in various forms, platforms and hosts, the need to collect information for a user, in order to watch his performance, form his learning profile and offer him the most suitable content has arisen. In order to realize Ubiquitous learning and produce valuable conclusions in the form of Learning Analytics, a new specification that tracks the user activities has been developed, the Experience API (xAPI). With the latter, personalized and more efficient learning can be provided, since e-learning can focus on adaptivity, i.e. the adaptation of the learning content to meet a learner's specific needs, match his learning background or personal interests.

Towards these directions we have created an innovative LMS- independent, adaptive learning system that, with the use of Experience API , tracks users' actions and scores over various educational resources, adapts its content according to the user's previous actions and provides the instructor with a powerful dashboard to inform him upon his class overall and student's personal performance. Our learning system contains two stand-alone courses that interact with each other seamlessly, and is augmented with extra multimedia content which can be distributed across different hosts. It changes dynamically the content based on the user's previous activities and tracks activities from diverse sources. Finally, it comes with powerful Learning Analytics tools to help the instructor evaluate the learning process, the content provided and the performance of the students. Our implementation is 'platform-agnostic', totally independent of Learning Management Systems and contributes to the monitoring of ubiquitous learning, where learning is not only formal, but part of our everyday life.

Table of Contents

Statement of Originality.....	ii
Abstract.....	iii
Table of Contents.....	iv
List of Figures	vii
List of Tables	x
Acknowledgements.....	xi
Preface	xii
Chapter 1 - Introduction	1
1.1 Overview	1
1.1.1 Sequencing.....	2
1.1.2 Adaptivity.....	3
1.1.3 The transformation of e-Learning to Ubiquitous learning	5
1.1.4 Learning Analytics - a new promising field	8
1.2 Scope and Objective	9
1.3 Outline of Thesis	10
1.4 Publications.....	11
Chapter 2 - Overview of the xAPI Ecosystem	12
2.1 Related specifications	12
2.2 ADL's SCORM: the predecessor of Experience API.....	13
2.3 Inside the Experience API specification	14
2.3.1 Activity Streams	14
2.3.2 Activity Providers in xAPI	15
2.3.3 Learning Record Store (LRS).....	15
2.3.4 Activity Consumers in xAPI	15
2.3.5 Interaction between LMS and LRS.....	16
2.3.6 Vocabulary	17
2.3.7 An example of an xAPI statement from our implementation.	21
Chapter 3 - Related work in xAPI	23
3.1 Applications of xAPI	23
3.1.1 Applications of xAPI in Education	23

3.1.2	Applications of xAPI in Training	27
3.1.3	xAPI in Defense	28
3.2	Developers' tools for xAPI.....	28
3.2.1	ADL's LRS	28
3.2.2	The JavaScript Library tincan.js.....	28
3.2.3	The xAPI wrapper	29
3.2.4	The xAPI Dashboard	29
3.2.5	xAPI-youtube.....	29
3.3	Research projects on xAPI	29
Chapter 4 - Technical Implementation		31
4.1	LRS.....	31
4.2	Learning Content.....	33
4.3	Teacher's Dashboard	34
Chapter 5 - Learning Content.....		36
5.1	An LMS- independent Adaptive Learning System	37
5.1.1	Overview	37
5.1.2	Structure	40
5.1.3	Code	45
5.2	A video activity tracker	50
5.2.1	Overview	50
5.2.2	User Interface	50
5.2.3	Code	52
5.3	H5P xAPI-enabled content	53
5.3.1	Overview	53
5.3.2	H5P content	55
5.4	A sequencing template	58
5.4.1	Overview	58
5.4.2	User's interface	60
5.4.3	Code	62
Chapter 6 - Teacher's Dashboard		70
6.1	ADL LRS interface	71
6.2	xAPI statement viewer	71

6.2.1	Overview	71
6.2.2	User Interface	72
6.2.3	Code	73
6.3	Visualization	74
6.3.1	Overview	74
6.3.2	Code	82
6.4	WordPress H5P Monitoring Tool	85
Chapter 7 - Evaluation		86
Chapter 8 - Conclusions		99
References		104

List of Figures

Figure 1.1 Learning content without sequencing vs. learning content with sequencing (Figure taken from [6]).	3
Figure 1.2 Comparisons and flow of e-learning, m-learning and u-learning (Figure taken from: [19]).	6
Figure 1.3 Comparison of Frameworks for Learning Analytics (Taken from: [22]).	8
Figure 2.1 An Activity Provider interacts with an LRS which in turn interacts with an Activity Consumer (Figure taken from [37]).	16
Figure 2.2 A standalone LRS interacting with various activity providers. (Figure taken from: [40]).	16
Figure 2.3 The LRS is incorporated in the LMS (Figure taken from: [40]).	17
Figure 2.4 Multiple LRS (Figure taken from: [40]).	17
Figure 3.1 Some of the adopters of Tin Can API (Figure taken from [41]).	23
Figure 4.1 The Architecture of our System	31
Figure 4.2 The details of our account from our LRS	32
Figure 4.3 Our LRS started successfully and listens to 83.212.100.157:8000	33
Figure 5.1 The learning content of our implementation	36
Figure 5.2 The architecture of our implementation.	38
Figure 5.3 Interactive content of our implementation: An interactive video and images.	38
Figure 5.4 : Different types of assessment: Multiple choice questions.	39
Figure 5.5 : Different types of assessment: Fill in the blanks/ Hotspot Area	39
Figure 5.6 : Interactivity: The instructor changes face when he sees a wrong answer.	39
Figure 5.7 : Interactivity: The results and score of each unit	40
Figure 5.8 An example of the delivery of the statement from Course 1 to the LRS.	41
Figure 5.9 An example of the request of a statement towards the LRS.	42
Figure 5.10 The user has previously completed Course 1 and automatically skips the slides concerning this content. In the menu on the left, the slides concerning the content of Course 1 appear to have been skipped.	42
Figure 5.11 If the user has not completed Course 1, he watches the slides concerning relevant content.	43

Figure 5.12 If the user has not passed the quiz of a unit, he watches the slides concerning the specific unit.....	43
Figure 5.13 If the user has passed the quiz of a unit, he is transferred to the next unit.	44
Figure 5.14 The architecture of the main Course 2.	44
Figure 5.15 A statement is sent to the LRS when the user clicks on the palette.	45
Figure 5.16 The architecture of our video activity tracker.	50
Figure 5.17 The welcome page of our video-tracking (index.html)	51
Figure 5.18 The page with the video-tracking script (Video-player.html).....	51
Figure 5.19 The statements sent to the LRS from our video-tracking app.....	52
Figure 5.20 The H5P plugin for WordPress.....	54
Figure 5.21 The results of registered users in WordPress for H5P content.	54
Figure 5.22 Configuration of wp-h5p-xapi plugin.....	55
Figure 5.23 A simple question on the content of the video	56
Figure 5.24 A question with adaptive branching.....	56
Figure 5.25 The interaction of the question	56
Figure 5.26 The user is redirected to the part which he did not understand.	57
Figure 5.27 The summary quiz shows progress.....	57
Figure 5.28 After the completion of the summary quiz, the total score is shown.....	57
Figure 5.29 The template's diagram (Figure taken from: [6])	58
Figure 5.30 The template's rules (Figure taken from: [6]).....	59
Figure 5.31 The flowchart of our implementation	60
Figure 5.32 The User's interface for the Sequencing Template	61
Figure 5.33 The interaction of the user with the Sequencing Template.....	62
Figure 6.1 The Teacher's Dashboard for our Implementation	70
Figure 6.2 The interface of ADL LRS for the administrator.....	71
Figure 6.3 The modified xAPI Statement Viewer.....	72
Figure 6.4 The Report for the LRS (Activity Per user bar chart).....	74
Figure 6.5 The Report for the LRS (Frequency of verbs bar chart).	75
Figure 6.6 The Report for the LRS (Distribution of score).....	75
Figure 6.7 The Table with the average score per Activity.	76
Figure 6.8 The Barchart with the average score per Activity.	76
Figure 6.9 The Report for the Activities (Average Score bar and pie chart).....	77

Figure 6.10 The Report for the Users (Average Score per User).	78
Figure 6.11 The Report for the Users (distribution per score pie chart and table).	78
Figure 6.12 The Report for the Users (Progress of user).	79
Figure 6.13 The Report for the Users that passed (table with aggregate data).	79
Figure 6.14 The Report for an action regarding the H5P content.	80
Figure 6.15 The Report for the Users that interacted with the xAPI-enabled video	80
Figure 6.16 The Report for the verbs regarding an xAPI-enabled video	81
Figure 6.17 The Report for the Users that interacted with our Sequencing Template and specifically with OBJ_1.	81
Figure 6.18 The Chart showing the frequency of 'passed' vs 'failed' in OBJ_1	82
Figure 7.1 The activity for every user of our evaluation	86
Figure 7.2 The frequency of actions for our evaluation	87
Figure 7.3 The score achieved in several statements of our evaluation	87
Figure 7.4 The average score for every user of our evaluation	88
Figure 7.5 The score for each statement of the user (pie chart and table).	88
Figure 7.6 The progress of a user of our evaluation	89
Figure 7.7 The minimum, maximum and average score of the user	89
Figure 7.8 The average score for every activity of the course (table and barchart)	90
Figure 7.9 The illustration of the score achieved for the 'Eisagwgi_sto_excel' part	91
Figure 7.10 The illustration of the score achieved for the first quiz 'Eisagwgi_sto_excel_PART_A'	92
Figure 7.11 The illustration of the score achieved for the first quiz 'Eisagwgi_sto_excel_PART_B'	93
Figure 7.12 The illustration of the score achieved for the first quiz 'Eisagwgi_sto_excel_PART_CD'	94
Figure 7.13 The illustration of the score achieved for the first quiz 'Eisagwgi_sto_excel_PART_E'	95
Figure 7.14 Only two students clicked on the Coloring Palette	97
Figure 8.1 xAPI supports a distributed architecture, where statement streams can originate from diverse sources and may be delivered to several endpoints.	99
Figure 8.2 The ubiquitous vision of xAPI (Figure taken from [35])	100

List of Tables

Table 2.1 A selection of Actions of Tin Can API.	19
Table 2.2 A selection of Activities of Tin Can API.....	20
Table 3.1 Support of Tin Can API from popular LMSs (LMSs in alphabetical order).	24
Table 3.2 Support of Tin Can API from popular authoring tools.	25
Table 3.3 Case Studies in Learning.....	25
Table 3.4 Case Studies in Training.	27
Table 7.1 The users that passed http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E	95
Table 8.1 Differences between SCORM and xAPI	101

Acknowledgements

I would like to express my deepest appreciation and my sincere gratitude my Supervisor, Dr Spyros Panagiotakis for his constant support, his endless willingness to help and his warm encouragement through the long way of completing this thesis. Special thanks to Dr Kostas Vassilakis for his precious guidance, interesting suggestions and useful remarks.

I am also heartily thankful to my life - companion, Nikos, for showing incredible patience, infinite support and encouragement and for all the sacrifices he went through during my Master Degree Course. Without him, this endeavor would not have been possible. My deepest apologies to my family for all the precious moments we missed from each other due to my devotion to my Master thesis.

Preface

Who is there that does not always desire to see, hear, or handle something new?

To whom is it not a pleasure to go to some new place daily, to converse with
someone, to narrate something, or have some fresh experience?

In a word, the eyes, the ears, the sense of touch, the mind itself, are, in their search
for food, ever carried beyond themselves;
for to an active nature nothing is so intolerable as sloth.

(Comenius)

Chapter 1 - Introduction

1.1 Overview

Learning has changed significantly with the evolution of technology and grants new options, chances and potential under the term 'e-learning'. Borders and geographical locations have been nullified, the cost has diminished significantly and learners may take a course according to their own needs and capabilities and at their own pace. Learning has exited the traditional classes and occurs in virtual classrooms or asynchronous courses, such as MOOCS (Massive Open Online Courses), which have opened new horizons to those who want to expand their knowledge or expertise in a learning field in distant, massive courses. Electronic classes host the learning material, offer online assessments and personalized learning paths and allow for online collaboration between the learners. E-learning offers access to disabled people or people with limited time and gives vast possibilities to people who want to expand their knowledge. However, as learners differ in many ways, *adaptivity* in learning is a key issue in making learning more effective, since a "one-suits-all" solution is not ideal for learners with different background, objectives and needs. Towards this direction, and as an adaptive learning system may end up complex, the design of learning content is critical for its reusability and cycle of life. Learning content needs to be divided into small units and interaction between them should be defined in detail. Such an example of interaction between minimal learning units is *sequencing*, which refers to the organization of the learning content and the navigation between them. Adaptivity and Sequencing will be addressed later in this section.

In the mobile era we live in, e-learning has evolved from online courses and Learning Management Systems (LMSs) to social networking, online sources, such as YouTube or Khan Academy and can now take place literally everywhere with the use of powerful mobile devices. Moreover, the importance of informal learning, i.e. learning that happens out of the 'formal' education, has been identified and the need for lifelong learning is a new challenge in a rapidly changing world. The shift from traditional learning to Ubiquitous learning is a new emerging paradigm that will be later discussed in this chapter. In this context, as any instructor might want to record his students' progress over a course, so

might a provider of content want to trace the impact of his content to the public. With interaction between the content, the user and the provider several conclusions can be drawn, the content may be improved and the user can acquire self- confidence. But as e-learning changes and shifts to mobile devices, and from formal learning to informal learning, tracking such interactions is not always feasible. The vision of tracking activities, such as reading an e-book, watching an online video or sharing an online article has arisen in our mobile era. The need to capture such learning activities, *seamlessly*, without the interference of the user, in order to form learning profiles, evaluate learning content and modify or adapt it according to user's needs, background and capabilities is now more evident than ever. Tracking interactions may also provide important data for the production of Learning Analytics, i.e. to provide powerful conclusions about the learning procedure. Learning analytics is a new field of Analytics that focuses in learning and will be shortly presented later in this section.

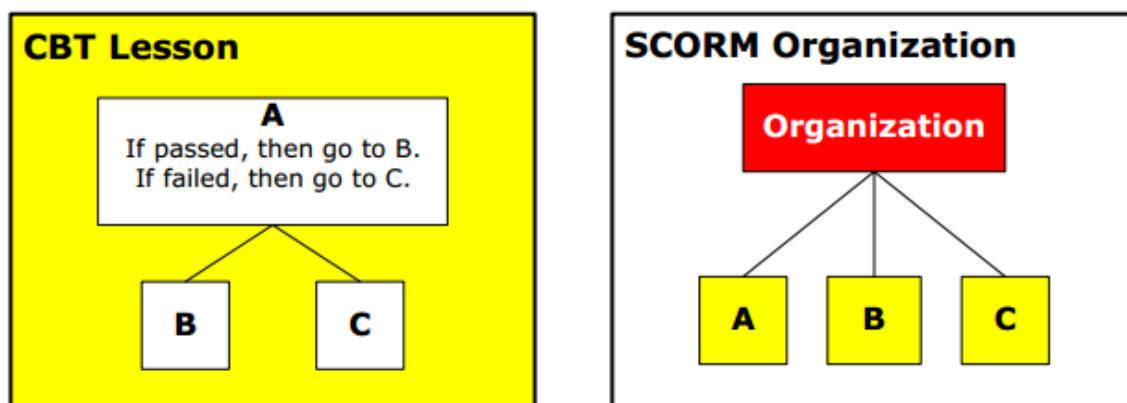
1.1.1 Sequencing

The ADL (Advanced Distributed Learning) Initiative [1] is a US government program aiming to augment flexible, lifelong learning through the use of technology [2]. It is widely known for its SCORM specification, introduced in 2001, and revised up to the SCORM 2004 4th Edition [3]. SCORM (Sharable Content Object Reference Model) was designed to overcome the major problems of interoperability and reusability of learning content. Before SCORM was proposed, the process of tracking the learner's progress was tailor-made for each platform; if the company or foundation changed its Learning Management System (LMS), the tracking process had to be redesigned and re-implemented. With the use of the SCORM model, the learning content is packaged into a format which can be transferred through various LMSs [3]–[5] accomplishing thus not only interoperability, but also reusability, traceability and longer lifecycle.

SCORM specification introduces SCO, a Shareable Content Object (SCO) as a launchable learning object (resource) that communicates with the run-time environment that launched it. It represents the lowest level of granularity of a learning resource and should be reusable and independent of its learning context. To achieve such reuse a SCO should be "self-contained" and contain no reference or link to other SCOs. SCORM not only provides a

specification for learning content, but also introduces good practices in learning design, that are desirable when designing a course. SCORM Sequencing defines the behaviors and data model used by the SCORM run-time environment to determine how a content package will be delivered as a learning experience [6]. It defines the functionality that a SCORM-conformant LMS must implement to process sequencing information based on learner interaction with content objects. Sequencing refers to the behaviors that an LMS or run-time system follows to deliver a specific learning experience as intended by an author or content developer. This learning experience may be free choice, in which the learner can choose any item in the content organization in any order, or it may be guided by flow through the structure of the content organization. SCORM sequencing implies that the navigation through the content is not conducted in the SCOs, but is done by the LMS or the learning system. This allows the reusability of the modules (SCOs) and their independence from the navigation decided by the content developer. The navigation and its options are implemented by the Learning Management system or an exterior module. Figure 1.1 illustrates sequencing, which is also a desirable feature when designing learning content.

Figure 1.1 Learning content without sequencing vs. learning content with sequencing
(Figure taken from [6]).



1.1.2 Adaptivity

Students in a traditional classroom come with a variety of traits, interests and background. Taking these into consideration and adapting teaching to meet the students' needs is effective and desirable [7], but not always feasible. In e-learning, personalized learning (i.e., tailored-learning according to the user's needs and capabilities) cannot be

accomplished, at a massive scale, using traditional approaches [8]. Therefore, the importance of adaptivity in e-learning has been identified and integrating adaptation in e-learning has been one of the goals of e-learning for a long time [7]–[9].

In order to improve learning and performance and to provide with a more individualized learning, the characteristics of a particular learner need to be correctly identified. Such characteristics include [7], [9], [10]:

- incoming knowledge, i.e. the prior knowledge each learner has
- cognitive abilities , e.g. memory or spatial conception
- learning styles, e.g. if one is more visual or audio learner
- skills , e.g. if one is more theoretical or practical
- personality traits, e.g. if one is social or lonely
- special needs/ disabilities, e.g. if one is hearing impaired
- demographic and sociocultural differences, e.g. the age of the user

Additionally, interests and needs of the learner should be analyzed in order to provide him with the most useful content. For e-learning to be most effective, we should capitalize on these characteristics when delivering content.

Embedded assessments can also improve e-learning, as they may be the tool for diagnosis of problems and for follow-up, e.g. to remediate on the topic, if the learner failed the assessment or supply him with extra material if he excelled in it and the content is among his interests [7].

With Adaptive Learning we should modify any individual student's learning experience [11], in order to deliver [7] :

- the right content
- to the right person,
- at the proper time,
- in the most appropriate way

Adaptivity in learning environments may be categorized as below [8]:

- *Adaptive Interaction*, where adaptation focuses on the user's interaction with the learning system through its interface, without affecting the content. Changing the size or colors of the learning system may be an example of such adaptation

- *Adaptive Course Delivery* is the most common type of adaptivity in e-learning and entails changing the content in order to meet the user's requirements and characteristics. This category includes: dynamic course structuring, adaptive navigation support and adaptive selection of content so that the learning result is maximized, with the minimum time and interactions.
- *Content Discovery and Assembly* uses adaptive techniques to track and assemble learning content from distributed sources / repositories.
- *Adaptive Collaboration Support* aims to implement adaptive support in learning processes that are collaborative and involve social interaction

Integrating characteristics from the aforementioned categories in a learning system should be achieved in order to meet the diverse requirements of several users.

1.1.3 The transformation of e-Learning to Ubiquitous learning

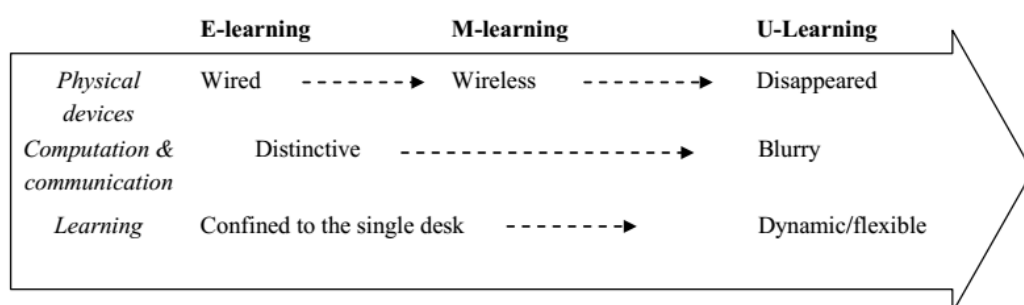
Instructors all over the world have always made use of technology in order to enhance the learning process and improve the learning impact. In fact, the main technological achievements of an era influence the educational theories and practices of that era [12]. With the revolution of computers, computer- enhanced learning rose as a modern and challenging trend. Over the last years, teaching is being aided by online simulations, electronic educational content, e-learning platforms and virtual classrooms. Distance learning has evolved to MOOCs (Massive Open Online Courses), changing the landscape of higher education forever. The use of technology in education is an asset instructors should definitely invest in, as students nowadays spend a lot of time using computers, tablets and the Internet and they are definitely “digital natives” [13]. We live in the era of ‘personal and technical mobility’ [14], where mobile devices, such as phones, tablets or MP3 players are carried everywhere. Due to their small size, light weight and always increasing features, together with the availability of connection to the Internet through Wi-Fi they become an integral part of our everyday lives and a powerful tool in our hands.

Research shows that use of mobile devices as a part of the learning procedure increases the motivation of the learners and urges them to engage more in the learning procedure [15], [16]. It is considered a pleasant experience, especially when combined with multimedia applications or games and seems to offer significant enhancement in student work.

Furthermore, the use of differentiated teaching methods is thought to stimulate the brains of the learners [17] and makes learning more enjoyable. Moreover, m-learning supports constructivism, a promising learning theory [16], which claims that learners discover knowledge through simulations or experiments and are not pathetic receivers. To continue with, m-learning can contribute to self-paced and self-directed learning, as learners can learn at their own step and make their own choices [15]. Furthermore, mobile devices can be used as collaborative tools and collaboration plays an important role in the process of learning, according to contemporary learning theories, such as social constructivism [15][18]. Last but not least, the use of mobile devices seems to achieve *seamless learning*. Seamless learning implies ‘that students can learn whenever they are curious, using personal devices and embedded learning technology to store, share and recall contextualized knowledge, creating an experience of continuity and the ability to switch rapidly from one learning project to another’ [15].

Extending the concept of seamless learning environments, emerges the concept of *ubiquitous learning*. With the omnipresence of computers and smart digital devices in our lives (ubiquitous computing), the transformation of e-learning, to a new modus of learning seemed inevitable. Ubiquitous learning (u-Learning) is conceived as learning where “all students have access to a variety of digital devices and services, including computers connected to the Internet and mobile computing devices, whenever and wherever they need them” [19]. A comparison regarding the physical devices, the communication and the learning between e-learning, m-learning and u-learning is illustrated in Figure 1.2.

Figure 1.2 Comparisons and flow of e-learning, m-learning and u-learning (Figure taken from: [19]).



To summarize and efficiently describe ubiquitous learning five characteristics have been proposed [20]:

- *Permanency*: The content is permanent, unless the user removes it
- *Accessibility*: The content is always available to the users
- *Immediacy*: The content can be immediately retrieved by the users.
- *Interactivity*: The learners may interact with peers, teachers, and experts efficiently and effectively
- *Context-awareness*: The environment may adapt to the learners real situation to provide adequate content for the learners.

Ubiquitous learning, described above, is a new vision not only for learning, but for our society and future. It is an educational paradigm that will change learning as we know it [21], as it will:

- Overcome traditional institutional, spatial and temporal boundaries of education, and enable 'lifelong and lifewide education'. It implements informal, semi-formal learning and 'over-the-shoulder-learning' from friends and colleagues.
- Change balances. Shifts the role of the student from a passive recipient of transmitted knowledge to a collaborative co-designer of knowledge, through the ability to select, modify and alter the content.
- Identify learner differences. Respects the learner's differences, interests and needs. Collaborative and team work is also enabled.
- Include a vast range of representational modes and change conceptualizing capacities. Ubiquitous learning involves oral, written, visual and audio mode of content and knowledge upon describing it and categorizing it should be developed.
- Connect one's own thinking into the social mind of distributed cognition. Now that knowledge is freely and widely available, knowledge should focus in deep understanding and not on simple memorization.
- Create collaborative knowledge cultures. Ubiquitous can build 'communities of practice' to support learning, regardless of gender, race or origin.

1.1.4 Learning Analytics - a new promising field

As learning becomes a ‘ubiquitous’ procedure and students may deploy learning in various places, formats and contexts, assessment of learning is even more difficult and important. Educators and instructors around the world not only need to assess their students, but also need to evaluate upon their content, courses and effectiveness of their teaching. This assessment leads to the improvement of the teaching process and the maximization of the educational results. To this direction, Learning Analytics (LA) has emerged to improve learning and education. Learning Analytics is a field that is closely related to business intelligence (BI), web analytics, academic analytics, educational data mining, and action analytics [22]. It aims to the statistical evaluation of data and the identification of patterns, which are used to predict future behavior and is used to support decisions aimed at improving outcomes and detect problems in student performance. Extracting Analytics and therefore knowledge from gathered data can be used by students as self-awareness tools; by teachers for self-evaluation and detection of issues in their classroom or as a motive for improvement, while schools may use tools for their planning, decision- making and as part of Business Intelligence [22]–[24]. Several representations of the analytical process have been developed over time. A comparison among five popular frameworks is attempted in [22], and illustrated in the following Table. As one can notice, data collection is essential in all models and critical for successful analysis [23].

Figure 1.3 Comparison of Frameworks for Learning Analytics (Taken from: [22]).

Knowledge Continuum	Five Steps of Analytics	Web Analytics Objectives	Collective Applications Model	Processes of Learning Analytics
Data	Capture	Define goals	Select	Select
Information	Report	Measure	Capture	Capture
Knowledge	Predict		Aggregate	Aggregate & Report
Wisdom	Act	Use	Process	Predict
	Refine		Display	Use
		Share		Refine
				Share

1.2 Scope and Objective

The scope of this thesis is the development and implementation of an Adaptive Learning Ecosystem that is independent of LMS, tracks users' activities from learning content that is distributed among various systems and adapts its content accordingly. Our software interacts with the learners via interactive content and provides powerful monitoring tools to the instructors. Our Teacher's Dashboard gathers information from all learning activities from diverse sources and offers reports and charts as Learning Analytics to the teacher.

In this context, the present thesis aims to introduce the reader to Learning Analytics and present Activity Streams, their uses and their potential in e-learning. It focuses on xAPI (Experience API), a specification designed for activity streams in learning and cites its architecture, vocabulary and integration with Learning Management Systems (LMSs). Moreover, it demonstrates its adopters and applications using it, in order to manifest its wide acceptance and vast range of uses. Eventually, it compares the specification to an older e-learning content specification, SCORM (Sharable Content Object Reference Model) and discusses its impact on learning and its position in a constantly changing world.

To be more specific, our Adaptive Learning Ecosystem makes use of xAPI (Experience API) – a technology used to track learner's activities- to adjust the content of a course according to a user's previous activities and allow for the seamless communication of two different courses without the use of an LMS. Additionally, our implementation provides xAPI- enabled YouTube videos and xAPI-enabled interactive videos that send statements about a user's actions and adjust the content according to the user's interaction respectively. Moreover, it implements sequencing via xAPI streams at the provision of learning resources and allows for communication between independent learning modules. All the statements gathered from our implementation, although deriving from diverse sources and hosts, accumulate in a central repository and provide data for the Teacher's Dashboard. The Teacher's Dashboard is a powerful monitoring tool that expands existing modules and represents data about the users and learning content. In particular, it represents data about the users that used the Adaptive Learning Ecosystem and illustrates charts about the use of the course, the average scores of users and per activity, the progress of the users etc.

1.3 Outline of Thesis

This thesis is organized in chapters, as follows:

In **Chapter 1 - Introduction**, the current chapter, a brief introduction of the topic is made, Adaptivity and sequencing in e-learning are discussed, the transition from e-learning to m-learning and the vision of ubiquitous learning are presented and the emerging trend of Learning Analytics is introduced. Additionally, the scope and the main objectives of this dissertation are highlighted, and the publications that were produced in the context of the present Master's thesis are presented.

In **Chapter 2 - Overview of the xAPI Ecosystem**, the Experience API is being presented in detail. First, the need for its introduction is announced, then some short historical clues are given and the details for the specification are explained.

In **Chapter 3 -** , related work regarding xAPI is presented. Adopters of xAPI in education and training are presented, while libraries, tools and record stores that have been developed for xAPI and that will be used in our implementation are mentioned.

In **Chapter 4** - the architecture of our implementation is presented and details about the infrastructure are given.

In **Chapter 5** - our courses and learning content are explained in detail. The features, technical specifications, screenshots and flowcharts along with excerpts of code are displayed here.

In **Chapter 6** - the Teacher's Dashboard is presented. The features, screenshots and excerpts of code implementing charts, reports and monitoring tools are illustrated here.

In **Chapter 7** - an evaluation of our implementation is presented, along with charts and reports regarding the evaluation group.

In **Chapter 8 - Conclusions** a review on Experience API from many perspectives and a rough comparison to its predecessor SCORM is attempted. Future work and extensions of our implementation are also presented.

1.4 Publications

Part of the research for this thesis has been published in our chapter "Handling Big Data in the Era of Internet of Things (IoT)." of *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, published by Springer International Publishing in 2017. Significant part of the work reported in this thesis has been submitted and accepted (peer-reviewed) as full-text paper under the title 'Implementing an Adaptive Learning System with the Use of Experience API' for the 2nd EAI International Conference on Design, Learning & Innovation¹ to take place in 30-31 October 2017, in Heraklion, Greece.

¹ <http://designlearninginnovation.org/2017>

Chapter 2 - Overview of the xAPI Ecosystem

In this chapter, learning analytics specifications will be shortly described, then SCORM, xAPI's predecessor, will be shortly introduced, later the evolution from SCORM to xAPI will be addressed and finally the core components, vocabulary and architecture of xAPI along with its integration with LMS will be presented and an example of an xAPI statement will be demonstrated.

2.1 Related specifications

As earlier stated, capturing Data from various sources is essential in Learning Analytics. In order to obtain successful analysis for learning data, the data should follow some specification and for this purpose three specifications have been proposed [23], [25]: the Caliper (IMS Caliper Analytics) [26], the IEEE Standard for Learning Technology [27] and the Experience API (xAPI).

IMS Global Learning Consortium Inc., a non-profit, member organization that aims to enable the adoption and impact of innovative learning technology in education, advertises Caliper as “the world's first interoperability standard for educational click stream data” [25], [28].

The main elements of the specification include [26], [28][29]:

- A *Learning Activity*, which is an activity that can be a component of a learning sequence
- *Caliper SensorAPI™* and *Caliper Sensor* which are the tools that implement the capturing of activities
- *Metric Profiles* are profiles that describe the types of learning activities which can be tracked

Additionally, Caliper leverages and extends the Learning Tools Interoperability® (LTI®), Learning Information Services, and Question and Test Interoperability® (QTI®) standards [26]. An extensive comparison between the Caliper and the xAPI specification is discussed in [30], whereas a contrast deployed in [28] stresses out their differences but also their potentials as well as a suggestion for convergence. However, from both comparisons, the vast versatility and openness of the xAPI specification is obvious.

The IEEE 1484.11 Standard for Learning Technology includes a Content Management Instruction (CMI) specification, i.e. a set of guidelines for interoperability between web courses and the LMS. The CMI provides both a data structure for student interactions with learning contents as well as an API for managing these data [23], [25]. It consists of two parts: the IEEE 1484.11.1-Data Model for Content Object Communication which describes the data model and the IEEE 1484.11.2-Application Programming Interface for Content to describe the communication. The latter also defines an API for sending and requesting data between the educational content and the LMS, which is only accessible by the educational content. This is the most serious drawback of IEEE 14811.11 specification [23].

In the next sections the xAPI specification is introduced and shortly described. The versatility and openness of the specification along with its vast adoption [28] and simplicity [31] were our basic criteria to choose and elaborate with it.

2.2 ADL's SCORM: the predecessor of Experience API

Although SCORM was welcomed with applauses, adopted, supported and compliant with popular LMSs and perhaps the most “widely used e-learning format” [32], rapid rise of technology caused its glory to gradually fade away. To start with, SCORM is tightly connected to the LMS (“LMS-centric” as stated in [4]) and cannot exist autonomously [33]. However, in a constantly changing world, where learning happens also outside the LMS and through mobile devices (tablets, smartphones, smart television sets even gaming consoles), there is a need for support of informal and ubiquitous education [33], which is neatly described with the motto “Learning is happening everywhere” [34].

That was the vision Learning-Education-Training Systems Interoperability (LETSI) tried to realize in 2008, when it started investigating the requirements of the next generation of SCORM (SCORM 2.0) [35]. After lots of whitepapers and suggestions, ADL focused on standardized experience tracking capabilities and in 2010 a Broad Agency Announcement (BAA) project evolved: the “Experience API.” Rustici Software - the company that undertook the project - renamed it to “Project Tin Can” as this term implied the two-way conversation between the company and the e-learning industry [34], [35], but today this term seems to retreat.

2.3 Inside the Experience API specification

The new technical specification that was born is called Experience API (also known as xAPI or Tin Can API) and was formally launched in 2012, under the version 0.9. Up to today several versions have been introduced to add extra functionality and clarify identified issues. The current version (at the time of writing, August 2017) 1.0.3, was launched in September, 2016 [2]. The xAPI was and remains an open source, learning technologies interoperability specification that defines with details how tracking of learner activities and experiences between technologies should be delivered [36]. It is licensed under the Apache License, Version 2.0 and has the warm support by the community which maintains and updates it [35].

2.3.1 Activity Streams

Activity Tracking is a challenging new area of research, promising to extend our knowledge about people's behaviors, through Activity Streams. Popular social media, such as the Facebook and the Twitter use Activity Streams to provide information about activities a user made, e.g. Ann posted a photo, Mike commented on a post, Dennis likes imdb.com. These activity streams are mainly in the format of *someone did something*. The concept of xAPI is mainly generating such an activity stream, when a learner does something, e.g. when he accomplishes a course, or passes a test. Later, the educator can access these activity streams, assess the learners and come to valuable conclusions about the course. What is novel in xAPI is that activity streams may originate from various means and contexts and not necessarily from a course [5], [33].

Activity streams in xAPI are named "**Statements**" (xAPI statements) and describe how the learner interacted with an object. In their core format they follow the triple pattern of *<Actor, Verb, Object>* obeying to the *someone did something* model, which is very common in natural languages. However, this format may be extended to support a variety of objects. This is done by adding extra optional information, such as the result of a quiz, the timestamp of the activity or the context of an activity [36], [37]. The Tin Can API supports a predefined Vocabulary, comprising of a large set of Verbs, and Activity Types to support various occasions. The statements are identified by a unique UUID (Universally Unique Identifier) and are transferred and stored in JSON (JavaScript Object Notation) format. JSON achieves minimal size transfers, combines library support and relatively easy human

readability and has been used in the development of a plethora of APIs; it is therefore the ideal solution for building xAPI statements [38]. The Vocabulary is presented in detail in the xAPI Registry [39] and will be shortly described later in this section.

2.3.2 Activity Providers in xAPI

Sources that may provide xAPI statements are generally described as **Activity Providers** [34]. More technically an Activity Provider is “any tool or system that generates data about learning experiences, achievements and job performance and sends it to the LRS”. A wide range of learning technologies from a social learning platform to an interactive whiteboard or from an e-learning course to an event attendance application may play the role of an activity Provider in the xAPI Infrastructure [5]. This support of multiple and versatile Activity Providers makes xAPI an ideal solution for the realization of the vision “Learning Everywhere”.

2.3.3 Learning Record Store (LRS)

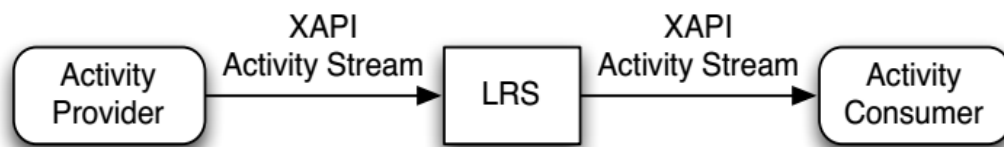
The xAPI specification does not only define the structure of the streams of learning experiences, but also specifies the details for their transfer and storage. All records generated from the xAPI are transferred and kept in a server, called Learning Record Store (LRS), which is the heart of the infrastructure [5], [36]. An LRS is responsible for receiving, validating and storing the statements. It also provides access to the statements, as it allows the retrieval of these statements by external applications. An LRS can host activities from diverse Activity Providers and that is its main advantage: whether the statement comes from a serious game, a mobile application or a webpage it can be stored under the same format in the LRS. At the same time, a LRS can seamlessly exchange data with other LRSs that may serve different purposes. This expands the potential of xAPI significantly. Due to the fact that an LRS collects data from diverse learning environments it is ideal for data aggregation and analytics [5], [33]. Apart from reporting and analytics, an LRS can provide valuable information for personalized approaches, as activities stored in the format *<who did what>* can be easily processed.

2.3.4 Activity Consumers in xAPI

The statements that derive from an LRS are used by further applications, the *Activity Consumers*. A Business Analytics application or an open badges platform may play the role

of the activity consumer. Additionally a dashboard may ‘consume’ the activity streams hosted in the LRS offering personalized information to the user. This flow of data is illustrated in Figure 2.1.

Figure 2.1 An Activity Provider interacts with an LRS which in turn interacts with an Activity Consumer (Figure taken from [37]).

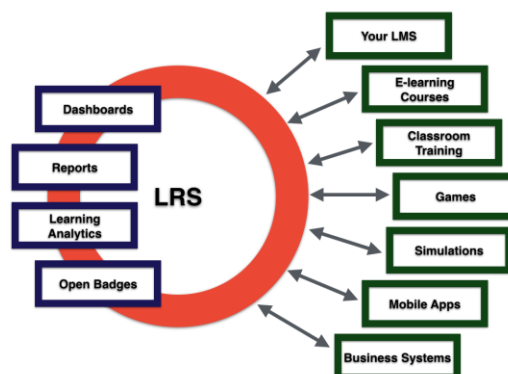


2.3.5 Interaction between LMS and LRS

As aforementioned an LRS need not coexist or communicate with an LMS. In reality, there are three architectures that are popular and describe how the LRS is used and how data stored in an LRS can integrate with other applications [40].

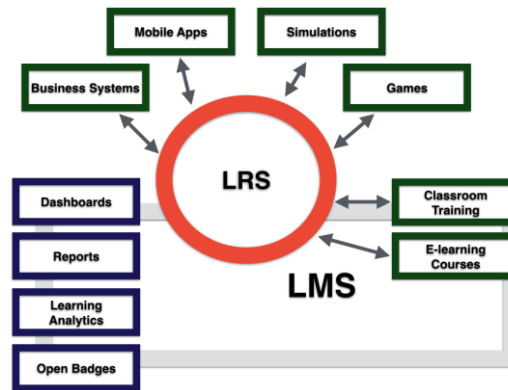
1st architecture: The LRS is independent of the LMS and may interact with it for some xAPI courses. The data aggregated can be used for reports, analytics or badges. This is shown in Figure 2.2. The LRS gathers data from LMSs, independent e-learning courses, games, simulations, mobile applications and Business Systems. It may provide data to Dashboards create reports and form Learning Analytics or serve as a provider for Open Badges

Figure 2.2 A standalone LRS interacting with various activity providers. (Figure taken from: [40]).



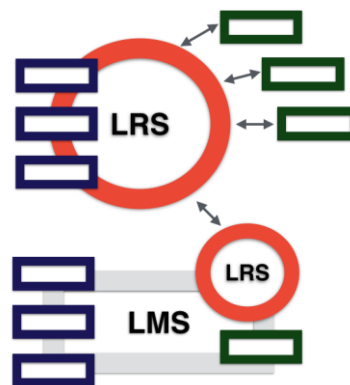
2nd architecture: The LRS is incorporated in the LMS but may interact with external Activity Providers or LRS. In this case, the LRS draws data from the LMS but also from alternative providers. Again it may provide data to Dashboards, Reports and Learning Analytics Applications or serve as a provider for Open Badges This is illustrated in Figure 2.3.

Figure 2.3 The LRS is incorporated in the LMS (Figure taken from: [40]).



3rd architecture: Multiple LRSs can be used. A LRS may be incorporated in the LMS but may interact with external LRSs. This is a hybrid architecture combining the architectures that are above described and is illustrated in Figure 2.4

Figure 2.4 Multiple LRS (Figure taken from: [40]).



2.3.6 Vocabulary

The core element of Tin Can API statements in JSON is the object and is denoted with a pair of braces {...}. Each object may have properties (keys) and each property is paired with a value. The property name is in double or single quote, while the value may be a text (quoted string), a number, Boolean or a nested object. Nesting objects is possible in Tin Can Statement [38].

2.3.6.1 Actors

The first component of the triplet, the **Actor**, indicates who is the person that ‘acts’. The Actor may be an Agent or a Group. An Agent, that is a user or a learner, is easily identified by his email and there the pairing between property *mbox* and an email address can be used. For example `mbox: "mtp130@edu.teicrete.gr" objectType: "Agent"`, is a part of the statement that defines the learner. An additional piece of information name can be used to make it even more readable. The following object represents the Agent Koralia with email `mtp130@edu.teicrete.gr`.

```
{mbox: "mtp130@edu.teicrete.gr" objectType: "Agent", name: "Koralia"}
```

In the case of Group, several members may constitute the group and may be included as enumeration in the object or the group may have its own identifier to be distinguished.

2.3.6.2 Verbs

The second and vital part of the statement, the **verb**, describes the action being taken, i.e. the experience being recorded. The action is described with the property *id* which points to a URI (or IRI), as in the example

```
{
  id:http://adlnet.gov/expapi/verbs/completed
}
```

Since xAPI aims to readability as well as activity transfer an additional field, *display*, may be used to indicate the action in various languages. The following action is displayed in American English.

```
{
  id: "http://adlnet.gov/expapi/verbs/experienced",
  display: {
    "en-US": "experienced"
  }
}
```

As the xAPI specification evolves and covers various needs, it supports not only traditional learning activities (attempted, passed, failed, completed) but social media actions (shared, like, tweeted, followed) as well. The list of statements is extended occasionally and may involve actions defined by ADL or other providers. Some examples of xAPI actions are provided in Table 2.1.

Table 2.1 A selection of Actions of Tin Can API.

URI	display	Description
http://activitystrea.ms/schema/1.0/attend	attended	Indicates that the actor has attended the object. For instance, a person attending a meeting.
http://adlnet.gov/expapi/verbs/experienced	experienced	A generic verb to say that someone experienced (e.g. viewed, listened to, read, etc.) some form of content.
http://future-learning.info/xAPI/verb/pressed	pressed	Indicates that the actor has pressed the object. For instance, a person pressing a key of a keyboard.
http://id.tincanapi.com/verb/tweeted	tweeted	Indicates that an agent tweeted on Twitter.
https://brindlewaye.com/xAPITerms/verbs/walked	walked	Indicates that the actor walked a distance indicated by the activity

A full list of verbs is available in the xAPI Registry [39] and may indeed describe a wide area of actions. However, new verbs may be created by content developers, when the existing ones cannot suffice.

2.3.6.3 Activities

The third part of the Tin Can API statement, the **Activity**, is perhaps the most abstract and describes the “thing” or the object. The activity is actually what is to be tracked and depends upon the requirements of the implementation. An Activity object may include

only three properties. The only requirement is an *'id'* property that has a value that is an Internationalized Resource Identifier (IRI). The other two optional properties are the *'objectType'* which has a value of "Activity" by default, and the *'definition'* which itself is an object. The IRI may be a non- resolvable URL, but in order to avoid collisions and to retain future compatibility, resolvability is highly suggested. An example of an activity from our implementation is given below and refers to a unit from a course.

id: "http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A"

Activity types are specified in the Registry[39], and may be of various types again serving diverse needs; some examples are given in Table 2.2.

Table 2.2 A selection of Activities of Tin Can API.

URI	display	Description
http://activitystrea.ms/schema/1.0/application	application	Represents any kind of software application.
http://adlnet.gov/expapi/activities/question	question	A question is typically part of an assessment and requires a response from the learner, a response that is then evaluated for correctness.
http://id.tincanapi.com/activitytype/tweet	tweet	A short message sent on Twitter.

2.3.6.4 Others

In the case additional information is needed, extra properties defined by the specification, can be used. The most common are *'id'*, *'timestamp'*, *'authority'*, and *'version'*.

The *id* parameter is the unique identifier of the statement and should be universally unique in order for communication between different LRSs to be accomplished. Assigning a unique ID to a statement is responsibility of the LRS.

The *timestamp* parameter is date and time value in a string format that indicates when the statement was created. It can be used for offline mode of the specification, for implementation of the queues and for identifying the statements based on time criteria.

The *authority* parameter represents information about the authorization of the statement and the level of trust.

Finally the *version* parameter suggests the version of xAPI the statement was created.

2.3.6.5 Result

As xAPI was mainly designed for tracking learning activities, result of an assessment or - more widely - learning activity should be tracked and captured. This is accomplished with the Result object which supports six optional properties:

success: takes a Boolean value (true/false) and indicates the successful (or not) result of an activity,

score: takes an object as value. This object describes the score with properties such as *raw* (points of activity), *scaled* (decimal number representing the percentage), *max* and *min*,

completion: takes a Boolean value (true/false) and indicates the completion of an activity,

duration: indicates the length of time elapsed for the experience captured by the statement,

response: is a string value used to capture additional information,

extensions: an object to catch all data necessary to capture the meaning of the result.

2.3.7 An example of an xAPI statement from our implementation.

The following statement comes from our implementation and is stored in JSON. It displays most of the objects and properties mentioned above. The UUID of the statement is e5a7d141-b4d2-4acd-8c37-8880f4b0cc02 which is assigned by our LRS. The statement can be summarized as *korialiap@test.com completed the Activity http://koralia/Test_in_text_Formatting successfully with score 100%*.

```
{
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/completed",
    "display": {
      "und": "completed"
    }
  },
  "version": "1.0.2",
```

```

"timestamp": "2017-05-18T17:30:32.680Z",
"object": {
  "id": "http://koralia/Test_in_text_Formatting",
  "objectType": "Activity"
},
"actor": {
  "mbox": "mailto:koralia@test.com",
  "objectType": "Agent"
},
"stored": "2017-05-18T17:30:31.282349+00:00",
"result": {
  "completion": true,
  "score": {
    "scaled": 1
  },
  "success": true
},
"id": "e5a7d141-b4d2-4acd-8c37-8880f4b0cc02",
"authority": {
  "mbox": "mailto:mtp130@edu.teicrete.gr",
  "name": "koralia",
  "objectType": "Agent"
}

```

Chapter 3 - Related work in xAPI

3.1 Applications of xAPI

xAPI gains ground every day in applications in Education as it supports tracking of learning activities. However its potential is recognized by various industries and xAPI is adopted not only by Learning Industry, but also in vocational training, games and defense. In this section enterprise software that use xAPI will be shortly presented, whereas open source developer's tools for the implementation of xAPI solutions will also be reported.

3.1.1 Applications of xAPI in Education

xAPI is becoming more and more popular in Learning Industry and adopters of xAPI in education include LMSs, Authoring Tools, Assessment Systems, Simulations, Games and Social Learning [41]. Some of the adopters of xAPI are indicatively displayed in Figure 3.1 , for a full list please visit [41].

Figure 3.1 Some of the adopters of Tin Can API (Figure taken from [41]).



As the LRS is the key component of xAPI, various implementations of LRSs have been developed. SCORM CLOUD (<https://cloud.scorm.com>) offers full functionality of a LRS and much more; it supports both SCORM content and xAPI statements, offers automatic conversion from SCORM content to xAPI, may store the courses in the cloud and provides basic reporting. According to the user's needs it may offer free or billed accounts. ADL's LRS (<https://lrs.adlnet.gov/>), an open source standalone application that hosts xAPI statements, has the constant support of the community and communicates with other ADL's xAPI

modules. lxHive (<http://www.lxhive.com/>) is another open source solution, while Learning Locker (<https://learninglocker.net/>) is offered as an open source product for personal use or as commercial for enterprise use. Several commercial products, like GrassBlade LRS (<https://www.nextsoftwaresolutions.com/grassblade-lrs-experience-api/>) or waxLRS (<http://www.saltbox.com/>) which have been also developed to provide services to major companies and organizations have also included xAPI support.

While the LRS provides the storage for the statements, the course may be hosted in an LMS. LRSs successfully cooperate with LMSs in order to track activities from courses hosted in the LMS. A large number of popular LMSs support the xAPI specification; either supporting xAPI content or also come with an integrated LRS. In Table 3.1, the degree of integration of xAPI for popular LMSs, such as Moodle, Docebo, Talent LMS, SAKAI is presented [41].

Table 3.1 Support of Tin Can API from popular LMSs (LMSs in alphabetical order).

LMS	Supports xAPI content	Built-in LRS
1. Blackboard	Yes	No
2. DoceBo	Yes	Yes
3. Litmos	Yes	Yes
4. Moodle	Yes (via plugin)	No
5. Sakai	Under construction	Under construction
6. TalentLMS	Yes	Yes
7. TOPYX	Yes	Yes

As popular LMSs support xAPI content, it is expected for authoring tools to support creation of xAPI content, as well. In Table 3.2 popular authoring tools that support xAPI learning content are presented, along with details upon their support of xAPI [41].

Table 3.2 Support of Tin Can API from popular authoring tools.

	Authoring Tool	Manufacturer	Features
1.	Captivate® (7+)	Adobe	Courses can be exported in xAPI packages
2.	Storyline & Presenter	Articulate	Creates xAPI-enabled e-learning courses
3.	Exam Builder	Exam Builder	Issues xAPI statements to send to one or multiple LRSs
4.	dominKnow Platform	Claro	Standard statements sent and may send one or more statements about any type of action that is occurring in a course
5.	H5P	H5P	Records the user's actions and experiences available as xAPI statements which the publishing platform can retrieve and send on to an LRS.

Lastly, some more adopters of Tin Can API are presented in Table 3.3 to illustrate its possibilities and realization of ubiquitous learning [41].

Table 3.3 Case Studies in Learning.

	Software	Implementation
1.	AEL Data	AEL Data offers e-book reading and learning solutions for mobile and web platforms. Solutions are in compliance with international standards such as ePub2, ePub3, SCORM and xAPI as per client requirements.
2.	Curatr	Curatr is an online platform that enables instructors to transform any digital content into a social game for learning. The xAPI forms a core part of their platform, enabling enhanced tracking and personalisation features to report back advanced usage data seamlessly to any xAPI enabled LRS or LMS.

	Software	Implementation
3.	eLearning Studios	eLearning Studios create completely Virtual Reality Learning Simulations (VRLS) and engaging serious games that leverage the powerful xAPI to track not only traditional learning objectives, but also situational and advanced learner data
4.	Flipick	Flipick provides self-paced learning services. These modules can use 2D/3D animations and case studies, HTML/CSS3 simulations as well as mock assessments and tracking each step of the learning path is xAPI enabled.
5.	HiHaHo	HiHaHo connects online interactive videos with a Learning Management System (LMS) or Learning Record Store (LRS) with xAPI. Interactivity can be added to online videos and the behavior and results of viewers can be tracked through xAPI.
6.	PulseWeb	PulseWeb covers the whole spectrum of traditional learning, e-learning, workplace learning, and learning between companies. It uses the xAPI for registration of learning activities, populating learning dashboards and to power gamification inside the learning processes.
7.	Rapport	Rapport leverages the xAPI to provide robust tracking of not only learning progress and scores but also detailed recording of student interactions in a wide variety of learning activities and 3D simulations
8.	SquirrelThat (Kalleo People Group)	SquirrelThat uses the xAPI feed to tell the LRS what learners are learning, what action choices they are making and whether these choices were implemented (while reading a book, doing an online course, attending a conference, or even in a coaching session)
9.	Tappestry	Tappestry is an iOS app that encourages people to reflect on their daily experiences to sort out what they did and how they learned from it. The app collects what was learned, photos, people involved, ratings, and categories through xAPI.

3.1.2 Applications of xAPI in Training

Apart from its penetration into Education, xAPI becomes popular in Training as a rising number of companies use it in order to improve training, to record activities and to enhance performance of the workforce. Mobile applications and plugins are used during working procedures in assessing, developing and increasing the accessibility of content. XAPI helps trainees do skills assessments often with the help of capturing rich image and video evidence. At the same time, trainers can watch the progress of the trainees and combine these data with data collected from other sources. XAPI can prove valuable in decision making and Business Intelligence. Case studies of Applications of xAPI in training [41] are presented in Table 3.4.

Table 3.4 Case Studies in Training.

	Software	Implementation
1.	JobAider	JobAider is an app for creating, using and tracking job aids. It tracks who is creating a job aid, who is using a job aid, which part of a job aid is being viewed etc. The data collected is stored in an LRS. Clients can view the data collected using an analytics dashboard.
2.	Knowledge Guru™	Knowledge Guru™ is a game engine that uses xAPI as a bridge between its stand-alone database and a client's LMS.
3.	SurveyGizmo	SurveyGizmo may send xAPI statements to an LRS or xAPI-enabled-LMS. Statements may include the start/finish of a quiz, the score of a quiz or the pass/fail indication ²
4.	TREK	TREK guides employees along a learning path, providing on-demand content and using the sensors of the mobile device (GPS, camera, audio) to capture real world experiences with the use of xAPI.

² <https://help.surveygizmo.com/help/lmslrs-integration>

	Software	Implementation
5.	WebRatio	WebRatio uses the xAPI to create its own Cloud LMS and an innovative game system which tracks xAPI conformant web-based training, technical articles, job aids, online certifications, tips and tricks and other activities that reward knowledge.

3.1.3 xAPI in Defense

It is not only Education and Business Sector that are interested in xAPI, but Defense as well. The USA's Department of Defense (DoD), in its attempt to develop adaptive, learner-centric systems that drive learning and performance forward [42], engages xAPI in order to track activities from various sources, such as simulators or virtual worlds[43]. Additionally, they need to integrate already used solutions, such as GIFT or SCALE in order to achieve maximum results. To this direction, the Human Performance Measurement Language (HPML) has been proposed, to interact with GIFT and SCALE making use of xAPI statements [42].

3.2 Developers' tools for xAPI

A lot of research and development has been done to implement the xAPI specification and make the communication between the LRS and the content easier. In this section we will shortly present five open source tools that have been developed towards this direction and which we have used in several parts of our implementation. Four of the tools are implemented by the ADL, whereas the other one is developed by Rustici Software.

3.2.1 ADL's LRS

ADL's Learning Record Store (LRS), also mentioned in 3.1.1, is an Open Source software used to store learning data collected with the Experience API. It uses Postgres and Python [44] computer language, is based on the Django web framework and is widely supported by the community.

3.2.2 The JavaScript Library TinCanJS

As inventors of xAPI along with the community have developed libraries in several languages, e.g. JavaScript, C, Java, PHP and Python in order to implement the xAPI

specification, we could choose from those provided, in order to develop our software. The library in JavaScript, called `tincan.js` [45] , is constantly supported and updated by developers of Rustici Software .

3.2.3 The xAPI wrapper

xAPI wrapper [46] is a software designed to simplify communication to an LRS and is provided as open source software from ADL. It is written in JavaScript and among others, it provides with two functions for sending and retrieval of statements, respectively:

- The *sendstatements* function send statements to the LRS
- The *getstatements* function gets statements from the LRS.

Both statements are implemented via PUT/POST and GET HTTP methods. The xAPI wrapper is a module that has been used in other tools, such as the xAPI Dashboard and the xAPI statement viewer.

3.2.4 The xAPI Dashboard

The xAPI Dashboard [47] provides a quick and easy way to generate graphs from xAPI data, as well as a powerful query language to manipulate it. It is implemented in JavaScript and makes use of the library `nv.d3.css` in order to produce a variety of charts. It simplifies the process of extracting meaningful aggregate data from xAPI statements and can generate numerous types of charts and visualizations based on that aggregated xAPI data. It allows developers to run SQL-like queries, filters, and aggregations over xAPI data and can generate numerous types of charts and visualizations based on that aggregated xAPI data.

3.2.5 xAPI-youtube

The xAPI-youtube software [48] allows Youtube Video interactions to be sent to an LRS with the use of xAPI. It uses YouTube's `iframe` API to build statements with `xapi-youtube` statements.js when events are fired and is built upon the xAPI wrapper module (see 3.2.3).

3.3 Research projects on xAPI

Through an extensive research in bibliography, we have not found applications that make use of xAPI in order to develop interactive applications without the use of an LMS. Some applications make use of xAPI statements gathered in an LRS for monitoring and

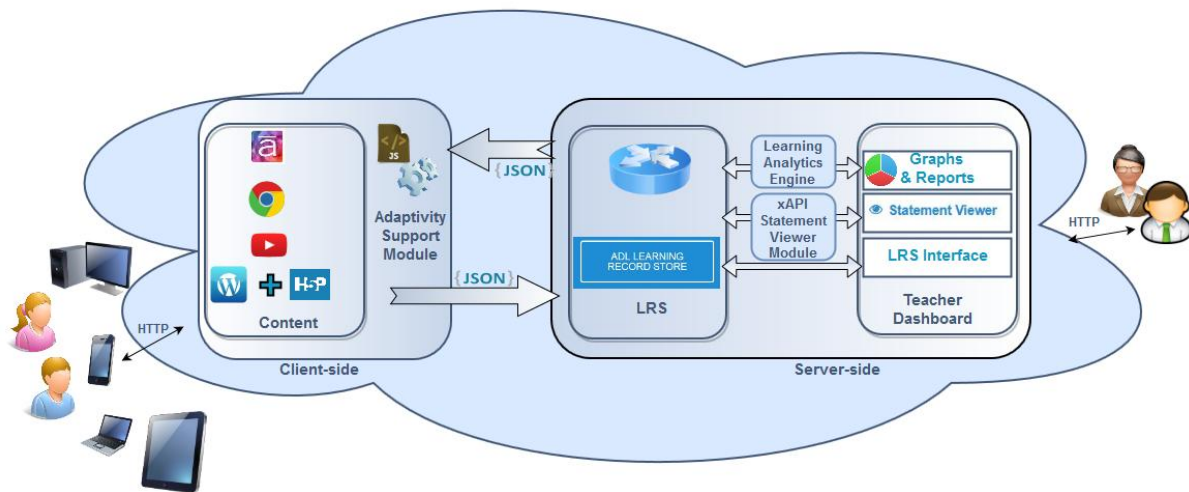
extracting data. The Oregon Trail Game, for example, is a classic e-learning game, that has been updated to track learner activities through xAPI [4], however does not support any interactivity based upon the statements. Other applications require LMS integration. The LIME project includes a recommender system based upon previous activities of the user [25], but it is not made clear, whether the recommendation is interactive based upon the specific user's activities. Moreover, it seems completely tight to the LMS [4] and rules in LIME cannot operate upon individual LRS records, but only upon averages and aggregated data, 'which offer a more equalized view of the learner situation' [25][49]. The Mobler Cards App [37] introduces an application with flashing cards that incorporates an LRS and bases its decisions upon previous activities of the user. Although it requests the LRS in order to orchestrate the sequence of the question, the LRS only stores activities based on the application itself. In our research we have not come across an LMS- independent application that enables adaptive sequencing path according to a user's previous activities and the interaction of two courses through xAPI. Towards this direction, we have implemented an xAPI enabled application that needs not the setup and configuration of an LMS and where users only use their email in order to enter a personalized course. Additionally, the augmentation of our course with extra tools, such as the video-activity tracker and the H5P xAPI enabled content adds extra value to our implementation.

Chapter 4 - Technical Implementation

In this section we will describe our Ecosystem for adaptive learning and its infrastructure. The architecture is illustrated in Figure 4.1 and consists of three main modules:

- The learning *content* which enables tracking activities with xAPI
- The *LRS* which stores the xAPI statements
- The *Dashboard* which provides aggregate and visualized information to the teacher

Figure 4.1 The Architecture of our System



The user may use a computer or a mobile device in order to interact with the content. The content tracks specific activities of the user and sends them to the LRS. An adaptivity support module is added in the client-side module in order to adapt learning content according to the activities recorded in the LRS, regarding the user. The instructor on the other hand, watches the activities of the user with visualizing and monitoring tools provided from our Dashboard. For the reports to be produced and the statement viewer to provide access to the statements a Learning Analytics Engine has been developed and the xAPI statement viewer module has been adapted to our needs.

4.1 LRS

For our implementations, an instance of ADL's Open Source Learning Record Store (LRS) [44] was installed in an UBUNTU server. For this reason, a virtual machine running

UBUNTU 16.04.1 LTS server was created in Grnet (OKEANOS ~Cyclades) (IP address: <http://83.212.100.157/>). Our instance was installed and configured to host our xAPI statements and listen to the port 8000. During the configuration the authentication credentials were given. The LRS provides with a user friendly interface to view basic configuration, a screenshot of which is shown in Figure 4.2. The endpoint is actually a URL within the LRS that listens for HTTP requests. In our case the endpoint is:

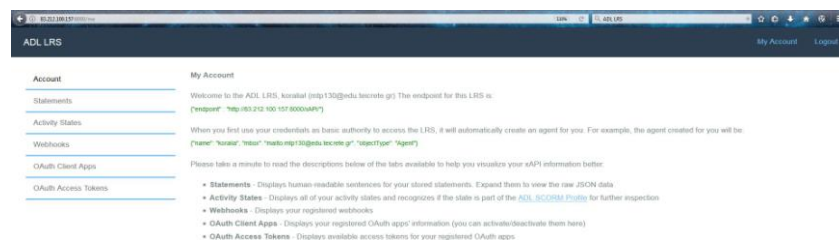
```
{"endpoint" : "http://83.212.100.157:8000/xAPI/"}
```

In order to send or receive statements from the LRS, the software sending/receiving the statements must have authorization. In our case the authorized user has the name koralia and email mtp130@edu.teicrete.gr as denoted by the JSON statement:

```
{"name": "koralia", "mbox": "mailto:mtp130@edu.teicrete.gr", "objectType": "Agent"}
```

During the configuration of the LRS, we also set a password for the user mtp130@edu.teicrete.gr to accomplish authentication.

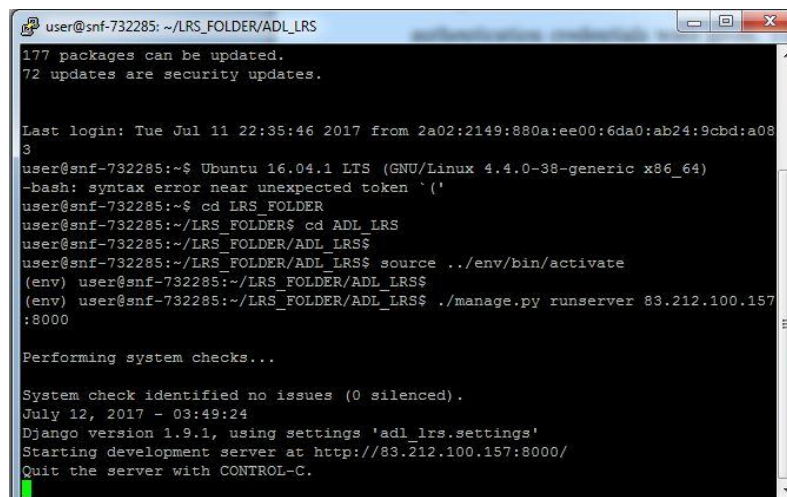
Figure 4.2 The details of our account from our LRS



The LRS runs in Linux and is activated with the following commands:

```
source ../env/bin/activate
./manage.py runserver 83.212.100.157:8000
```

Figure 4.3 Our LRS started successfully and listens to 83.212.100.157:8000

A terminal window titled 'user@snf-732285: ~/LRS_FOLDER/ADL_LRS'. The terminal output shows system updates, login information, and the successful execution of the LRS application. The application performs system checks, identifies no issues, and starts a Django development server at http://83.212.100.157:8000/. The terminal text is as follows:

```
user@snf-732285: ~/LRS_FOLDER/ADL_LRS
177 packages can be updated.
72 updates are security updates.

Last login: Tue Jul 11 22:35:46 2017 from 2a02:2149:880a:ee00:6da0:ab24:9cbd:a08
3
user@snf-732285:~$ Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)
-bash: syntax error near unexpected token `('
user@snf-732285:~$ cd LRS_FOLDER
user@snf-732285:~/LRS_FOLDER$ cd ADL_LRS
user@snf-732285:~/LRS_FOLDER/ADL_LRS$
user@snf-732285:~/LRS_FOLDER/ADL_LRS$ source ../env/bin/activate
(env) user@snf-732285:~/LRS_FOLDER/ADL_LRS$
(env) user@snf-732285:~/LRS_FOLDER/ADL_LRS$ ./manage.py runserver 83.212.100.157:8000

Performing system checks...

System check identified no issues (0 silenced).
July 12, 2017 - 03:49:24
Django version 1.9.1, using settings 'adl_lrs.settings'
Starting development server at http://83.212.100.157:8000/
Quit the server with CONTROL-C.
```

4.2 Learning Content

The learning content of our implementation consists of four independent parts that can be hosted in different hosts and can be accessed via web-browsers.

- The LMS- independent Adaptive System makes use of the JavaScript Library tincan.js (see 3.2.2) and is developed in Articulate Storyline 2. When specific actions are fired (e.g. the learner starts a part of a course, ends a quiz, or presses a button), a JavaScript script is triggered that sends a statement to the LRS. The statement is dynamically created to provide for the email of the user and for the action that has been fired. Our implementation comes with a module that checks whether related statements have been sent to the LRS regarding the specific user and adapts the content accordingly.
- A sequencing template developed with bootstrap [50] and making use of the JavaScript Library tincan.js (see 3.2.2). The sequencing template sends statements to the LRS and navigates the user through the content based upon his previous activities.
- YouTube xAPI-enabled content that makes use of xAPI-youtube module (3.2.5) and bootstrap to provide for a list of videos that enable the monitoring of the user's actions (e.g. played, stopped, seeked a video).
- Interactive video created in H5P [51], where monitoring user's activities is enabled via plugins.

All the modules of the learning content that use the tincan.js library communicate with the LRS via PUT/POST and GET HTTP methods, whereas the YouTube xAPI-enabled video makes use of the HTTP PUT/POST and GET functions implemented by xAPI Wrapper. The plugin for the H5P xAPI-enabled video uses HTTP POST and GET functions.

The learning content is available as web pages and hosted in <http://users.sch.gr/koraliapap/www/reporting/>. A main page that navigates the learner through the learning content is available at <http://users.sch.gr/koraliapap/www/reporting/index.html>.

4.3 Teacher's Dashboard

The Teacher's Dashboard integrates, expands and configures existing modules into a complete monitoring tool for the instructor of the course. In order to provide a thorough mechanism for the monitoring of users' activities and performance we have assembled four modules that provide an exhaustive review of the users' activities, scores and progress.

First, the Teacher's Dashboard provides access to the LRS via the interface of the LRS to illustrate the statements that have been registered to the LRS in short and full format. Secondly, the Teacher's Dashboard makes use of the ADL's xAPI-Statement-Viewer [52], a module that has been developed to pull statements from the LRS using xAPI Wrapper (see 3.2.3) and display them in a sortable and filterable table. Our Dashboard expands the xAPI-Statement-Viewer module by adding an extra filtering field to offer supplementary report upon specific parts of our content. Third, the Teacher's Dashboard employs the xAPI Dashboard (see 3.2.4) in our Learning Analytics Engine to provide for aggregate and analytical reports and charts about our implementation. With xAPI Dashboard and with the use of d3 and nvd3 libraries we have created depictive charts and reports that demonstrate the use of the LRS, the activities, and the scores and progress of the users. Finally, our Dashboard provides access to the WordPress Monitoring Tool that illustrates scores and statistics for the interactive H5P content developed.

The communication between the modules of our Dashboard and the LRS is made with HTTP GET method to collect and aggregate statements from the LRS. In specific, the xAPI-Statement-Viewer and our Learning Analytics Engine (using xAPI Dashboard) are built upon the xAPI wrapper which implements the communication via HTTP PUT/POST/GET methods.

The Word Press plugin for the H5P xAPI-enabled content implements the communication via HTTP POST/GET functions.

All features of the Dashboard are hosted in <http://users.sch.gr/koraliapap/www/reporting/> and are available to the Tutors via <http://users.sch.gr/koraliapap/www/reporting/index.html>.

Chapter 5 - Learning Content

In this section we will describe the parts of our content and the technical specifications behind it.

Our content consists of four things:

- *An Adaptive Learning System*: Two courses interacting with an LRS and through it with each other
- *YouTube xAPI enabled videos* that send activity to the LRS
- *H5P interactive and xAPI enabled content* hosted in WordPress sending statements to our LRS
- One *sequencing* template interacting with an LRS where decision making is based upon the statements provided to the LRS

All statements, although deriving from diverse sources, are directed to our LRS, as described in 4.1.

The content is html files, hosted in <http://users.sch.gr/koraliapap/www/> and is available via an html file (index.html) displayed in Figure 5.1 .

Figure 5.1 The learning content of our implementation



Tracking Learning Activities with xAPI (aka TinCan API)

A Master Thesis Implementation

Koralia Papadokostaki, BSc in Informatics

- All activities are recorded in our LRS ([credentials needed](#)).
- Course on Text editing with xAPI with Articulate 2 by Koralia Papadokostaki.
- Adaptive Course on Spreadsheets communicating with Text Editing Course through xapi with Articulate 2 by Koralia Papadokostaki
- Interactive content, with H5P and xAPI configured and modified by Koralia Papadokostaki.
- xAPI Youtube Video Listener, created by ADL, enriched by Koralia Papadokostaki .
- Sequencing Template built up from scratch by Koralia Papadokostaki with TinCan.js and Javascript.

All activities are recorded in our LRS ([credentials needed](#)) which is an Instance of ADL's LRS installed in our VM in Cyclades.

All implementations are hosted in our site in SCH.gr

5.1 An LMS- independent Adaptive Learning System

5.1.1 Overview

Our vision was to take advantage of the capabilities of xAPI in order to create an adaptive learning system [5], [8] that will adjust its content according to the previous activities a learner has accomplished. For the learning system to be effective and accurate we had to use the online delivery of the statements from the activity provider towards the LRS. Additionally, our learning system should have access to the statements in the LRS, in order to change the content accordingly. In our case, the Activity Provider and the Activity Consumer are actually parts of the same application. The intermediate service, the LRS, stores the statements and acts as a server to our client- server application. However, the decision-making is done in the client as the course runs on the browser.

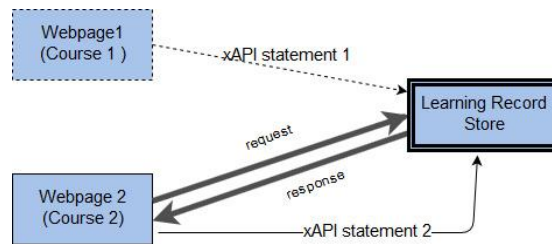
To our knowledge, there has been no other published work regarding An Adaptive Learning System based on xAPI. Activity consuming is mostly used for aggregating data and conclusion extracting. Our application is innovative in the section that the decision making is based upon activities the application itself has produced.

Our implementation is addressed towards fifth and sixth grade students of Primary Education. It is a brief course on spreadsheets that includes a short introduction on their use and usability and demonstrates basic concepts about sheets, cells and their format. As spreadsheets belong to the same office suite with word processors and presentations software, students may be familiar with some features of this software. Therefore, our application consists of two independent courses which are two separate webpages. Course 1 is about text formatting and may be included in word processors, spreadsheets and presentations, whereas Course 2 provides learning content for spreadsheets.

When the learner accomplishes Course 1, a statement is sent to the LRS. When he launches Course 2, the course 'asks' the LRS if that statement exists in the LRS (Figure 5.2) and if it does, it does not show the content which was included in Course 1. If the statement does not exist - i.e. the learner has not come across text formatting - the content regarding the Course 1 is displayed to the learner. For instance, if Course 1 was offered as part of a word processing lesson but the student was absent at that time or skipped that course, he should revise this content and therefore our implementation in Course 2 should provide him with the information included in the Course 1. This way, our implementation offers personalized

pathway to the learner according to his previous activities and adapts the content according to the learner's history and needs.

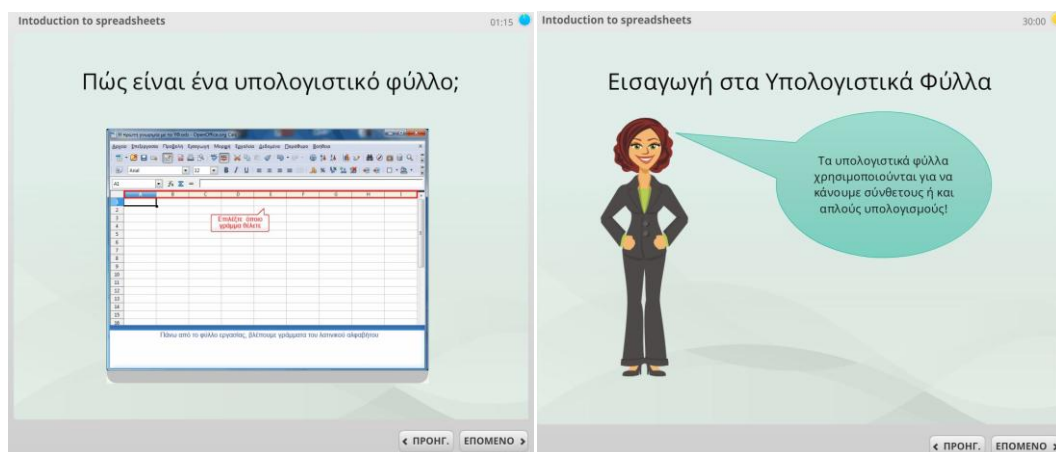
Figure 5.2 The architecture of our implementation.



A dashed box suggests that the course is not required and dashed lines that the relevant statement may not be sent to the LRS. Thick arrows show the communication between the course and the LRS regarding the existence of xAPI statement1.

The course is enriched with multimedia and images (Figure 5.3) and is very user- friendly. It contains different types of assessment (Figure 5.4, Figure 5.5) and has high interactivity with the user (Figure 5.6 , Figure 5.7.)

Figure 5.3 Interactive content of our implementation: An interactive video³ and images.



³ (taken from <http://photodentro.edu.gr/aggregator/lo/photodentro-lor-8521-960> and incorporated in our implementation)

Figure 5.4 : Different types of assessment: Multiple choice questions

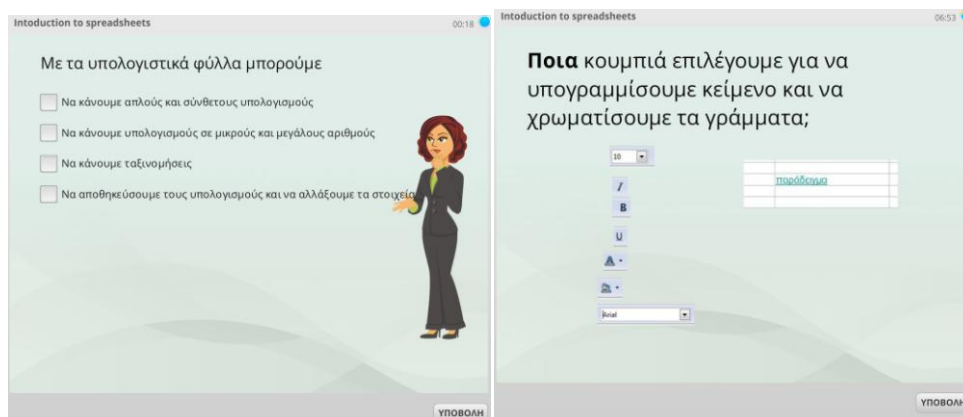


Figure 5.5 : Different types of assessment: Fill in the blanks/ Hotspot Area

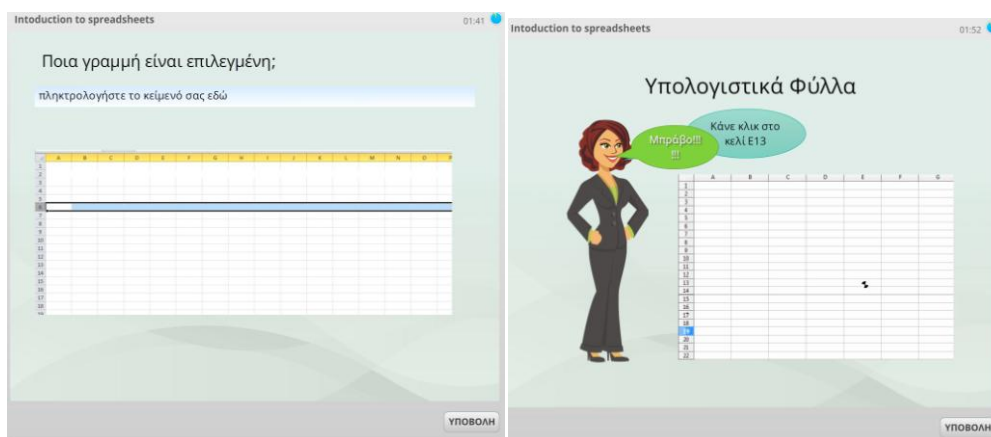


Figure 5.6 : Interactivity: The instructor changes face when he sees a wrong answer

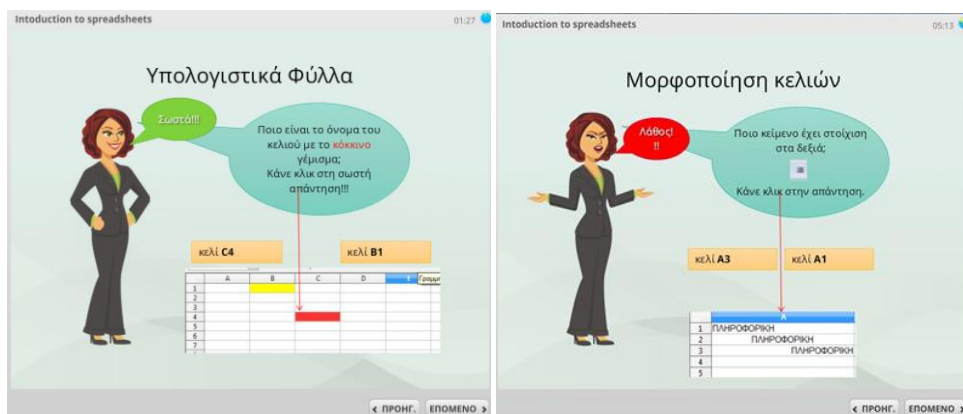
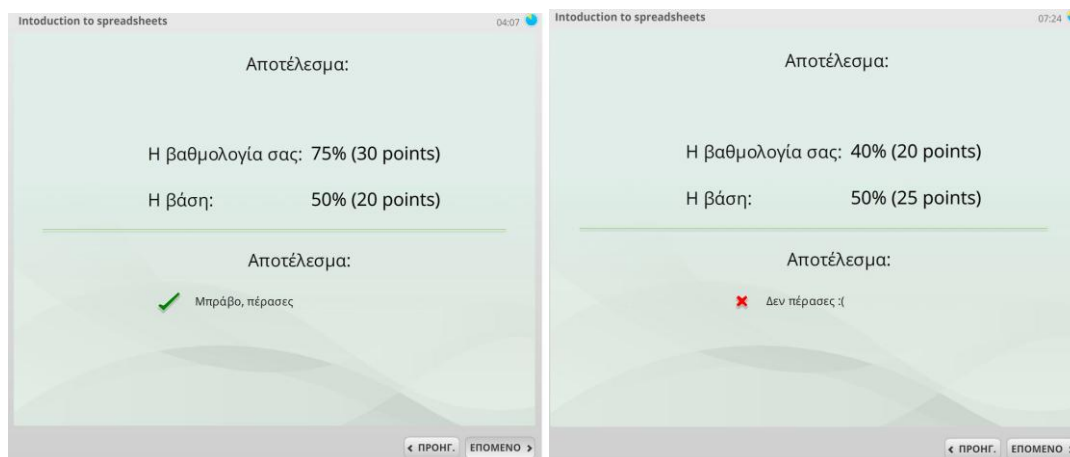


Figure 5.7 : Interactivity: The results and score of each unit



5.1.2 Structure

The courses are webpages (.html files) with JavaScript code which performs the communication between the courses and the LRS. In order to develop user-friendly courses that would be enriched with attractive interface, multimedia content and interactive quizzes, we used a demo version of Articulate Storyline 2 [21], which is popular software for creating learning content. Articulate Storyline 2 efficiently supports xAPI, but its integrated features provided one-way delivery of statements, i.e. from the course to the LRS and not vice versa, and were not sufficient in our implementation. Therefore, the tincan.js JavaScript library, described in 3.2.2, has been used to provide all the necessary functions making *bidirectional* communication between our html courses and the LRS possible. Our courses in Articulate Storyline 2 were augmented with JavaScript code calling functions to make delivery of the statements to and from the LRS possible. Moreover, the statements the xAPI statements may be sent whenever we want them and not only as results of quizzes, as Articulate 2 supports. As stated earlier, Course 1 is an optional part which might have been attempted in the past by a student and may belong to a different class. If it is completed successfully, our implementation sends a statement (via the JavaScript tincan.js function `sendStatement`) to the LRS with the indication that the student has completed Course 1. The tincan.js JavaScript function `sendStatement` is implemented via Restful HTTP PUT (or POST) method [11]. Figure 5.8 shows the statement that is sent via a PUT function towards the LRS.

Figure 5.8 An example of the delivery of the statement from Course 1 to the LRS.

```
[18/May/2017 13:30:31] "OPTIONS /xAPI/statements?statementId=e5a7d141-b4d2-4acd-8c37-8880f4b0cc02 HTTP/1.1" 200 0
[18/May/2017 13:30:32] "PUT /xAPI/statements?statementId=e5a7d141-b4d2-4acd-8c37-8880f4b0cc02 HTTP/1.1" 500 30
```

The statement is stored in our instance of ADL's LRS and is shown in JSON format below.

```
{
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/completed",
    "display": {
      "und": "completed"
    }
  },
  "version": "1.0.2",
  "timestamp": "2017-05-18T17:30:32.680Z",
  "object": {
    "id": "http://koralia/Test_in_text_Formatting",
    "objectType": "Activity"
  },
  "actor": {
    "mbox": "mailto:koraliaip@test.com",
    "objectType": "Agent"
  },
  "stored": "2017-05-18T17:30:31.282349+00:00",
  "result": {
    "completion": true,
    "score": {
      "scaled": 1
    },
    "success": true
  },
  "id": "e5a7d141-b4d2-4acd-8c37-8880f4b0cc02",
  "authority": {
    "mbox": "mailto:mtpl30@edu.teicrete.gr",
    "name": "koralia",
    "objectType": "Agent"
  }
}
```

The short format of the statement is

mailto:koraliaip@test.com (Actor)

http://adlnet.gov/expapi/verbs/completed (Verb)

http://koralia/Test_in_text_Formatting (Object)

and follows the structure of <Actor, Verb, Object> which was mentioned earlier in this paper.

When the student attempts Course 2, which might be in a posterior point in time, the student need not repeat Course 1. Therefore, our implementation sends – via JavaScript Tin Can getStatements function– a request towards the LRS asking whether the statement with the indication that the student has completed Course 1 exists. Again tincan.js JavaScript

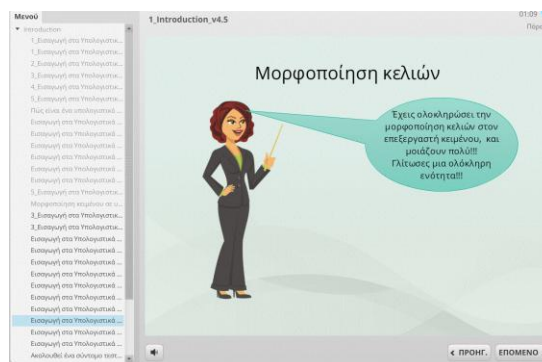
function `getStatements` is implemented via a Restful HTTP method (illustrated in Figure 5.9), the GET method [11].

Figure 5.9 An example of the request of a statement towards the LRS.

```
[18/May/2017 13:33:29] "OPTIONS /xAPI/statements?verb=http%3A%2F%2Fadlnet.gov%2Fexpapi%2Fverbs%2Fcompleted&activity=http%3A%2F%2Fkoralia%2Ftest_in_text_Formatting HTTP/1.1" 200 0
[18/May/2017 13:33:30] "GET /xAPI/statements?verb=http%3A%2F%2Fadlnet.gov%2Fexpapi%2Fverbs%2Fcompleted&activity=http%3A%2F%2Fkoralia%2Ftest_in_text_Formatting HTTP/1.1" 200 6375
[18/May/2017 13:33:44] "OPTIONS /xAPI/statements?statementId=37f2150b-d2bf-45aa-8e22-af0aa3e3656c HTTP/1.1" 200 0
```

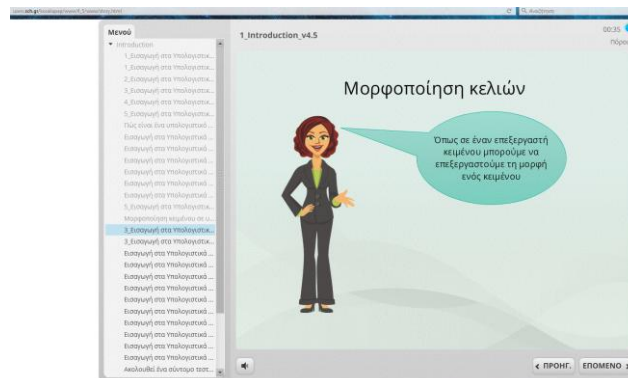
The LRS responds and if the requested statement is found, our implementation skips this part and continues with new content (Figure 5.10). If the statement is not found, our implementation shows the content of the first course as well as the new content (Figure 5.11). To avoid bottlenecks and delays, the request to the LRS has been sent at the beginning of the second course and the response is stored in a local variable.

Figure 5.10 The user has previously completed Course 1 and automatically skips the slides concerning this content. In the menu on the left, the slides concerning the content of Course 1 appear to have been skipped.



The exported courses are in html format and can be hosted in any website together or independently. They are standalone applications that need not an LMS to integrate, but instead need an LRS to communicate with. Along with the course files, the `tincan.js` file should be uploaded in the web servers, whereas the browser should support JavaScript.

Figure 5.11 If the user has not completed Course 1, he watches the slides concerning relevant content.



The second course is the main course and is attempted by all users. It consists of five units (PART_A, PART_B, PART_C, PART_D and PART_E) and four quizzes (for PART_A, PART_B, PART_C&D and PART_E) and if the user fails assessment he has to re-watch the content related to this assessment (Figure 5.12); otherwise he watches the content of the next unit (Figure 5.13)

Figure 5.12 If the user has not passed the quiz of a unit, he watches the slides concerning the specific unit.

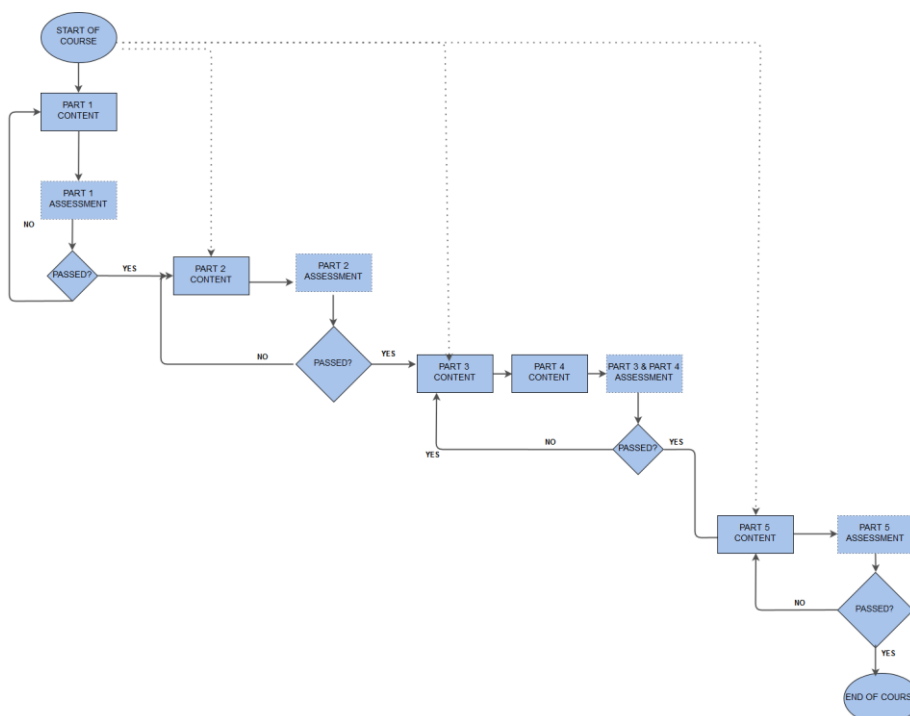


Figure 5.13 If the user has passed the quiz of a unit, he is transferred to the next unit.



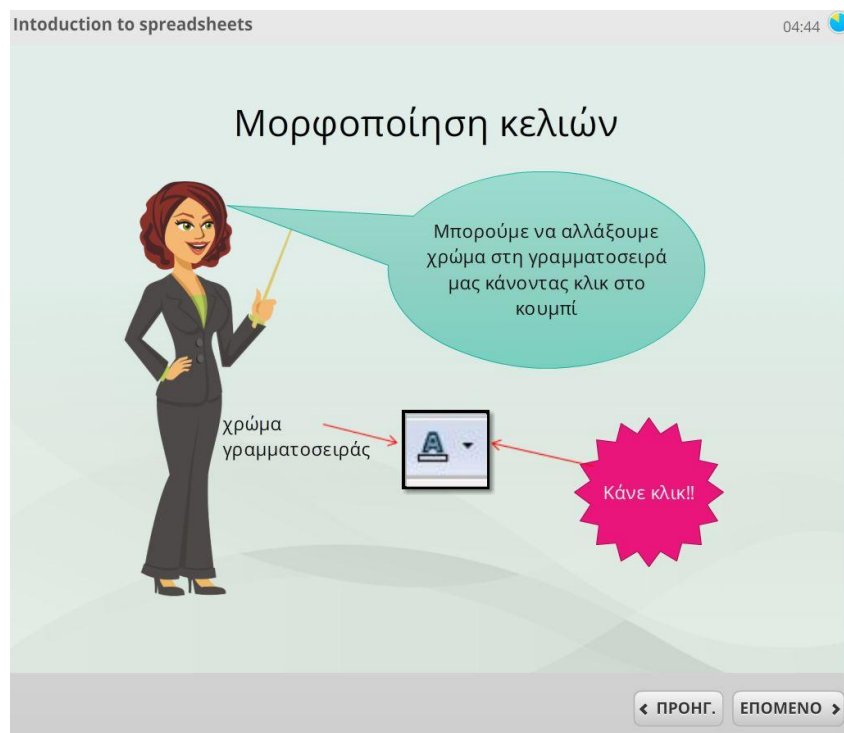
The architecture of Course 2 is shown in the Figure 5.14: every dotted line is an entrance point to our implementation. The user enters his email in the START OF COURSE. If the user stops the course, when he starts over from START OF COURSE, the application searches in the LRS which parts he has completed (based on his email) and redirects him to the next part.

Figure 5.14 The architecture of the main Course 2.



Additionally, our implementation may record several activities (e.g. clicked) in order to track specific activities and check how the user interacted with the content. That would not have been able with a traditional LMS content, where only performance in assessment is recorded. In the following example from our implementation, the learner is requested to click on the palette to see the available contents. When the user clicks the palette (Figure 5.15), an activity is fired and the statement <user> clicked <http://koralia/ColoringPalette> is sent to the LMS. Tracking such actions can help us extract valuable conclusions, especially when it comes to user- friendliness of the interface and its interaction with the user.

Figure 5.15 A statement is sent to the LRS when the user clicks on the palette.



5.1.3 Code

The tincan.js JavaScript Library described in 3.2.2 is used in this part of our implementation as Articulate supports JavaScript. The tincan.js library needs to be included in the html page we are developing via the `<script src="js/tincan.js"></script>` code.

In the following example our course connects to the LRS:

```
var tincan = new TinCan (
{
  recordStores: [
    {
      endpoint: "http://83.212.100.157:8000/xAPI/",
      username: "koralia",
      password: "*****", //**** represent the password
    }
  ]
})
```

```

        allowFail: false
    }
}
});

```

The tincan.js includes a variety of classes and methods to implement xAPI. In our implementation we have focused on the communication to and from the LRS which is achieved through the `sendStatement` and through the `getStatements()` methods.

The basic elements of the `sendStatement` action are the Actor, the Verb and the Activity. Similarly, when needed to retrieve statements we use the `tincan.js /getStatements`; `getStatements` can only retrieve statements with parameters verb and activity and therefore we need to parse the actor from the array of results.

Excerpts from the code developed for our implementation are presented here. The code is in JavaScript and is executed as a Trigger in Articulate Projects.

Excerpt 1: This excerpt of code dynamically sends a statement of PASS/FAIL to the LRS. When the user completes an assessment, his score determines whether he passed that assessment. If the score $\geq 50\%$, he passes, otherwise he fails. The following code “reads” the score and email from our implementation and sends a statement to the LRS with the verb ‘passed’ if he passed or ‘failed’ if he failed and the achieved score. It also dynamically changes the value of parameter completion and success. This excerpt examines the user’s performance in http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A and similar code sends statements to the LRS for other activities.

```

var player = GetPlayer();
var email = player.GetVar("txtEmail");

var tincan = new TinCan (
{
    recordStores: [
        {
            endpoint: "http://83.212.100.157:8000/xAPI/",
            username: "koralia",
            password: "*****", //**** represent the password
            allowFail: false
        }
    ]
}
);

var scoreT= player.GetVar("ScorePartA")/100;

if (scoreT>= 0.5) {
    var varCompletion=true;
    var varSuccess = true;
    var varId="http://adlnet.gov/expapi/verbs/passed" ;
}

```

```

else
{var varCompletion=false;
var varSuccess = false;
var varId="http://adlnet.gov/expapi/verbs/failed" ;
}
tincan.sendStatement(
{
  actor: {
    mbox: "mailto:"+email+"",
  },
  verb: {
    id: ""+varId
  },
  target: {
    id: "http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A"
  },

  "result": {
    "score": {
      "scaled" : scoreT
    },
    "completion": varCompletion,
    "success": varSuccess
  }
},
function (err, result) {

}
);

```

Excerpt 2: This excerpt of code dynamically redirects the user to the next Unit. It implements the dynamic redirection of the user according to his previous activities. If the user stops the course, when he starts over from START OF COURSE, this excerpt of code searches in the LRS which parts he has completed (based on his email) and when finds the last part he completed it redirects him to the next part. Our implementation searches sequentially into the LRS for the completion of PART_E, PART_D, PARTC, PARTB, PARTA. It uses a variable GoToUnit and changes its value according to the following cases. According to the value of the variable GoToUnit, the user is redirected to the next unit.

Activity completed	Val	Next Unit to be displayed
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E	6	Finishes course
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_CD	5	"http://koralia/Eisagwgi_sto_excel_partE"
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_B	4	"http://koralia/Eisagwgi_sto_excel_partD"
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A	2	"http://koralia/Eisagwgi_sto_excel_partB"
none found		"http://koralia/Eisagwgi_sto_excel_partA"

```

//CHECKS WHETHER THE STATEMENT EXISTS AND CHANGES THE VALUE OF A VARIABLE
ACCORDINGLY
// IS IMPLEMENTED AS EXECUTE JavaScript WHEN THE TIMELINE STARTS

```

```

var player = GetPlayer();
var email = player.GetVar("txtEmail");

var tincan = new TinCan (
{

```

```

        recordStores: [
            {
                endpoint: "http://83.212.100.157:8000/xAPI/",
                username: "koralia",
                password: "*****", //***** represent the password
                allowFail: false
            }
        ]
    }
};

var statementFound = false;

var result = tincan.getStatements({
    params: {
        verb: {
            id: "http://adlnet.gov/expapi/verbs/passed"
        },
        activity: {
            id: "http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E"
        }
    },
    callback: function (err, result) {

        var i = 0;
        while ( (i< result.statements.length) && (statementFound == false))
        {

            if (result.statements[i].actor.mbox == 'mailto:'+email)
            {
                statementFound = true;
                player.SetVar("GoToUnit", 6) ; //end
            }
            i = i + 1;
        }
        if (statementFound == true)
        {
        }
    }
});

if (statementFound == false)
{
result = tincan.getStatements({
    params: {
        verb: {
            id: "http://adlnet.gov/expapi/verbs/passed"
        },
        activity: {
            id: "http://koralia/QUIZ_Eisagwgi_sto_excel_PART_CD"
        }
    },
    callback: function (err, result) {

        i = 0;
        while ( (i< result.statements.length) && (statementFound == false))
        {

```

```

        if (result.statements[i].actor.mbox == 'mailto:'+email)
        {
            statementFound = true;
            player.SetVar("GoToUnit", 5) ;
        }
        i = i + 1;
    }
    if (statementFound == true)
    {
    }

}
});
}

if (statementFound == false)
{
result = tincan.getStatements({
    params: {
        verb: {
            id: "http://adlnet.gov/expapi/verbs/passed"
        },
        activity: {
            id: "http://koralia/QUIZ_Eisagwgi_sto_excel_PART_B"
        }
    },
},

callback: function (err, result) {

    i = 0;
    while ( (i< result.statements.length) && (statementFound == false))
    {

        if (result.statements[i].actor.mbox == 'mailto:'+email)
        {
            statementFound = true;
            player.SetVar("GoToUnit", 4) ;
        }
        i = i + 1;
    }
    if (statementFound == true)
    {
    }

}
});
}

if (statementFound == false)
{
result = tincan.getStatements({
    params: {
        verb: {
            id: "http://adlnet.gov/expapi/verbs/passed"
        },
        activity: {
            id: "http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A"
        }
    },
},

callback: function (err, result) {

    i = 0;
    while ( (i< result.statements.length) && (statementFound == false))

```

```

{

    if (result.statements[i].actor.mbox == 'mailto:'+email)
    {
        statementFound = true;
        player.SetVar("GoToUnit", 2) ;
    }
    i = i + 1;
}
if (statementFound == true)
{
}

}
});
}

```

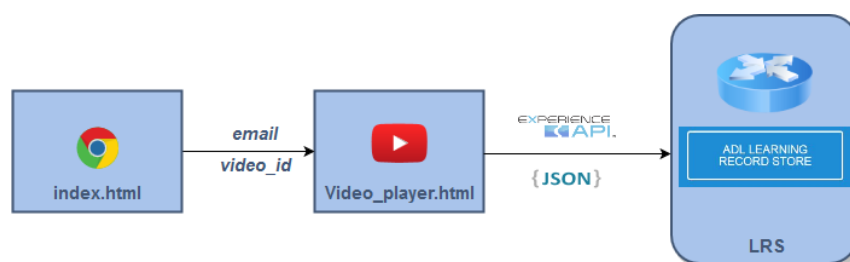
The value of variable GoToUnit is handled by Articulate Storyline to jump to next units.

5.2 A video activity tracker

5.2.1 Overview

The video activity tracker we implemented is based upon the xAPI-youtube script that has been described in 3.2.5 and has been adjusted to take the user's email and provide him with a list of videos that are related to our course. The video is played in another webpage that uses the xAPI-youtube tool and when the user starts, stops, seeks or completes a video, a statement regarding this activity is sent to the LRS. The data flow is given in Figure 5.16.

Figure 5.16 The architecture of our video activity tracker.



5.2.2 User Interface

The user inputs his email and selects a video from the list provided (Figure 5.17) in the index.html. As soon as he presses the submit button the video is loaded in Video_player.html and statements are sent according to the user's activities (Figure 5.18).

Figure 5.17 The welcome page of our video-tracking (index.html)⁴

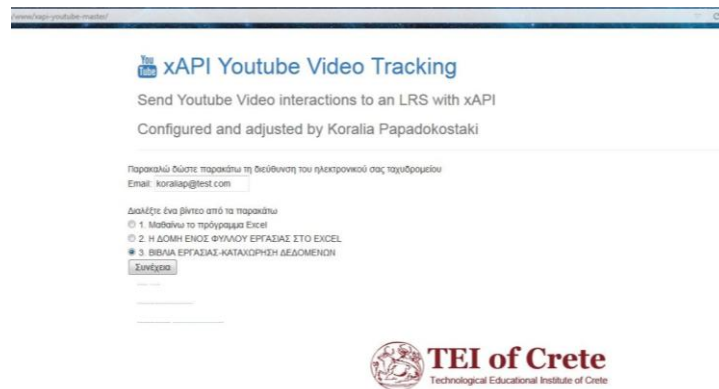
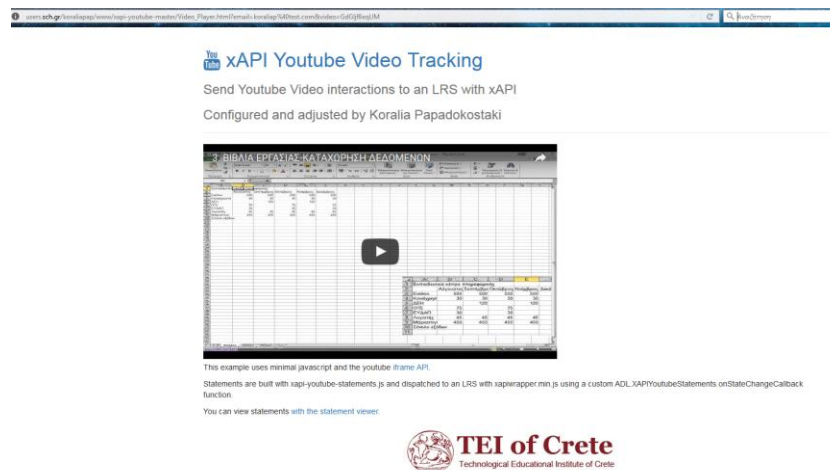


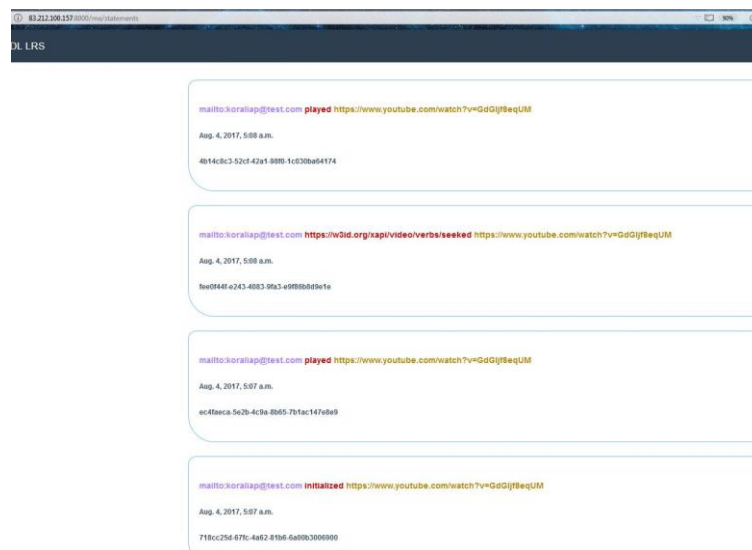
Figure 5.18 The page with the video-tracking script (Video-player.html)



When the user starts a video, plays it or seeks it xAPI statements are sent to the LRS (Figure 5.19).

⁴ Videos found at <https://www.youtube.com/watch?v=m8HHwQfvVMU>,
<https://www.youtube.com/watch?v=1qA1-CkPkxw> <https://www.youtube.com/watch?v=GdGJlf8eqUM> and
uploaded by <https://www.youtube.com/channel/UCvZhDlreAxTjBym7zCFYW3A>

Figure 5.19 The statements sent to the LRS from our video-tracking app



5.2.3 Code

This tool uses xAPI wrapper presented in 3.2.3 in order to establish connection to the LRS and the following code establishes connection and passes credentials.

```
ADL.XAPIWrapper.changeConfig({
  "endpoint": "http://83.212.100.157:8000/xAPI/",
  "user": "koralia",
  "password": "*****" //password represented by ****
});
```

Two excerpts from JavaScript code are illustrated here to display functionality added.

Excerpt 1: index.html asks the user to enter his email address and choose from one video from the list. Both the email and the video-id are passed as parameters to the next page.

```
<form action="Video_Player.html" >
  Email: <input type="email" name="email" required > <br>
  <br>
  Διαλέξτε ένα βίντεο από τα παρακάτω <br>
  <input type="radio" name="video" value="m8HHwQfvVMU" required> 1. Μαθαίνω το
  πρόγραμμα Excel<br>
  <input type="radio" name="video" value="1qA1-CkPkxw"> 2. Η ΔΟΜΗ ΕΝΟΣ ΦΥΛΛΟΥ ΕΡΓΑΣΙΑΣ
  ΣΤΟ EXCEL<br>
  <input type="radio" name="video" value="GdG1jf8eqUM"> 3. ΒΙΒΛΙΑ ΕΡΓΑΣΙΑΣ-ΚΑΤΑΧΩΡΗΣΗ
  ΔΕΔΟΜΕΝΩΝ <br>
  <input type="submit" value="Συνέχεια">
</form>
```

Excerpt 2: Video_player.html parses the parameters sent from index.html (email and video-id) and loads the requested video.

```
var parameters = location.search.substring(1).split("&");
// extracts quotes to input in variable
var temp = parameters[0].split("=");
```



```

var email = unescape(temp[1]);

// takes video id as a parameter
var temp = parameters[1].split("=");
var video = unescape(temp[1]);

myXAPI.object = {id: "https://www.youtube.com/watch?v="+video, definition: {name: {"en-US":
video}}};
myXAPI.actor = {mbox: "mailto:" + email}

```

The rest of the functionality is provided by xAPI-youtube script.

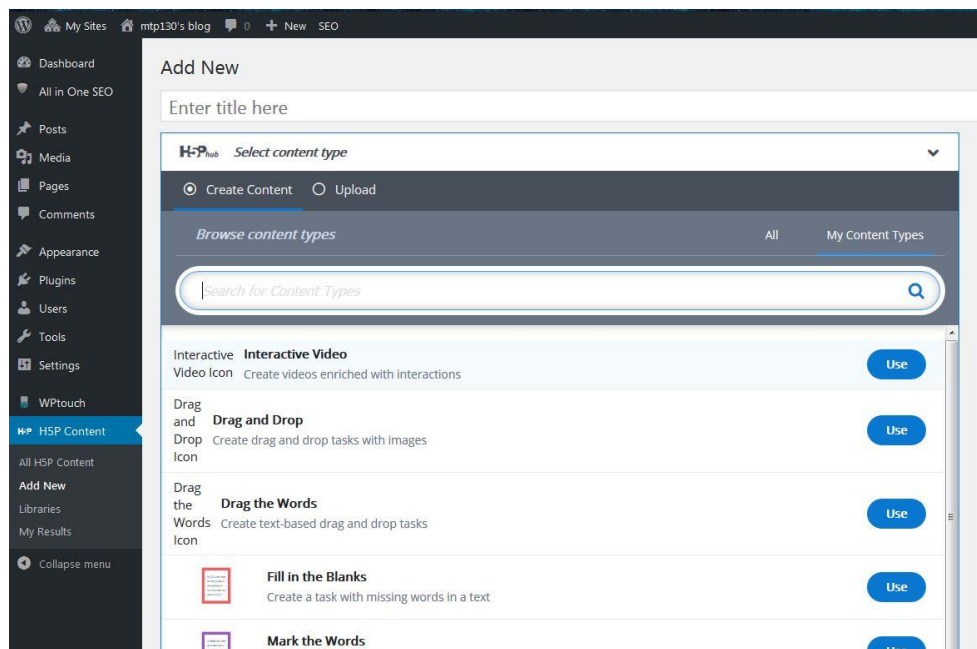
5.3 H5P xAPI-enabled content

5.3.1 Overview

H5P is an open source authoring tool that creates interactive learning content, such as videos, presentations, quizzes and games. It creates HTML5 content that is rich, responsive and mobile friendly. H5P content is currently supported by WordPress, Moodle and Drupal. It is a completely free and open technology, licensed with the MIT license [53]. As stated earlier in Table 3.2, H5P is among the adopters of xAPI and supports the communication between the content and the LRS, in order to send the activities and achievements of the user [53]. Specific interactive tools created with H5P do support xAPI, such as Interactive Videos, Course Presentation, Drag and Drop, Memory Games etc. For the installation of H5P we used the WordPress Blogs Service of Technological Institute of Crete, and for this reason the H5P plugin for WordPress was installed. H5P plugin for WordPress (Figure 5.20) allows us to use already existing H5P content by uploading to our blog, or create H5P content through the WordPress plugin⁵.

⁵ For a full list of xAPI enabled activities please visit <https://h5p.org/node/617/xapi-coverage>

Figure 5.20 The H5P plugin for WordPress



H5P also provides minimal LMS functions and may record the score of registered users of our H5P content, as shown in Figure 5.21.

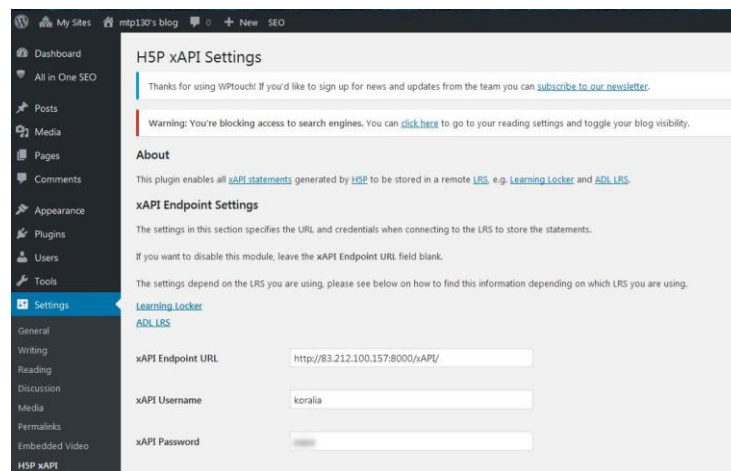
Figure 5.21 The results of registered users in WordPress for H5P content.

The screenshot shows the 'Results for A video about spreadsheets' screen in the H5P Content plugin. It displays a table with the following data:

User	Score	Maximum Score	Opened	Finished	Time spent
spenag@icloud.gr	1	1	21 July 2017 12:18	21 July 2017 12:27	0:44

As H5P becomes more and more popular and both H5P and xAPI are open source technologies, several plugins have been developed in order to facilitate the communication between H5P and xAPI. In our case, we had the wp-h5p-xapi [54] plugin installed in our WordPress instance. This plugin enables tracking xAPI enabled activities from our H5P interactive content and sends them to the LRS. The configuration of the extra wp-h5p-xapi plugin, results to the communication between the H5P content and our LRS- see Figure 5.22 for details.

Figure 5.22 Configuration of wp-h5p-xapi plugin



5.3.2 H5P content

In H5P we added interaction to a video⁶, in order to focus on certain parts of the video and check whether the user understands the content of the video. With H5P we added the following interactions:

- Labels for the user to stress out the most important information of the video
- *xAPI enabled questions* for the user to check his knowledge (An example is given in Figure 5.23)
- *xAPI enabled questions with adaptive branching*, where the user answers a question and is redirected to a specific point in the video if his answer is wrong. In our example (see Figure 5.24), the question is about the address of a cell and if the user gives the wrong answer (Figure 5.25), the content redirects him to where the cells are introduced (see the slide bar in Figure 5.26) .
- An *xAPI enabled summary quiz* of the video, in order to investigate whether the user has understood the basic concepts of the video. (Figure 5.27 and Figure 5.28).

⁶ Video taken from <https://www.youtube.com/watch?v=1qA1-CkPkxw>

Figure 5.23 A simple question on the content of the video

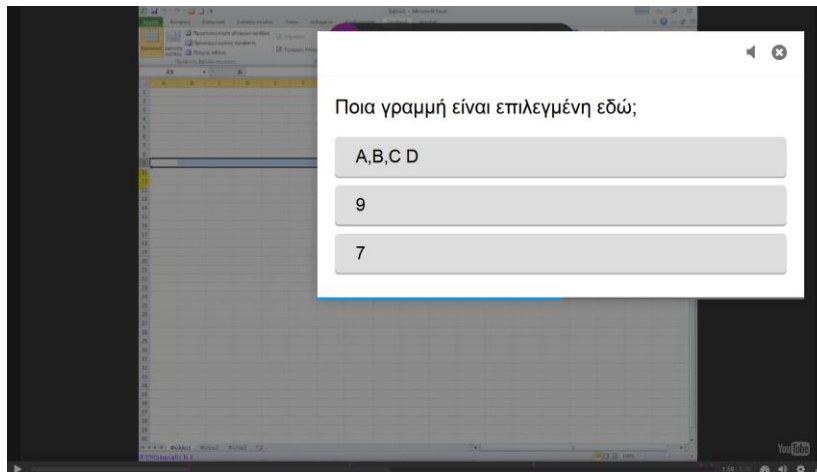


Figure 5.24 A question with adaptive branching

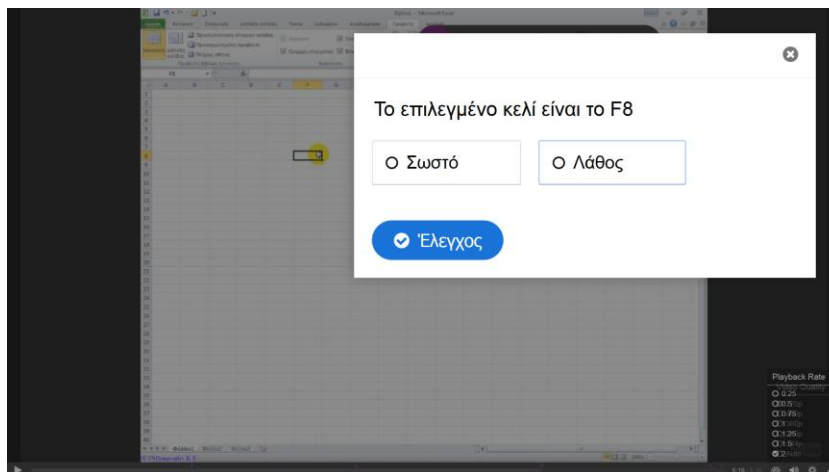


Figure 5.25 The interaction of the question

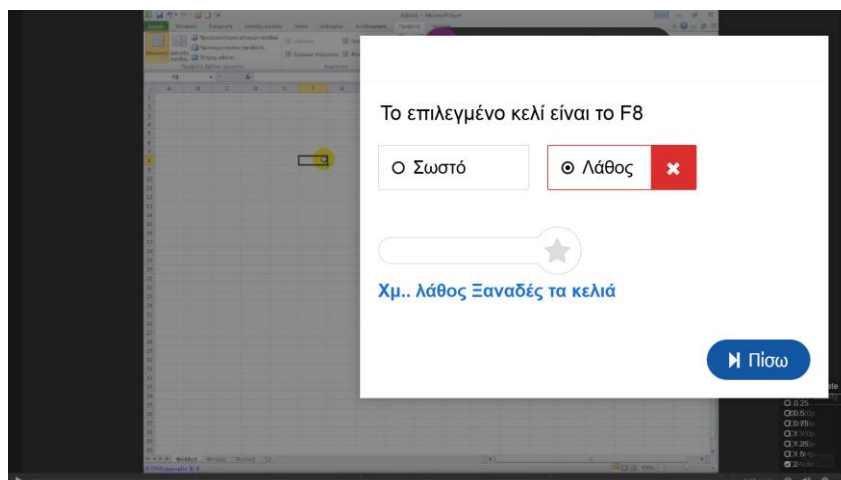


Figure 5.26 The user is redirected to the part which he did not understand.

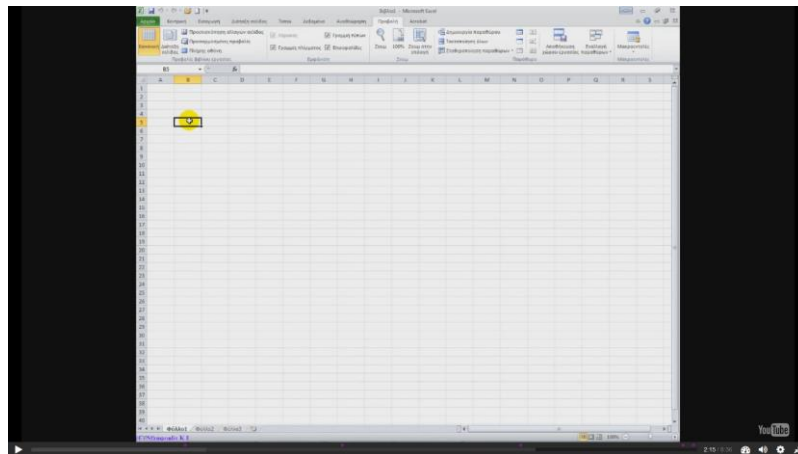


Figure 5.27 The summary quiz shows progress

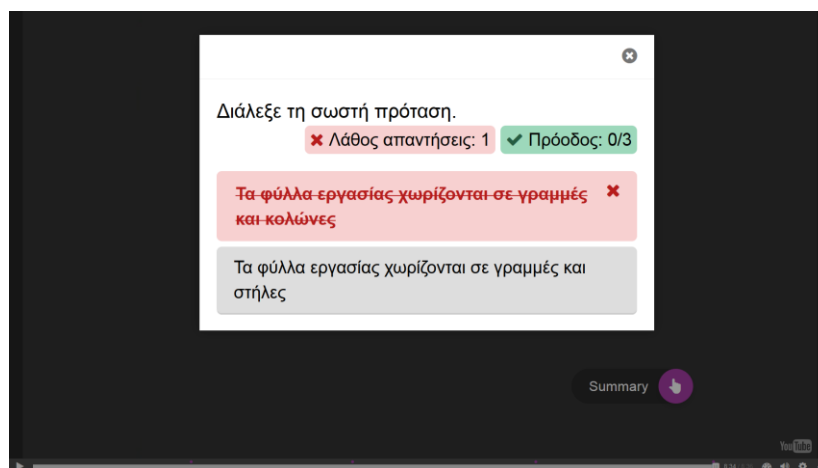
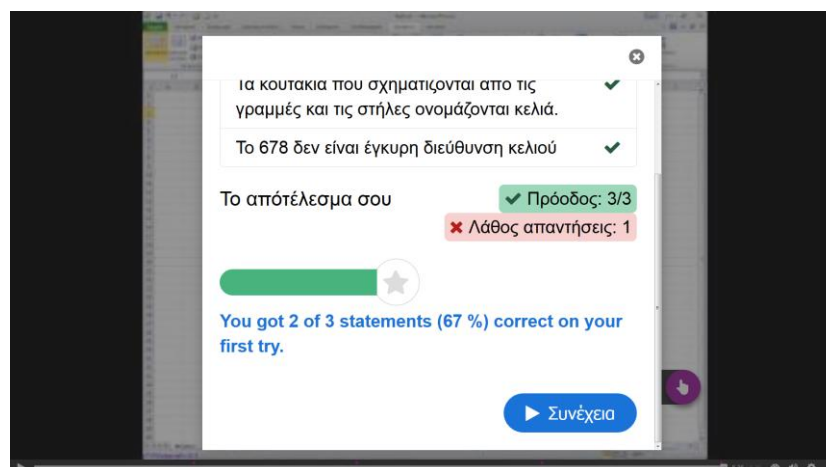


Figure 5.28 After the completion of the summary quiz, the total score is shown.



Our H5P content is hosted in <https://blog.teicrete.gr/mtp130/>

5.4 A sequencing template

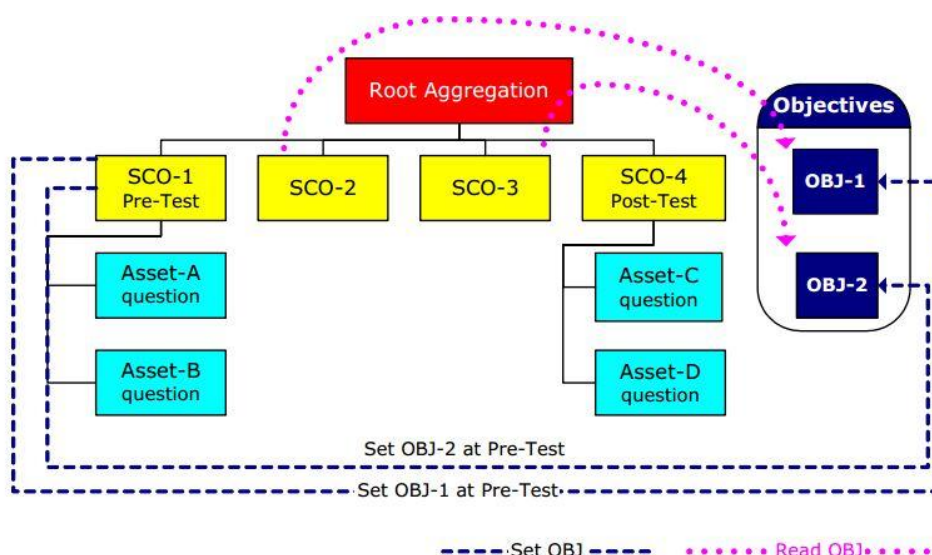
5.4.1 Overview

Although xAPI seems to overshadow SCORM, SCORM was the first successful specification to achieve reusability, interoperability, durability and accessibility to educational content. The specification introduced the term SCO (Sharable Object Content) to describe a logical unit of educational content, which might be a lesson, object or module, depending on the context. Although in instructional design, an Objective is used to measure the attainment of a skill, knowledge or skill, in SCORM, the Objective (OBJ) is a more general term.

In this thesis, we recognize the importance of SCORM and the content design principles that have been produced due to it. Therefore, we have implemented with xAPI one of the templates the [6] proposes. In particular, the No 6 Pre- and Post- Test Sequencing Template has been implemented with the use of xAPI using HTML and the tincan.js library earlier mentioned in this document.

The diagram in Figure 5.29 explains the sequencing that is implemented

Figure 5.29 The template's diagram (Figure taken from: [6]) .



Template 6 presents a sequencing option for pre- and post- tests of learner's knowledge or skill. Both pre- and post- tests are individual SCOs. The SCOs are linked to objectives that correspond to test items within the SCO. Based upon the learner's response to the pre-test item, the OBJ is either set to passed or failed. When the pre-test in SCO-1 is completed, the

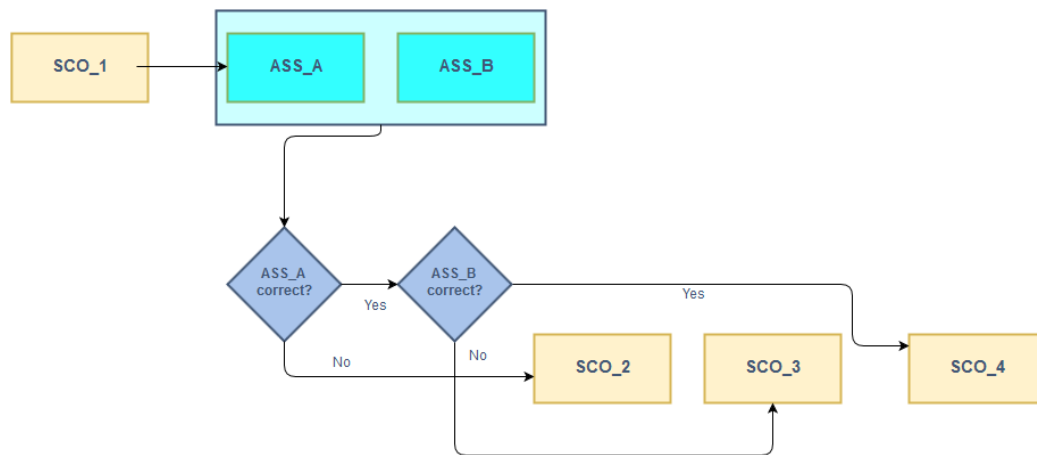
LMS shows the learner the SCOs corresponding to the missed test questions so the learner can complete the instruction before taking the post-test. If the user passes both pre-test items in SCO_1 he has the choice to skip the next SCOs or complete them and then go to SCO_4. However, he is not allowed to access the SCO_4 if he has not passed OBJ_1 and OBJ_2. The rules are illustrated in the following Figure 5.30.

Figure 5.30 The template's rules (Figure taken from: [6]).

Template 6 Rules:	
Behavior	SCORM Function
1. To complete the Root Aggregation, the learner must pass the post-test in SCO-4.	Root Aggregation Rollup: If All Satisfied, satisfied. SCO-1: isRolledup=false SCO-2: isRolledup=false SCO-3: isRolledup=false SCO-4: isRolledup=true
2. The learner must complete the pre-test in SCO-1 before attempting SCO-2 or SCO-3.	Root Aggregation: Choice=false; Flow=true
3. The learner can return to SCO-1 from SCO-2 at any time.	Root Aggregation: Forward Only=false
4. If the learner fails OBJ-1 in SCO-1, then present SCO-2.	SCO-1: set OBJ-1 SCO-2: skip if OBJ-1 satisfied
5. If the learner fails OBJ-2 in SCO-1, then present SCO-3.	SCO-1: set OBJ-2 SCO-3: skip if OBJ-2 satisfied
6. To complete SCO-4, both test items must be passed.	<i>No unique SCORM function</i>
7. If the learner fails SCO-4, then the learner is halted in training and requires manual intervention.	<i>No unique SCORM function</i>

For simplicity, we have related Asset-A question to OBJ-1 and Asset-B question to OBJ-2. In the case learner fails Asset-A, OBJ-1 is not passed (failed) and our implementation sends an xAPI statement to the LRS. In the case learner passes Asset-A, but fails Asset-B, he has passed OBJ-1 but has failed OBJ-2. For simplicity reasons again, SCO-2 is related to the content regarding Asset-A and SCO-3 related to Asset-B. According to the description of the template, both Assets must be passed for content of SCO_4 to be presented. As the statements pass to the LRS, we may stop the interaction and start again later. The flowchart of our implementation is illustrated in Figure 5.31.

Figure 5.31 The flowchart of our implementation



5.4.2 User's interface

The user is asked to input his email and to select from the four SCOs (Figure 5.32), but if he accesses a SCO he's not allowed to, he comes across a message and is navigated to the previous screen (Figure 5.33).

Figure 5.32 The User's interface for the Sequencing Template

Sequencing Template with xAPI

Sequencing Template

Created by Koralia Papadokostaki

Please input your email address

Email:

Choose one from the objects below

- ☐ 1. SCO_1 (Pre-test)
- ☐ 2. SCO_2
- ☐ 3. SCO_3
- ☐ 3. SCO_4 (Post-test)

- SCO_1 includes two assessments that need to be passed in order to attend SCO_4 (Post-test)
- The first assessment is related to OBJ-1
- The second assessment is related to OBJ-2
- SCO_4 includes two assessments
- The user may move to SCO_2 or SCO_1 any time
- The user need watch SCO_2 if he fails OBJ_1
- The user need watch SCO_3 if he fails OBJ_2

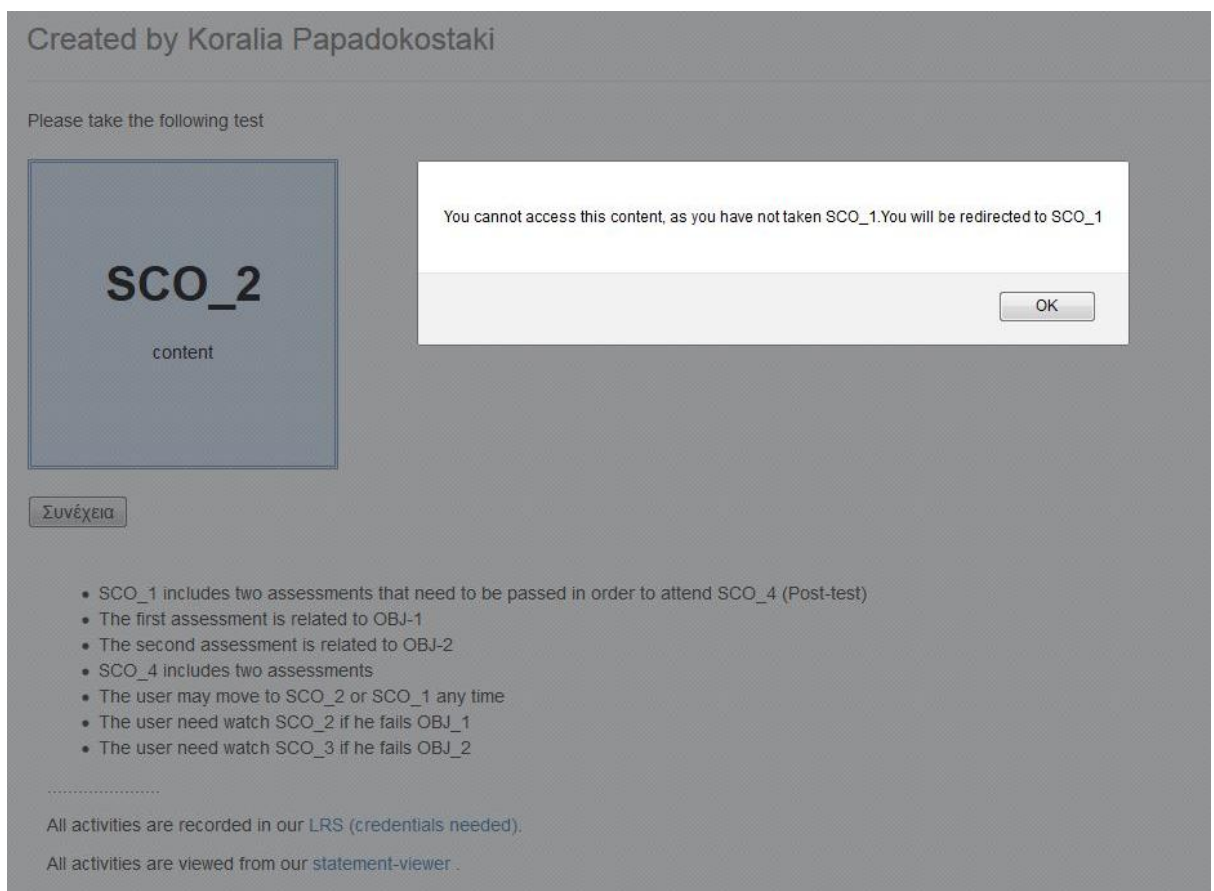
.....
All activities are recorded in our [LRS \(credentials needed\)](#).

All activities are viewed from our [statement-viewer](#) .



TEI of Crete
Technological Educational Institute of Crete

Figure 5.33 The interaction of the user with the Sequencing Template



5.4.3 Code

For the generic implementation of this template, HTML code has been used and the tincan.js library that was described in 3.2.2. Our implementation consists of four HTML pages and the navigation between them is made by a module deciding if access is possible based upon the statements saved in the LRS.

Excerpts from the code developed for our implementation are presented here.

Excerpt 1: The following module sends statements to the LRS, depending on the user's answers to ASS_A/ ASS_B.

```
function myfunction() {  
  
var parameters = location.search.substring(1).split("&");  
    // extracts email from parameters ~in this case it is the first parameter  
var temp = parameters[0].split("=");  
var email = unescape(temp[1]);  
  
console.log("parsed email");  
    //prepares connection to LRS  
var tincan = new TinCan (  
    {  
        recordStores: [  
            {
```

```

        endpoint: "http://83.212.100.157:8000/xAPI/",
        username: "koralia",
        password: "****", //password represented by ****
        allowFail: false
    }
}
}
);

tincan.sendStatement(
{
    actor: {
        mbox: "mailto:"+email+"",
    },
    verb: {
        id: "http://adlnet.gov/expapi/verbs/attempted"
    },
    target: {
        id: "http://koralia.com/SCO_1"
    }
},
function (err, result) {

});

var answer1 = form_1.answer1_1.value;
var answer2 = form_1.answer1_2.value;

//if answer1_1 was wrong sends statement to the lrs that user failed OBJ_1
if (answer1 == "1") {

tincan.sendStatement(
{
    actor: {
        mbox: "mailto:"+email+"",
    },
    verb: {
        id: "http://adlnet.gov/expapi/verbs/passed"
    },
    target: {
        id: "http://koralia.com/OBJ_1"
    }
},
function (err, result) {

});
}
else
{
tincan.sendStatement(
{
    actor: {
        mbox: "mailto:"+email+"",
    },
    verb: {
        id: "http://adlnet.gov/expapi/verbs/failed"
    },
    target: {
        id: "http://koralia.com/OBJ_1"
    }
},
function (err, result) {

});
}
document.getElementById

```

```

    //if answer1_2 was wrong sends statement to the lrs that user failed OBJ_2
    if (answer2 == "1")
    {
        tincan.sendStatement(
        {
            actor: {
                mbox: "mailto:"+email+"",
            },
            verb: {
                id: "http://adlnet.gov/expapi/verbs/passed"
            },
            target: {
                id: "http://koraliap.com/OBJ_2"
            }
        },
        function (err, result) {
        }
    );
    }
    else
    {
        tincan.sendStatement(
        {
            actor: {
                mbox: "mailto:"+email+"",
            },
            verb: {
                id: "http://adlnet.gov/expapi/verbs/failed"
            },
            target: {
                id: "http://koraliap.com/OBJ_2"
            }
        },
        function (err, result) {
        }
    );
    }
    //submits form and sends email along with other parameters
    alert("submitting form");
    //populates form_1.email with email from parameters
    document.getElementById("email").value = email;
    //next page
    document.getElementById("form_1").action = "SCO_1_results.html";
    // Setting form action to "success.php" page
    //submits form
    document.getElementById("form_1").submit(); // Submitting form
}

```

Excerpt 2: Following code searches in the LRS for the statements regarding OBJ_1 and OBJ_2. If neither Objective is passed, the content of SCO_4 is not accessible. If OBJ_1 is failed, SCO_2 is displayed, if OBJ_1 is passed but OBJ_2 is failed, SCO_3 is presented.

```

<script>

    // searches for email in the input parameters
    var parameters = location.search.substring(1).split("&");
    // extracts email from parameters
    var temp = parameters[0].split("=");
    var email = unescape(temp[1]);
    var statement1Found = false;

```

```

var statement2Found = false;
var statement3Found = false;
var statement4Found = false;
alert ("LOADED. ");
var tincan = new TinCan (
{
    recordStores: [
        {
            endpoint: "http://83.212.100.157:8000/xAPI/",
            username: "koralia",
            password: "****", //password represented by ****
            allowFail: false
        }
    ]
}
);

// will search in the LRS for a statement that shows that user passed OBJ_1 and passed OBJ_2
// will search in the LRS for a statement that shows that user failed OBJ_1, if found will
redirect to SCO_2.html
// will search in the LRS for a statement that shows that user failed OBJ_2, if found it
will redirect to SCO_3.html


//checks for OBJ_1
var result3 = tincan.getStatements({
    params: {

        verb: {
            id: "http://adlnet.gov/expapi/verbs/passed"
        },
        activity: {
            id: "http://koralia.com/OBJ_1"
        }
    },

    callback: function (err, result3) {

        var i = 0;

        while ( (i< result3.statements.length) && (statement3Found == false))
        {

            if (result3.statements[i].actor.mbox == "mailto:"+email+"")
            {
                statement3Found = true;

            }
            i = i + 1;
        }

        if (statement3Found == true)
        {

            var result4 = tincan.getStatements({
            params: {

                verb: {
                    id: "http://adlnet.gov/expapi/verbs/passed"
                },
                activity: {
                    id: "http://koralia.com/OBJ_2"
                }
            },

```

```

callback: function (err, result4) {

    var i = 0;

    while ( (i< result4.statements.length) && (statement4Found == false))
    {

        // console.log(result4.statements[i].actor.mbox + "passed OBJ_2");
        if (result4.statements[i].actor.mbox == "mailto:"+email+"")
        {
            statement4Found = true;

        }
        i = i + 1;
    }

    if ((statement4Found == true) && (statement3Found == true ))
    {
        alert ("You passed both OBJ_1 and OBJ_2. You will be redirected to the next part");
        window.location = "SCO_4.html?email="+email;
    }
    else
    {

    }

}

});

    }

    else
    {

    }

}

});

//checks for OBJ_2

//looks for a statement that shows that user failed OBJ_1,
var result1 = tincan.getStatements({
    params: {
        verb: {
            id: "http://adlnet.gov/expapi/verbs/failed"
        },
        activity: {
            id: "http://koraliap.com/OBJ_1"
        }
    },
    callback: function (err1, result1) {

        var i = 0;
        //
        while ( (i< result1.statements.length)&& (statement1Found == false))
        {

            if (result1.statements[i].actor.mbox == "mailto:"+email+"")
            {
                statement1Found = true;

            }
            i = i + 1;
        }

        if (statement1Found == true ) // αν βρεθηκε
        {
            alert ("You failed OBJ_1 and will be redirected to related content");

```

```

        window.location = "SCO_2.html?email="+email;
    }
    else
    {
//looks for a statement that shows that user failed OBJ_2,
    var result2 = tincan.getStatements({
        params: {
            verb: {
                id: "http://adlnet.gov/expapi/verbs/failed"
            },
            activity: {
                id: "http://koraliap.com/OBJ_2"
            }
        },
        callback: function (err, result2) {
            var i = 0;
            while ( (i< result2.statements.length) && (statement2Found == false))
            {
                if (result2.statements[i].actor.mbox == "mailto:"+email+"")
                {
                    statement2Found = true;

                    i = i + 1;
                }
                if (statement2Found == true)
            {
                alert ("You failed OBJ_2 and will be redirected to related content");
                window.location = "SCO_3.html?email="+email;
            }
            else
            {
            }
            }
        }
    });

}

});

</script>

```

Excerpt 3: The following JavaScript code checks the LRS whether the user has passed ASS_C and ASS_D. If he has, he has completed the aggregation, otherwise he has not.

```

<script>

    // searches for email in the input parameters
    var parameters = location.search.substring(1).split("&");
    // extracts email from parameters
    var temp = parameters[0].split("=");
    var email = unescape(temp[1]);

    var statement3Found = false;
    var statement4Found = false;
    alert ("LOADED....");
    var tincan = new TinCan (
    {
        recordStores: [
            {
                endpoint: "http://83.212.100.157:8000/xAPI/",
                username: "koralia",
                password: "*****", //password represented by ****
            }
        ]
    }
    );

```



```

        window.location = "SCO_4.html?email="+email;
    }
    }
    });

    else
    {
        alert ("You have not passed both ASS_C. You have not finished the
aggregation");
        window.location = "SCO_4.html?email="+email;
    }
    }
    });
</script>

```

Chapter 6 - Teacher's Dashboard

The Dashboard provides the teacher with four powerful tools for monitoring and visualization of the user's activities. As earlier mentioned, these tools may be used in order to provide learning analytics. More specifically they can be used to measure user's activity or performance, to track the impact of the content upon the learner, or even to investigate the difficulty of several parts of our content. As the content may come from various sources the frequency of each medium may as well be measured. To this direction, our implementation's dashboard (Figure 6.1) provides with

- Link to the built in ADL LRS interface
- A statement viewer
- Visualization tools (charts, pies and reports)
- Link to the monitoring tools that WordPress provides

All features of the Dashboard are hosted in <http://users.sch.gr/koraliapap/www/reporting/> and are available via a main page for the Dashboard (<http://users.sch.gr/koraliapap/www/reporting/index.html>) (see Figure 6.1).

Figure 6.1 The Teacher's Dashboard for our Implementation

Tracking Learning Activities with xAPI

DASHBOARD FOR THE TEACHER

A Master Thesis Implementation

Koralia Papadokostaki, BSc in Informatics

- All activities are recorded in our [LRS \(credentials needed\)](#).
- [Report for our LRS](#) created with ADL's xAPI Dashboard by Koralia Papadokostaki.
- [Report for the Activities of the LRS](#) created with ADL's xAPI Dashboard by Koralia Papadokostaki.
- [Report for the Users of the LRS](#) created with ADL's xAPI Dashboard by Koralia Papadokostaki.
- [Report for the Users of the LRS that passed Activities](#) created with ADL's xAPI Dashboard by Koralia Papadokostaki.
- ADL's LRS instance created by ADL, enriched by Koralia Papadokostaki.
- [Statement viewer](#) based on ADL's xapi-statement-viewer, adapted & configured by Koralia Papadokostaki
- [Link to Blog](#) (H5P WordPress Monitoring) hosted by Koralia Papadokostaki

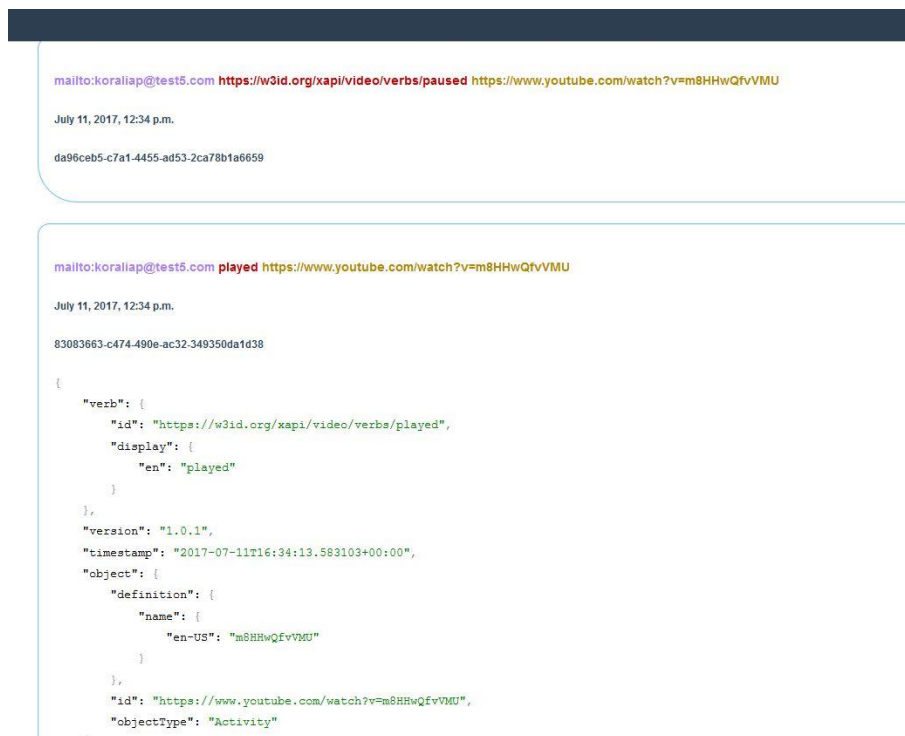
All activities are recorded in our [LRS \(credentials needed\)](#) which is an Instance of ADL's LRS installed in our VM in Cyclades.

All implementations are hosted in our [site](#) in SCH.gr

6.1 ADL LRS interface

ADL LRS provides with a statement viewer that can be used for the monitoring of the statements. It shows a list of statements with chronological order from the most recent to older ones. When clicking on each statement the JSON format of the statement is shown (see Figure 6.2). Unfortunately ADL LRS's interface does not support dynamic searching or filtering on the statements.

Figure 6.2 The interface of ADL LRS for the administrator



6.2 xAPI statement viewer

6.2.1 Overview


Since ADL LRS does not provide with a tool for filtering and searching of the statements, we had to provide our implementation with a multifunctional statement viewer. For that reason, we modified the ADL's xAPI-Statement-Viewer [52] in order to provide more fields for searching and to make the display of our format more practical. The ADL's xAPI-Statement-Viewer pulls xAPI Statements from a Learning Record Store (LRS) using the xAPI wrapper (see 3.2.3) and displays them in user-friendly table, whereas it also offers filtering and sorting options. The filters provided were the verb, the mail of the user, filters for the date of the statement etc.


However, in order to provide more options in filtering, we modified the original ADL's xAPI-Statement-Viewer and added an extra filtering field which refers to the activity name. This way we may easily extract the conclusion as to who interacted with a specific object (Quiz, video or interaction). The xAPI statement viewer was also configured with the credentials for our LRS and with verbs where the display field was not en-us.

6.2.2 User Interface

The Teacher has access to this form, in order to extract valuable information regarding his course and students. He may select the filters under the banner Query Options and fill in the ones he wants. In the Figure 6.3 statements regarding the activity `http://koraliap.com/SCO_1` are presented in chronological order. The teacher may combine filters, e.g. he may see which students passed a specific quiz or completed a video.

Figure 6.3 The modified xAPI Statement Viewer

 **xAPI Statement Viewer**

 Configured and modified by Koralia Papadokostaki

Statement List [Get Statements](#)

Query Options

Predefined Verb

Verb Sort

Descending

Verb ID

http://adlnet.gov/verbs/answered

Agent

{"mailbox": "mailto:koraliap@hotmail.com"}

Related Agents

Activity ID

http://adlnet.gov/activities/question

Related Activities

Registration ID

51a6f860-1997-11e3-8ffd-0800200c9a66

Statement ID

e18a3c05-7c88-4d46-a821-c812f061ef23

Voided Statement ID

00d9f122-e9cd-4ea3-b180-9daa49c8a7e5

Object Name (Added For Our Implementation)

http://koraliap.com/SCO_1

Since Date

Until Date

Limit

25

xAPI Query

http://83.212.100.157:8000/xAPI/?format=exact&activity=http%3A%2F%2Fkoraliap.com%2FSCO_1







[Clear Form](#)

Show

25

 entries

Search:

Timestamp	Actor Name	Verb	Object Name	Object Type	Authority Name	
23 days ago	kostas@teicrete.gr	attempted	http://koraliap.com/SCO_1	Activity	koralia	
23 days ago	kostas@teicrete.gr	attempted	http://koraliap.com/SCO_1	Activity	koralia	
23 days ago	kostas@teicrete.gr	attempted	http://koraliap.com/SCO_1	Activity	koralia	
23 days ago	spanag@teicrete.gr	attempted	http://koraliap.com/SCO_1	Activity	koralia	
23 days ago	spanag@teicrete.gr	attempted	http://koraliap.com/SCO_1	Activity	koralia	
23 days ago	spanag@teicrete.gr	attempted	http://koraliap.com/SCO_1	Activity	koralia	

6.2.3 Code

The xAPI Statement Viewer consists of a webpage (index.html), the xAPI wrapper, Bootstrap files and css libraries. The script that implements the retrieval and display is in the js/main.js script. Two modifications were added to meet our requirements.

- A. In the webpage index.html an extra text field was added with default value 'http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A' and name search-object-id

```
<div class="row">
<div class="form-group col-md-6">
<label class="control-label" for="search-object-id">Object Name (Added For Our
Implementation)</label>
<input type="text" class="form-control" value="" aria-describedby="Object ID"
placeholder="http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A" id="search-object-id"
pattern="[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}" />
</div>
</div><!-- .row -->
```

- B. In the js/main.js function modifications were also made to include the new filtering field. The code added is highlighted.

```
function getStatementsWithSearch(more, curPage) {
    var verbSort = $("#search-verb-sort").val();
    var verbId = $("#search-user-verb-id").val();
    var actor = $("#search-actor").val();
    var relatedAgents = $("#search-related-agents").val();
    var activityId = $("#search-activity-id").val();
    var relatedActivities = $("#search-related-activities").val();
    var registrationId = $("#search-registration-id").val();
    var statementId = $("#search-statement-id").val();
    var objectId = $("#search-object-id").val();
    var voidedStatementId = $("#search-voided-statement-id").val();
    var sinceDate = $("#search-statements-since-date input").val();
    var untilDate = $("#search-statements-until-date input").val();
    var limit = $("#search-limit").val();

    // Build Search
    var search = ADL.XAPIWrapper.searchParams();
    if (verbId != "") { search['verb'] = verbId; }
    if (verbSort != "") { search['ascending'] = verbSort; }
    if (actor != "") { search['agent'] = actor; }
    if (relatedAgents != "") { search['related_agents'] = relatedAgents; }
    if (activityId != "") { search['activity'] = activityId; }
    if (relatedActivities != "") { search['related_activities'] = relatedActivities; }
    if (registrationId != "") { search['registration'] = registrationId; }
    if (statementId != "") { search['statementId'] = statementId; }
    if (objectId != "") { search['activity'] = objectId; }
    if (voidedStatementId != "") { search['voidedStatementId'] = voidedStatementId; }
    if (sinceDate != "") { search['since'] = sinceDate; }
    if (untilDate != "") { search['until'] = untilDate; }
    if (limit != "") { search['limit'] = limit; }
```

6.3 Visualization

6.3.1 Overview

With the use of xAPI Dashboard, which was shortly described in 3.2.4, we created four reports that visually represent important information about our content, activities and users.

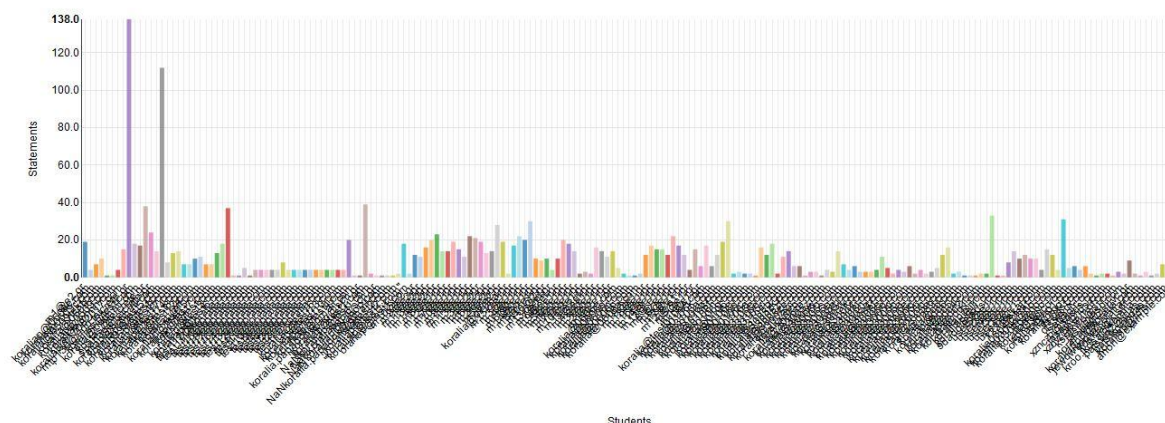
A. A report regarding the LRS, shows the frequency of use of the LRS, consisting of
three parts:

- The first graph illustrates the activity per user and indicates how active each user has been in our Application (Figure 6.4), by illustrating how many statements have been sent to the LRS, regarding the specific user.

Figure 6.4 The Report for the LRS (Activity Per user bar chart).

This Monitoring Tool provides information about the LRS.

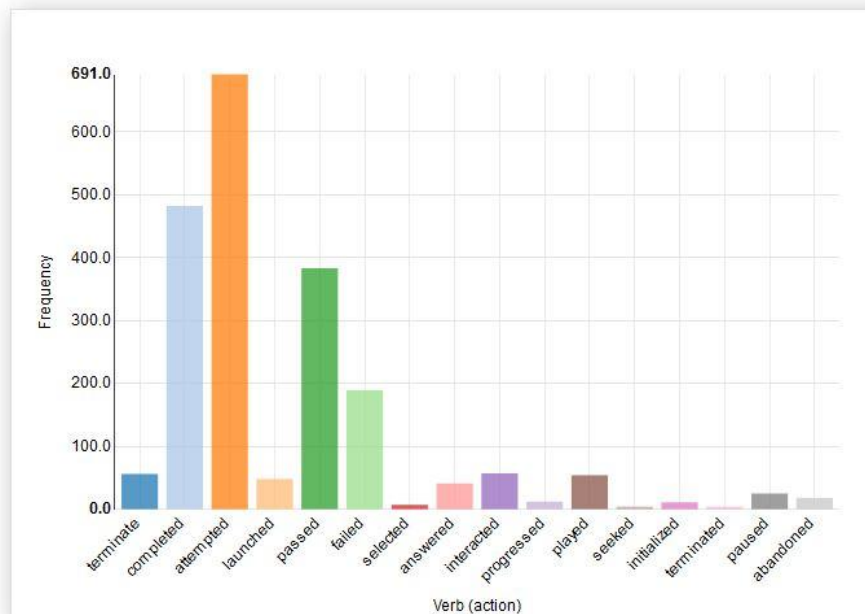
Activity per user (Number of Statements in the LRS)



- The second graph is a bar chart indicating the popularity of the verbs in our LRS, by counting the number of statements per verb. This is useful in order to compare, for example passed vs. failed or attempted vs. completed activities (Figure 6.5).

Figure 6.5 The Report for the LRS (Frequency of verbs bar chart).

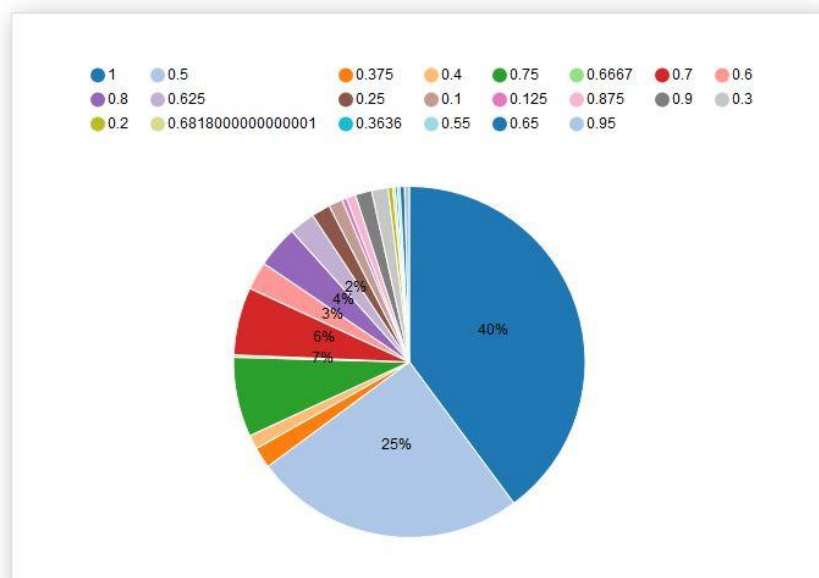
Frequency of verbs (Number of statements in the LRS)



- The third graph shows the distribution of score for all activities in our LRS. It illustrates the average score in a piechart and the percentage of statements with this score. This helps us gather information about the general performance of our students in the LRS (Figure 6.6).

Figure 6.6 The Report for the LRS (Distribution of score).

Distribution of score in the LRS (Percentage per score achieved)



B. A report regarding specific **Activities** displays visually information about certain parts of our course, e.g. a quiz. It includes a table and a bar chart with the average score per Activity. This is very useful, as displays the average score of each activity, as the teacher may detect difficult quizzes, vs. easy ones just with one look (Figure 6.7 , Figure 6.8).

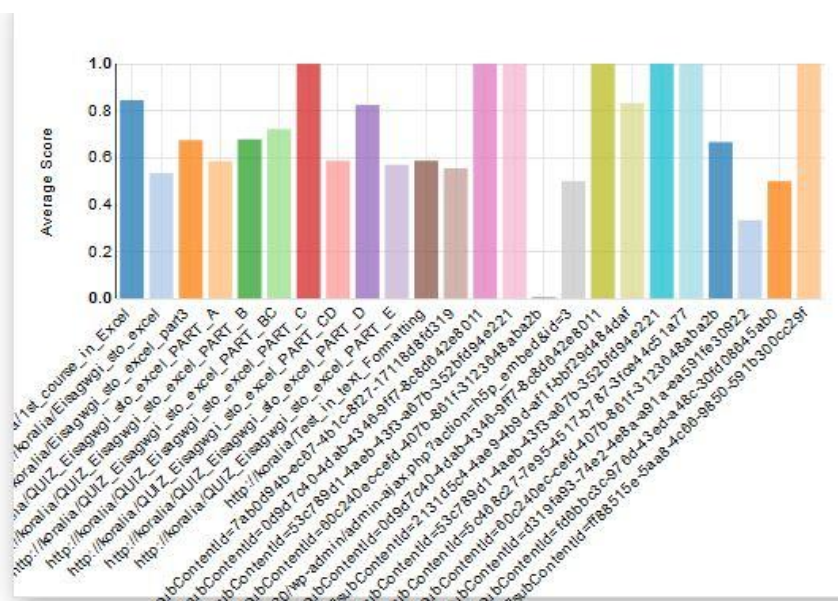
Figure 6.7 The Table with the average score per Activity.

Aggregate Data for our Activities

Average Score per Activity

object.id	result.score.scaled
http://koralia/1st_course_in_Excel	0.8454545454545453
http://koralia/Eisagwgi_sto_excel	0.534118987341772
http://koralia/Eisagwgi_sto_excel_part3	0.675
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A	0.5851612903225807
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_B	0.6785714285714286
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_BC	0.7222222222222222
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_C	1
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_CD	0.5870967741935483
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_D	0.825
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E	0.57
http://koralia/Test_in_text_Formatting	0.5873015873015873
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=1?subContentId=7ab0d94b-ec67-4b1c-8f27-17118d8f319	0.5555555555555556
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=2?subContentId=0d9d7c40-4dab-4346-9ff7-8c8d642e8011	1
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=2?subContentId=53c789d1-4aeb-43f3-a67b-352b6f94e221	1
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=2?subContentId=60c240ec-cefd-407b-861f-3123648aba2b	0
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3	0.5
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=0d9d7c40-4dab-4346-9ff7-8c8d642e8011	1
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=2131d5c4-4ae9-4b9d-af1f-bbf29d484daf	0.83333
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=53c789d1-4aeb-43f3-a67b-352b6f94e221	1
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=5c468c27-7e95-4517-b787-3fce44c51a77	1
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=60c240ec-cefd-407b-861f-3123648aba2b	0.6666666666666666
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=d319fa93-74e2-4e8a-a91a-ea591fe30922	0.3333333333333333
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=fd6b3c-976d-43ed-48c-30f008645ab0	0.5
https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=f88515e-5aa8-4c66-9850-591b300cc29f	1

Figure 6.8 The Barchart with the average score per Activity.

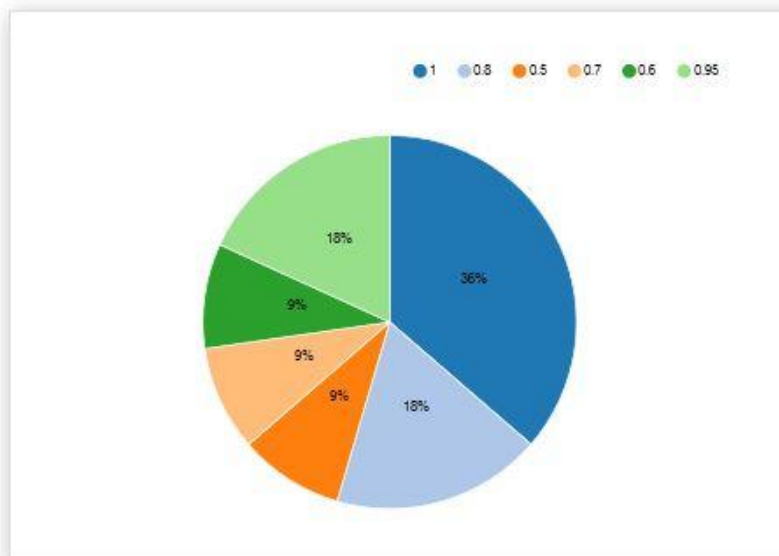


When the teacher clicks on a specific bar, regarding a specific activity the distribution of the score for this activity is illustrated in two graphs: a pie and a bar chart, indicating the percentage of statements for the score (Figure 6.9).

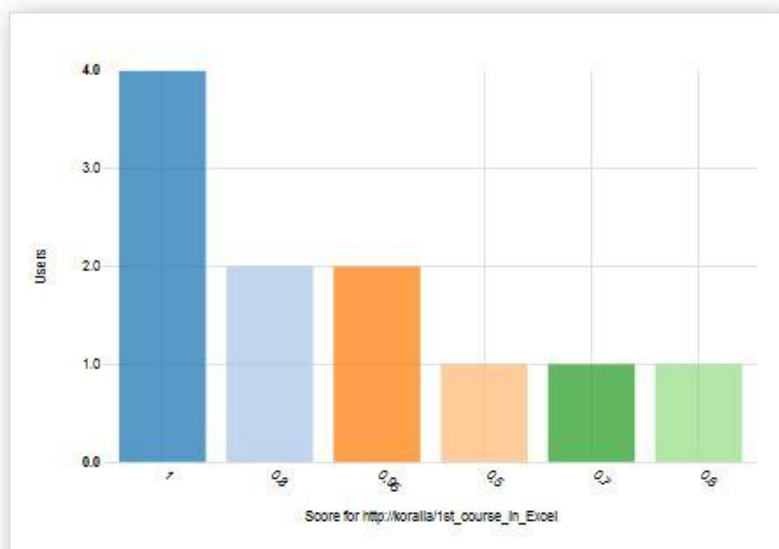
Figure 6.9 The Report for the Activities (Average Score bar and pie chart).

Distribution of score (Percentage for each score achieved)

Data for activity: http://koralia/1st_course_in_Excel



Distribution of score (Frequency of score)



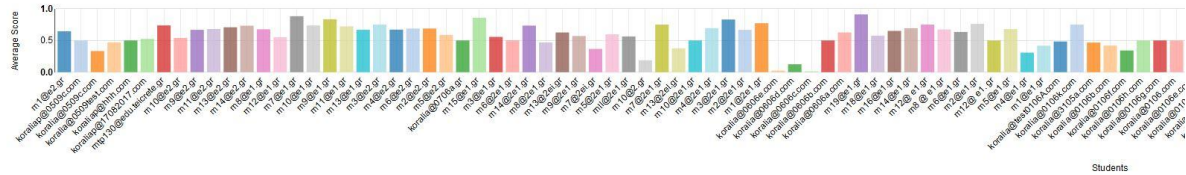
- C. A report regarding specific users, displaying the average score for all the users of the LRS and for all his activities (Figure 6.10).

Figure 6.10 The Report for the Users (Average Score per User).

Average score per student for all his activities

Activities/ users without score are not illustrated in this graph

Click on the bar of a user to see his statistics

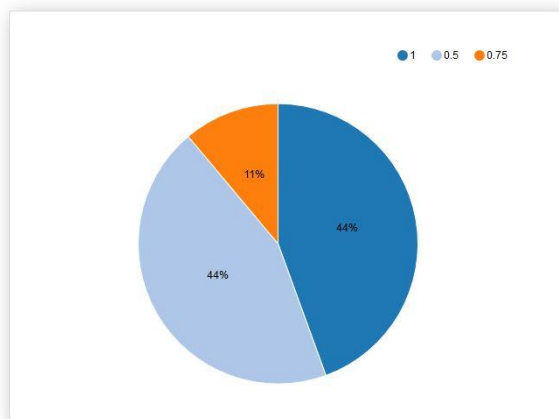


When the teacher clicks on a specific bar, regarding a specific user the distribution of the score for this user is illustrated in a pie and a table, indicating the percentage of scores for the user (Figure 6.11).

Figure 6.11 The Report for the Users (distribution per score pie chart and table).

Data for user: <mailto:m3@e2.gr>

Distribution of score (piechart)

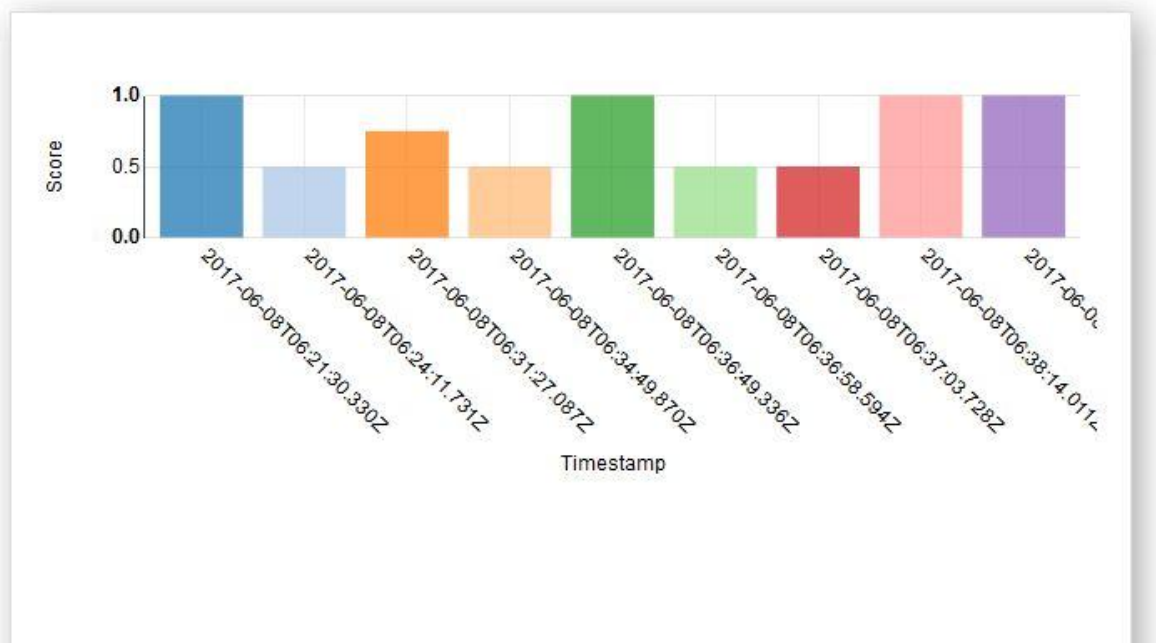


timestamp	object.id	verb.display.und	result.score.scaled
2017-06-08T06:37:03.728Z	http://koralia/Eisagwgi_sto_excel	completed	0.5
2017-06-08T06:36:58.594Z	http://koralia/Eisagwgi_sto_excel	completed	0.5
2017-06-08T06:24:11.731Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A	passed	0.5
2017-06-08T06:31:27.087Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_B	passed	0.75
2017-06-08T06:34:49.870Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_CD	passed	0.5
2017-06-08T06:38:14.011Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E	passed	1
2017-06-08T06:36:49.336Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E	passed	1
2017-06-08T07:11:08.210Z	http://koralia/Test_in_text_Formatting	completed	1
2017-06-08T06:21:30.330Z	http://koralia/Test_in_text_Formatting	completed	1

The score is also displayed in a chronological bar chart to illustrate the user's progress throughout the courses (Figure 6.12).

Figure 6.12 The Report for the Users (Progress of user).

Progress of score (barchart)



D. A report regarding users and the activities they **passed**, similar to C, with an additional table with aggregate data (Average, minimum and maximum score) about the users' scores (Figure 6.13).

Figure 6.13 The Report for the Users that passed (table with aggregate data).

Minimum, Maximum and Average score per user

actor.mbox	count	min	max	average	verb.display.und
mailto:m14@2e1.gr	3	0.5	1	0.7333333333333334	passed
mailto:m13@2e1.gr	2	0.5	0.75	0.625	passed
mailto:m9@2e1.gr	4	0.5	1	0.875	passed
mailto:m8@2e1.gr	2	0.5	1	0.75	passed
mailto:m7@2e1.gr	2	0.5	0.5	0.5	passed
mailto:m5@2e1.gr	4	0.5	0.75	0.5875	passed
mailto:m11@2e1.gr	4	0.5	1	0.625	passed
mailto:m6@2e1.gr	4	0.5	0.75	0.6125	passed
mailto:m10@2e1.gr	1	0.5	0.5	0.5	passed
mailto:m10@2e1.gr	1	0.5	0.5	0.5	passed
mailto:m3@2e1.gr	4	0.5	1	0.825	passed
mailto:m2@2e1.gr	2	1	1	1	passed
mailto:m1@2e1.gr	4	0.5	1	0.8125	passed
mailto:m4@2e1.gr	4	0.5	1	0.825	passed
mailto:m19@e1.gr	5	0.75	1	0.9099999999999999	passed
mailto:m11@e1.gr	5	0.5	1	0.9	passed
mailto:m18@e1.gr	5	0.5	1	0.64	passed
mailto:m16@e1.gr	5	0.5	1	0.64	passed
mailto:m15@e1.gr	5	1	1	1	passed
mailto:m13@e1.gr	4	1	1	1	passed
mailto:m14@e1.gr	5	0.5	1	0.6900000000000001	passed
mailto:m10@e1.gr	4	0.7	1	0.8625	passed
mailto:m9@e1.gr	2	0.5	1	0.75	passed

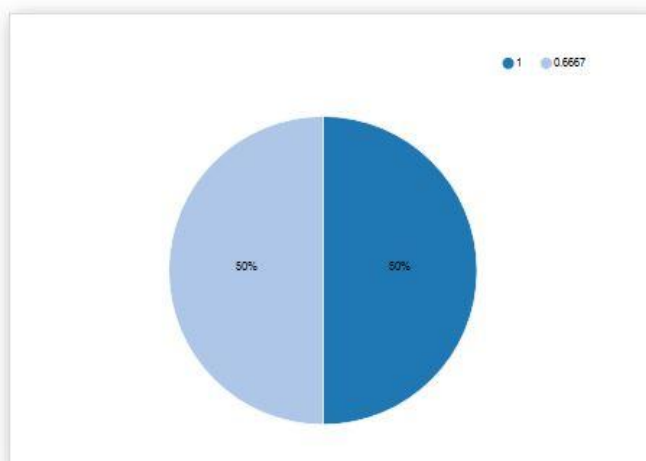
Additionally, specific reports were constructed to show the statistics of the tools created. The following charts illustrate the score for a specific part in our H5P content.

Figure 6.14 illustrates actions regarding H5P content.

Figure 6.14 The Report for an action regarding the H5P content.

Distribution of score (Percentage for each score achieved)

Data for activity: https://blog.teicrete.gr/mtp130/wp-admin/admin-ajax.php?action=h5p_embed&id=3?subContentId=2131d5c4-4ae9-4b9d-af1f-bbf29d484daf



Distribution of score (Frequency of score)

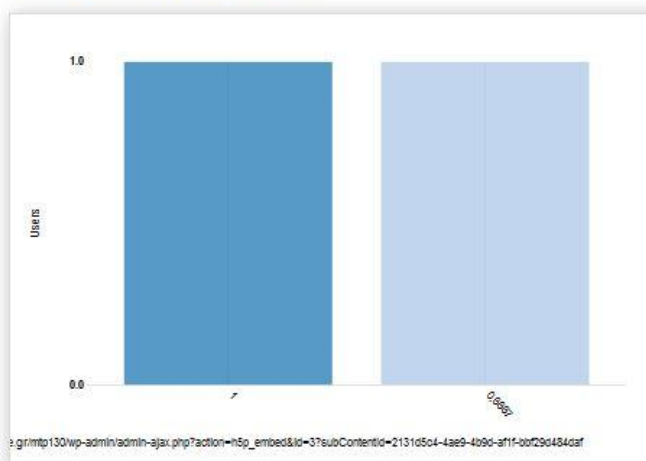


Figure 6.15 shows the activity per user in one xAPI- enabled video and Figure 6.16 the frequency of the verbs regarding the specific video.

Figure 6.15 The Report for the Users that interacted with the xAPI-enabled video .

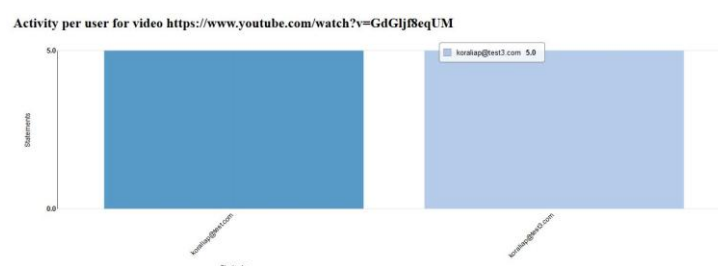


Figure 6.16 The Report for the verbs regarding an xAPI-enabled video .

Frequency of verbs for <https://www.youtube.com/watch?v=GdGlf8eqUM>

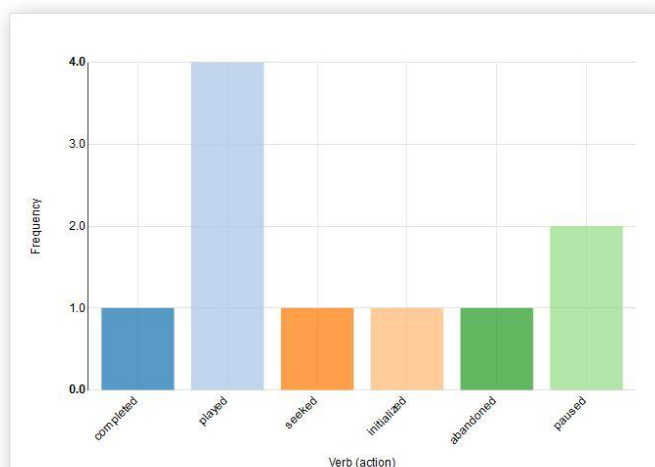


Figure 6.17 shows the activity per user regarding our Sequencing Template and Figure 6.18 shows the frequency of the verbs ‘passed’ and ‘failed’ for OBJ_1.

Figure 6.17 The Report for the Users that interacted with our Sequencing Template and specifically with OBJ_1.

This Monitoring Tool provides information about http://koraliaip.com/OBJ_1

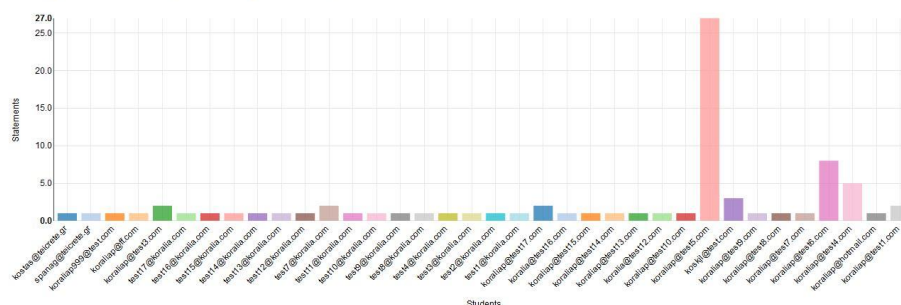
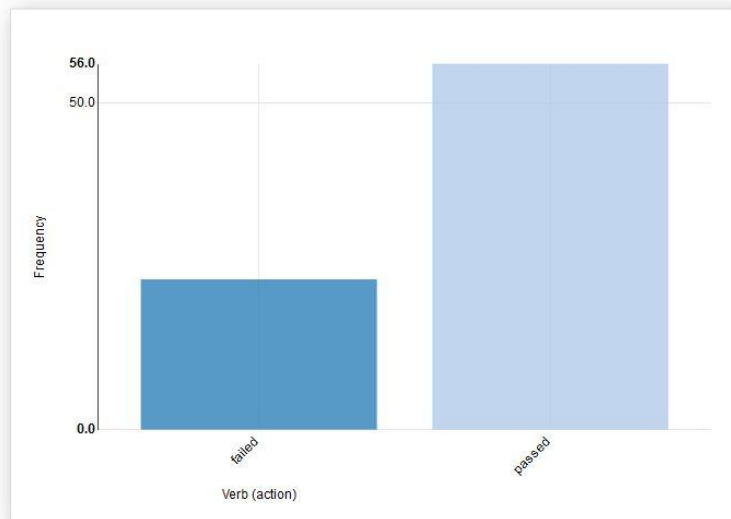
Activity per user for http://koraliap.com/OBJ_1

Figure 6.18 The Chart showing the frequency of ‘passed’ vs ‘failed’ in OBJ_1

Frequency of verbs for http://koraliap.com/OBJ_1



6.3.2 Code

Excerpts of code for the development of Charts and Reports are presented here. They are implemented with xAPI Dashboard.

Excerpt 1: This snippet of code illustrates the activity for each user (function graphActors) and a graph grouping by verbs (graphActions). It is used in Report A.

```
function graphActors() {
    //creates a graph where the statements are grouped by Actors
    (activity of users)

    var actors = dash.createBarChart({
        container: '#chart1',
        groupBy: 'actor.mbox', //groupby to mbox
        aggregate: ADL.count(),
        customize: function (chart){ //customized to rotate labels
            chart.xAxis.rotateLabels(-45);
            chart.xAxis.axisLabel("Students");
            chart.xAxis.tickFormat(function(d) {
                return /^[^:]+$/.exec(d)[0]; });
            chart.yAxis.axisLabel("Statements");
            chart.height(400);
        }
    });
    actors.clear();
}
```

```

        actors.draw();

    }

    function graphActions() {
    //creates a graph where the statements are grouped by actions (frequency of
actions)

    var actors = dash.createBarChart({
        container: '#chart2',
        groupBy: 'verb.id', //groupby verbid
        aggregate: ADL.count(),
        customize: function (chart){ //customized to rotate labels
        chart.xAxis.rotateLabels(-45);
        chart.xAxis.tickFormat(function(d){
        return /[^\./]+\$/ .exec(d) [0]; });
        chart.xAxis.axisLabel("Verb (action)");
        chart.yAxis.axisLabel("Frequency");

        }

    });
    actors.clear();
    actors.draw();

}

```

Excerpt 2: This snippet of code is used in graph C and illustrates the cascading flow of the graphs. The Barchart creates the bar chart which displays the average score for each student. When a specific bar is clicked, the scoreRangesPie creates the pie with the distribution of the score, the StudentsTable creates the table with the activities and score per student and the scoreRanges creates a bar chart with chronological order.

```

function drawCharts() {

    //shows average score of each student
    var Barchart = dash.createBarChart({
    pre: function(data, event){

        return data.where('result.score.scaled != null');

    },
    container: '#multi',
    aggregate: ADL.average('result.score.scaled'),
    groupBy: 'actor.mbox',
    customize: function (chart){ //customized to rotate labels
    chart.xAxis.rotateLabels(-45);
    chart.xAxis.axisLabel("Students");
    chart.xAxis.tickFormat(function(d){
    return /[^\.:]+\$/ .exec(d) [0]; });

}

```

```

        chart.yAxis.axisLabel("Average Score");
        chart.height(200);
    },

    });
    Barchart.addChild([scoreRangesPie, scoreRanges,
StudentsTable,multiTable]);
    Barchart.draw();
    }

    var scoreRangesPie = dash.createPieChart({
    container: "#svg2",
    groupBy: "result.score.scaled",

    pre: function(data, event){

        return data.where('actor.mbox = "' + event.in + '"');

    },
    post: function(data){
        data.contents.map(function(el){

            });
        },
        customize: function (chart){ //
        { chart.height(450).width(500);
            chart.labelType("percent");

        }

        },
        aggregate: ADL.count(),

    } );

    var StudentsTable = dash.createTable({
        groupBy: 'timestamp',
        pre: function(data, event){
            score= event.in;
            data.where('result.score.scaled != null and actor.mbox = "' +
event.in + '"').orderBy('object.id');
        },

        aggregate: ADL.multiAggregate('timestamp', ADL.select('object.id'),
ADL.select('verb.display.und'), ADL.select('result.score.scaled')),
        container: '#table2',

    });

    var scoreRanges = dash.createBarChart({
        container: "#svg3",
        groupBy: "timestamp",
        range: {start: 0, end: 1.00, increment: 0.1},
        pre: function(data, event){
            user = event.in;
            return data.where('result.score.scaled != null and actor.mbox
= "' + event.in + '"').orderBy('timestamp');
        },
        aggregate: ADL.select("result.score.scaled"),

        customize: function(chart, event){
            chart.xAxis.rotateLabels(45).axisLabel("Timestamp ");

```



```

        chart.yAxis.axisLabel("Score");
        chart.height(200);
        ADL.$('h2').textContent = 'Data for user: ' + user ;
    },
    });

```

Excerpt 3: In this snippet, an aggregate table is created, which calculates, the number of activities a user had, the minimum, maximum and average score for these activities and the verb . This code is used in Reports C and D to create the table with min, max and average score per student.

```

var multiTable = dash.createTable({
    container: '#multiTableContainer',
    groupBy: 'actor.mbox',
    pre: function(data, event){
        data.where('result.score.scaled != null and actor.mbox = "' +
event.in + '"');
    },
    aggregate: ADL.multiAggregate('result.score.scaled', ADL.min,
ADL.max, ADL.average)
});

```

6.4 WordPress H5P Monitoring Tool

This monitoring tool is provided by the H5P plugin for WordPress and displays aggregate data (Results for the registered Users). See Figure 5.21 for a screenshot of the actions monitored by this tool.

Chapter 7 - Evaluation

The Adaptive Learning System of our implementation was used by 46 students of 5th Grade of Primary School in Crete, Greece. They were given an email address (which was virtual) for their actions to be recorded. One group of 25 was asked to take first Course 1 and then Course 2, whereas the other one was asked to take only Course 2. As expected, the users that took Course 1 skipped the part of Course 2 that was related to formatting. The students who failed a part (section) had to re-watch the slides regarding this content. In the following screenshots we present some charts and reports regarding our evaluation.

A. Charts and reports regarding our evaluation from 46 students

Figure 7.1 illustrates how active every user was in our evaluation courses.

Figure 7.1 The activity for every user of our evaluation

This Monitoring Tool provides information about the LRS.

Activity per user (Number of Statements in the LRS)

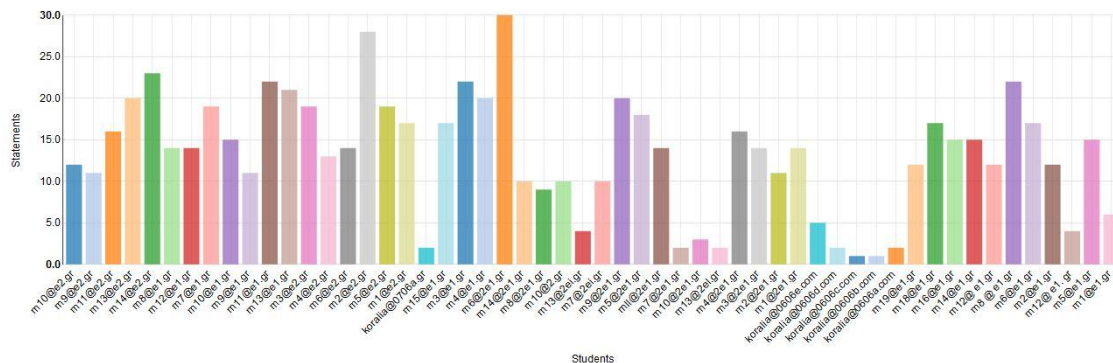
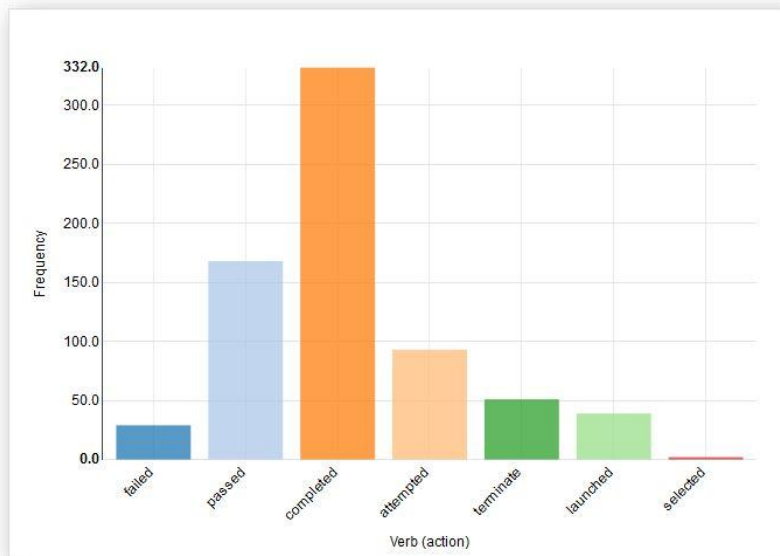


Figure 7.2 displays how frequently each verb appeared in our evaluation.

Figure 7.2 The frequency of actions for our evaluation

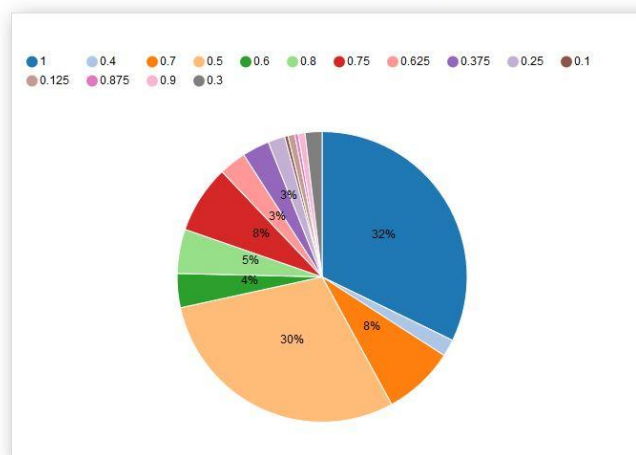
Frequency of verbs (Number of statements in the LRS)



During the evaluation, 978 different statements were generated from the courses, tracking the actions and scores of the users. Out of them 264 statements contained information for the score achieved. The percentage of each score is illustrated in Figure 7.3.

Figure 7.3 The score achieved in several statements of our evaluation

Distribution of score in the LRS (Percentage per score achieved)



B. Charts and reports regarding the users

Figure 7.4 illustrates all users of our evaluation along with their average score.

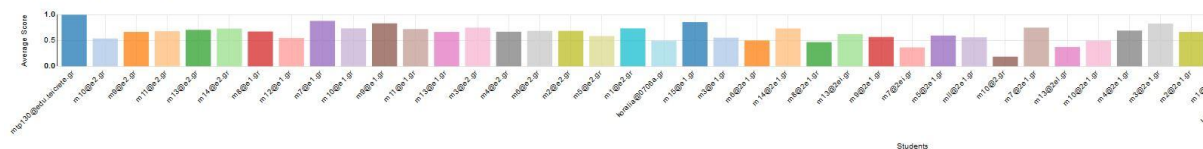
Figure 7.4 The average score for every user of our evaluation

This Monitoring Tool provides information about each user.

Average score per student for all his activities

Activities/ users without score are not illustrated in this graph

Click on the bar of a user to see his statistics

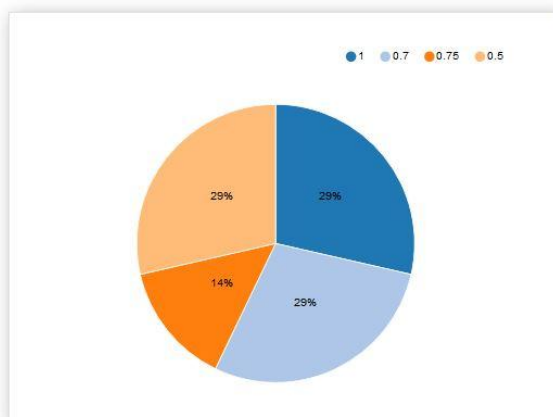


When we click on a user's barchart we may see charts regarding the specific user. An example from charts is given below. Figure 7.5 illustrates the score of the statements for a specific user in chart and as a report.

Figure 7.5 The score for each statement of the user (pie chart and table)

Data for user: mailto:m1@e2.gr

Distribution of score (piechart)



timestamp	object.id	verb.display.und	result.score.scaled
2017-06-08T06:38:31.605Z	http://koralia/Eisagwgi_sto_excel	completed	0.7
2017-06-08T06:26:28.583Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A	passed	0.5
2017-06-08T06:26:06.859Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A	passed	0.5
2017-06-08T06:33:38.060Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_B	passed	0.75
2017-06-08T06:37:24.986Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_CD	passed	0.7
2017-06-08T06:38:25.663Z	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E	passed	1
2017-06-08T06:23:35.542Z	http://koralia/Test_in_text_Formatting	completed	1

Figure 7.6 illustrates the progress of a user during the course.

Figure 7.6 The progress of a user of our evaluation

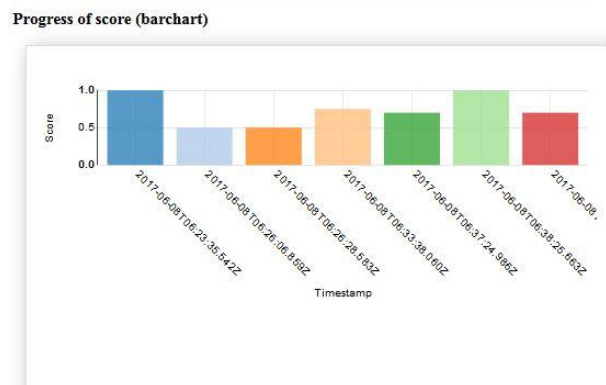


Figure 7.7 illustrates the minimum, maximum and average score of a user of our evaluation.

Figure 7.7 The minimum, maximum and average score of the user

Minimum, Maximum and Average score per user

actor.mbox	min	max	average
mailto:m1@e2.gr	0.5	1	0.7357142857142858

C. Charts and reports regarding the activities (Units) of the course taken by the students

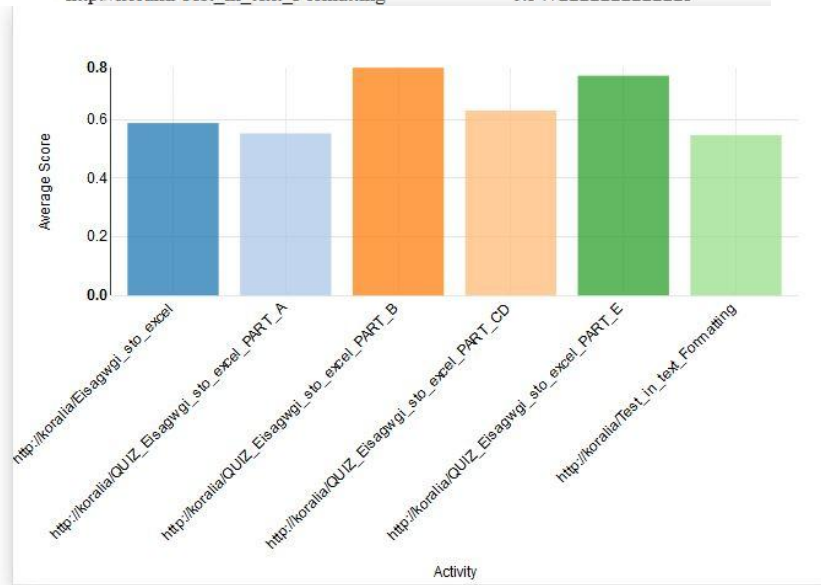
Figure 7.8 illustrates the average score of each activity (Unit) of our Evaluation Course

Figure 7.8 The average score for every activity of the course (table and barchart)

Aggregate Data for our Activities

Average Score per Activity

object.id	result.score.scaled
http://koralia/Eisagwgi_sto_excel	0.588235294117647
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A	0.553030303030303
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_B	0.7772727272727272
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_CD	0.6309523809523809
http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E	0.75
http://koralia/Test_in_text_Formatting	0.547222222222223



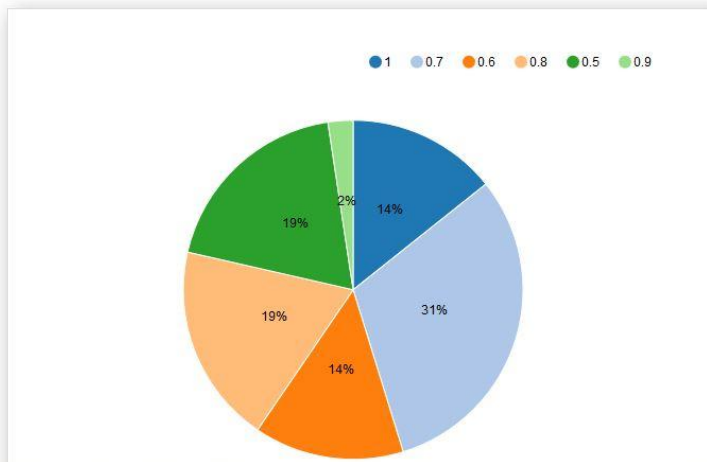
This report illustrates PART_B (from Course 1) to be the easiest (highest average score) and Test_in_text_Formatting (Course 1) to be the hardest.

Figure 7.9, Figure 7.10, Figure 7.11, Figure 7.12 and Figure 7.13 illustrate the distribution of score for the parts 'Eisagwgi_sto_excel', 'Eisagwgi_sto_excel_PART_A', 'Eisagwgi_sto_excel_PART_B', 'Eisagwgi_sto_excel_PART_CD', 'Eisagwgi_sto_excel_PART_E' respectively.

Figure 7.9 The illustration of the score achieved for the 'Eisagwgi_sto_excel' part

Distribution of score (Percentage for each score achieved)

Data for activity: http://koralia/Eisagwgi_sto_excel



Distribution of score (Frequency of score)

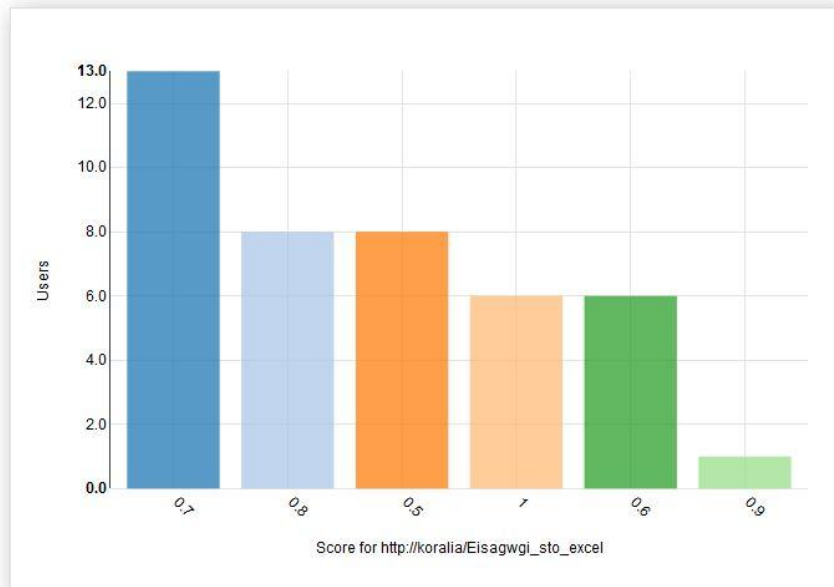
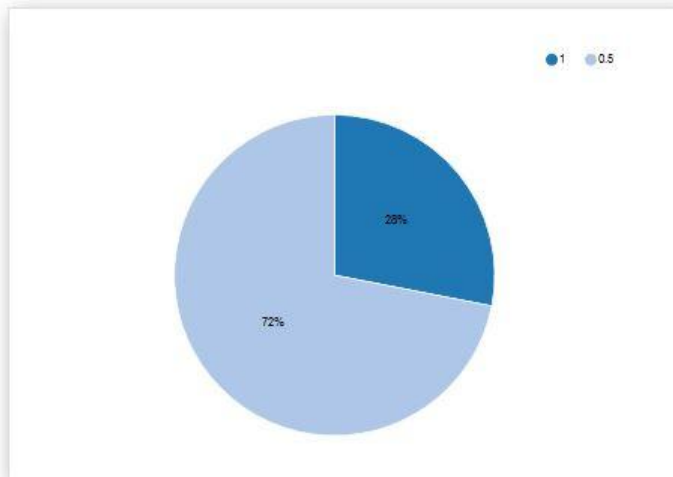


Figure 7.10 The illustration of the score achieved for the first quiz 'Eisagwgi_sto_excel_PART_A'

Distribution of score (Percentage for each score achieved)

Data for activity: http://koralia/QUIZ_Eisagwgi_sto_excel_PART_A



Distribution of score (Frequency of score)

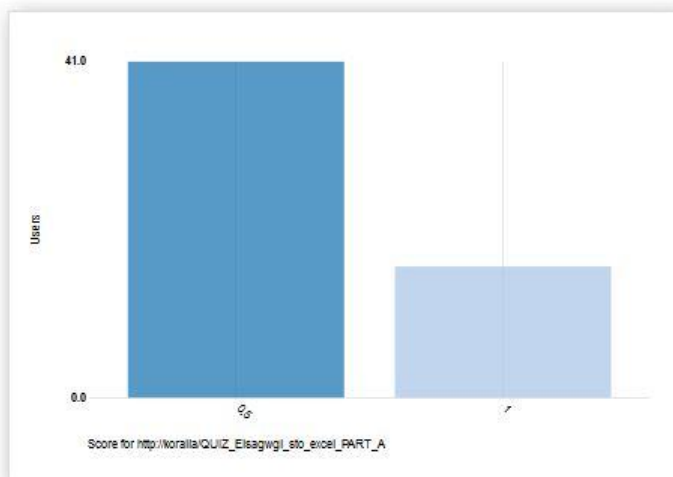
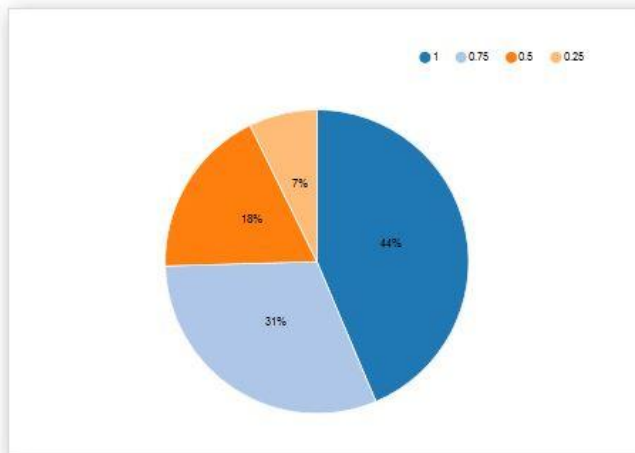


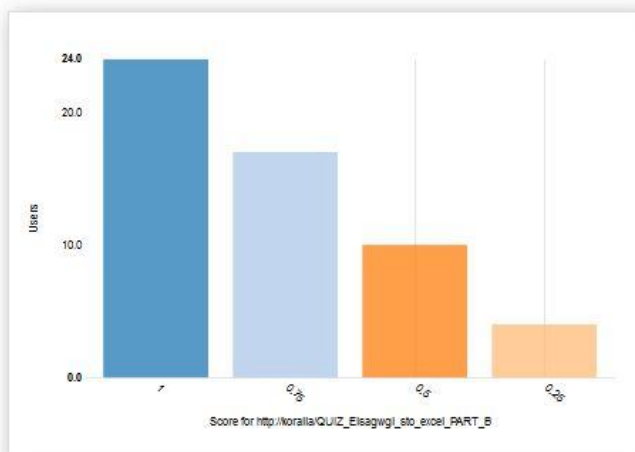
Figure 7.11 The illustration of the score achieved for the first quiz 'Eisagwgi_sto_excel_PART_B'

Distribution of score (Percentage for each score achieved)

Data for activity: http://koralia/QUIZ_Eisagwgi_sto_excel_PART_B



Distribution of score (Frequency of score)

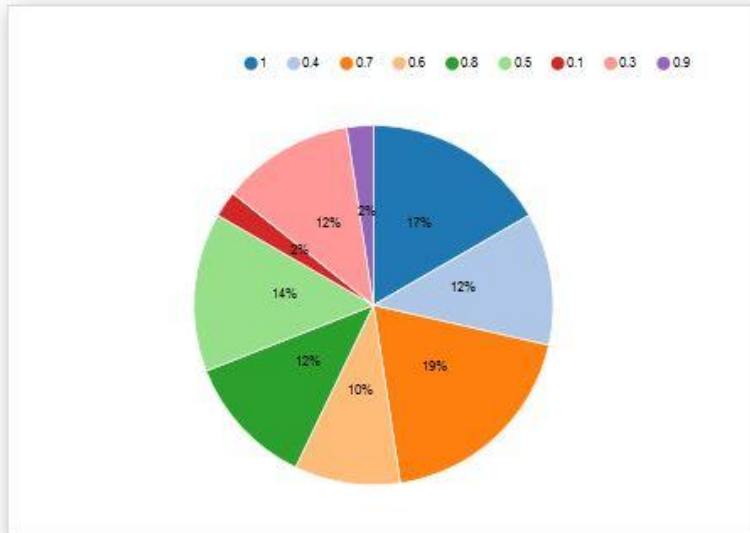


Graphs and Tables created with ADL's xAPI Dashboard

Figure 7.12 The illustration of the score achieved for the first quiz 'Eisagwgi_sto_excel_PART_CD'

Distribution of score (Percentage for each score achieved)

Data for activity: http://koralia/QUIZ_Eisagwgi_sto_excel_PART_CD



Distribution of score (Frequency of score)

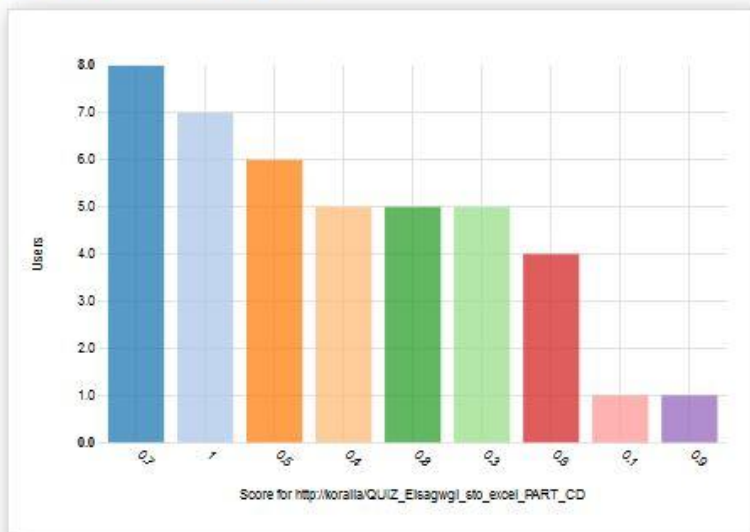
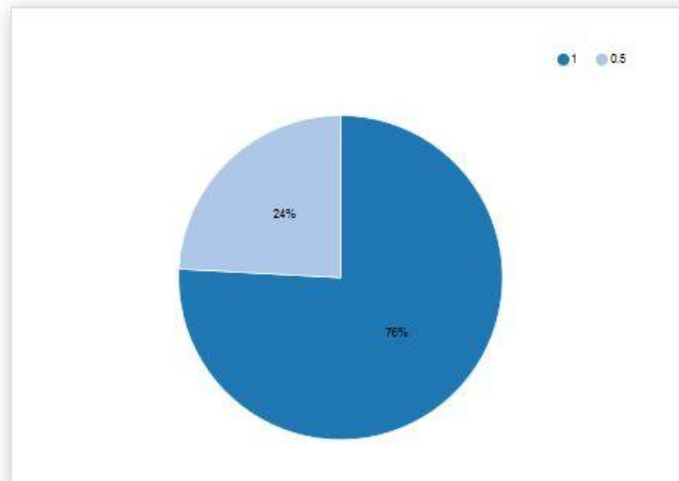


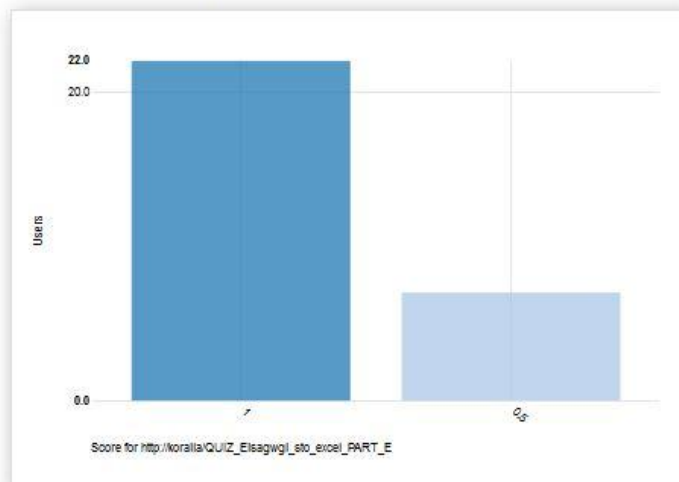
Figure 7.13 The illustration of the score achieved for the first quiz 'Eisagwgi_sto_excel_PART_E'

Distribution of score (Percentage for each score achieved)

Data for activity: http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E



Distribution of score (Frequency of score)



Out of the 46 students, only 28 passed the last QUIZ http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E.

Table 7.1 The users that passed http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E

	User	Verb	Activity
1	m1@2e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
2	m1@e2.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E

3	m10@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
4	m11@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
5	m11@e2.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
6	m12@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
7	m13@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
8	m13@e2.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
9	m14@e2.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
10	m15@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
11	m16@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
12	m18@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
13	m19@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
14	m2@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
15	m3@2e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
16	m3@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
17	m3@e2.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
18	m4@2e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
19	m5@2e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
20	m5@e2.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
21	m6@2e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
22	m7@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
23	m8 @e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
24	m11@2e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
25	m14@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
26	m5@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
27	m6@e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E
28	m9@2e1.gr	passed	http://koralia/QUIZ_Eisagwgi_sto_excel_PART_E

Out of the 46 students that took the course, only 2 clicked on the Coloring Palette (see Figure 5.15), to see the Colors of the palette (Figure 7.14).

Figure 7.14 Only two students clicked on the Coloring Palette

xAPI Statement Viewer
 Configured and modified by Koralia Papadokostaki

Statement List [Get Statements](#)

Query Options

Predefined Verb: Verb Sort:

Verb ID:

Agent: Related Agents:

Activity ID: Related Activities:

Registration ID:

Statement ID: Voided Statement ID:

Object Name (Added For Our Implementation):

Since Date: Until Date: Limit:

xAPI Query:

[Clear Form](#)

Show entries Search:

Timestamp	Actor Name	Verb	Object Name	Object Type	Authority Name	
2 months ago	m11@e1.gr		http://koralia/ColoringPalette	Activity	koralia	
2 months ago	m14@e1.gr		http://koralia/ColoringPalette	Activity	koralia	
Timestamp	Actor Name	Verb	Object Name	Object Type	Authority Name	

Showing 1 to 2 of 2 entries [Previous](#) [1](#) [Next](#)

All users were enthusiastic about the course, as they could watch it at their own pace and review their knowledge in short quizzes. They found the content user-friendly and simple and wanted to retake the courses or using e-learning in class.

Although the statements were massively sent to the LRS and the communication between the LRS and the course was two-way, our implementation responded with stability and accuracy; the young students noticed no delay or inconsistency during their engagement with the course. Despite the fact that the architecture of our implementation is distributed, i.e. the course and the LRS are hosted in diverse systems, the response time of our system is satisfactory (about 270 msecs, where 200 msecs is the roundtrip time between the user and the LRS) and does not affect the responsiveness of our implementation.

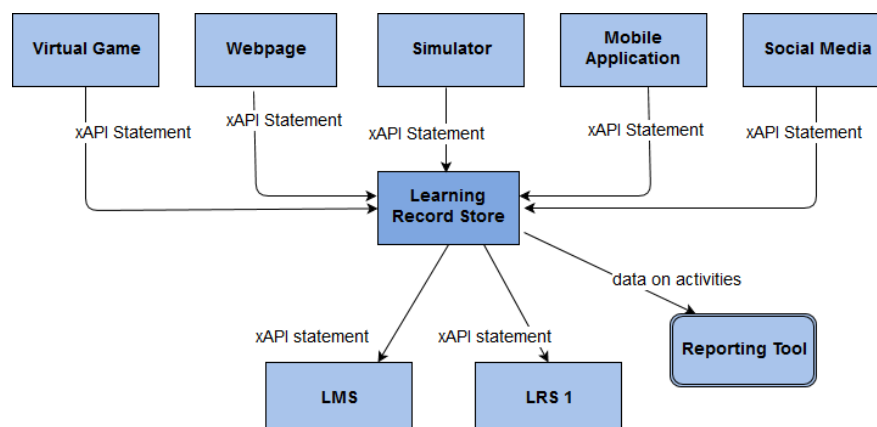
The time Charts and Reports need to load depends on the number statements of our LRS by our implementation. In our case, with an LRS of about 1800 statements, the time needed for our 'heaviest' report is about 20 secs, as it gathers and aggregates all statements from the LRS. The Statement viewer of our implementation presents the filtered results in a time interval of 780 msec, which is also satisfactory taking in consideration that it is not noticed by the user.

Chapter 8 - Conclusions

Now that its basic structure and functionality have been presented, a critical review upon xAPI will be attempted and a short comparison to its ancestor SCORM will be made.

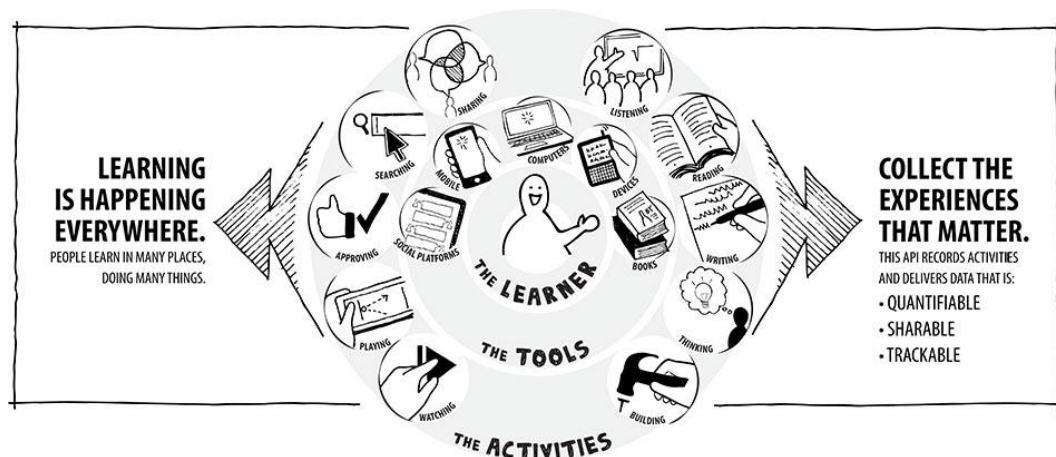
xAPI makes tracking activities from various sources a reality; statements concerning the same learner may originate from webpages, mobile applications, simulators, virtual games or social networking tools [33], [34]; all these diverse technologies can be used as training systems and data from them should end up in the same storage unit, the same LRS. Figure 8.1 illustrates the vast diversity of sources for the statements of xAPI and the flexibility in use of the data, that xAPI and its essential component, LRS, provide.

Figure 8.1 xAPI supports a distributed architecture, where statement streams can originate from diverse sources and may be delivered to several endpoints.



The xAPI extends learning environments further than SCORM and provides independence from LMSs, fulfilling this way the vision of ‘lifelong learning’, since learning can happen everywhere. It captures data in a consistent format about a person or group’s activities from many technologies as illustrated in Figure 8.2 and realizes ubiquitous [33] and lifelong [55] learning. With the use of xAPI learning is freed from the use of LMS and the learning content may be distributed across disparate systems. As [56] support ‘Learning beyond the LMS is both desirable and achievable’ through xAPI. Additionally, the students’ activities like interaction with mobile learning applications, museum exhibits, or any data that can be pushed to the cloud can be gathered to a single store, a concept similar to Personal Learning Environments (PLEs) [55].

Figure 8.2 The ubiquitous vision of xAPI (Figure taken from [35])



When compared to SCORM, xAPI is a much wider technology, which can be used in various circumstances and has many advantages compared to SCORM. Firstly, in order to use SCORM, the learning contents should be delivered in SCORM packages, which can be a serious limitation for the content developer. In xAPI, though, learning activities or contents can be totally independent of data formats [57], as simple web content can be a learning activity and libraries or applications implementing the xAPI specification can provide the infrastructure for the delivery of the statements to an LRS. Additionally, as xAPI is based on the delivery of statements relative to the content and not the content itself, they are easier to implement and give the content-developers flexibility concerning the content and the hosting of the content. For example, the content may be offline and xAPI may deliver the statements to the LRS through an occasional connection to the Internet [33], [36]. This is a major advantage for xAPI, as the learner need not be constantly online, but may still contribute to the LRS with the activities that he accomplished in form of statements. Moreover, xAPI may prove to be a priceless mechanism in the hands of data analysts, as it can cooperate well with Business Analytics and reporting tools, contrary to LMSs, the traditional hosts of SCORM content [34], [57]. Finally, xAPI may simultaneously integrate with one or several LRSs, and optionally with an LMS offering this way extra value to the administrator of the data. Table 8.1 illustrates these differences synoptically.

Table 8.1 Differences between SCORM and xAPI

	SCORM	xAPI
learning contents	should be delivered in SCORM packages	learning activities or contents can be totally independent of data formats
Availability of content	Need be online	may be offline
LMS- dependency	LMS- dependent	LMS- independent
Cooperation with BI tools	Does not cooperate well with BI	Cooperates with LRSs, LMSs, BI and reporting tools

From another perspective, xAPI can promote collaborative learning through the use of collaborative applications, social media or even serious or virtual games. Using it may augment teamwork and may convert e-learning from personal learning to team-learning [33], [56] . Therefore it may be used to support the social constructivist learning theory, where social interactions and relations are the key to learning [55],[56]. Along with the activities theories that they implement they seem to be the future of constructivist-aligned strategies [55], [56] .

With Internet and mobile devices, YouTube videos, serious games, simulations or posts on social media can provide valuable knowledge to the learner; with the use of xAPI and its implementations, all these learning activities can be captured and may contribute to the definition of each learner's personal profile. Till now, only knowledge that was delivered through formal e-education could be recorded, now tracking informal learning may give us additional information upon the learner, the content and the learning process. Keeping a record of an individual's learning experiences can play an important role in providing him with the proper content in the most efficient manner, which is the goal of Adaptive Learning/ training [8]. Building an adaptive learning system may alter the content to meet the learner's needs or might change the way the content is presented according to the learner's profile [5], [8], [42].

From the viewpoint of Learning Analytics, where educational data is collected and analyzed aiming in the discovery of patterns in learning process or problems in student performance, xAPI is indeed a very promising technology [23], [55]. In the five stages of collecting, reporting, predicting, acting and refining [23], xAPI can pioneer in collecting data from various sources (not necessarily LMSs) and provide aggregate or summarized data to third-party tools for reporting and predicting [23]. Kitto [56] stresses out, that ‘learning beyond the LMS does not rule out the possibility of data capture and Learning Analytics’ and presents the Connected Learning Analytics Toolkit (CLAT), an open source Learning Analytics toolkit for xAPI. In this dissertation we have explored the world of Activity Streams, and more specifically xAPI. xAPI is a specification that records learning activities from various sources and shares them with other applications. It can help us form the learning profile of a user and is therefore a valuable tool for Learning Analytics. Due to its openness and versatility it might be used in several occasions and for various uses, collecting thus activities from courses, videos, e-books, mobile applications, serious games, simulators etc. As a result, the data collected are huge and may contribute to the Big Data revolution that has already started. Additionally it is also widely used in Business Analytics, training and Defense.

xAPI can also be an integral part of the Internet of the Things (IoT), where sensors, devices and wearables will be interconnected. If they are also xAPI enabled, they might provide precious information upon the user, immediately and seamlessly, without his interference. An xAPI-enabled smart television, for example, might communicate with an LRS to transmit the link of a YouTube video the user watched, while an xAPI enabled GPS-sensor, might automatically send positioning information about the user, that might be related to a conference or seminar. It is also possible that multiple LRSs might cooperate in order to exchange information about a user and offer him more personalized options, e.g. suggested videos, based upon previous activities he may have had.

Generating xAPI statements massively, on the other hand, may contribute to the Big Data revolution that gathers astonishing volume of data from diverse sources with the penetration of the Internet. It complies with the 4-Vs characteristics of Big Data [59] which stand for Volume, Variety, Velocity and Value, as each action of a user may be represented with a statement (Volume), statements may come from a vast spectrum of actions, e.g. learning, training or everyday actions (Variety), statements are transmitted online and with

great speed (Velocity) and provide significant information upon a user's actions (Value). We believe xAPI has a great potential if integrated with IoT and may significantly expand the value of Big Data.

In this thesis, we have implemented various applications of xAPI, manifesting its vast range of uses and applications and its wide possibilities and promising future. We have developed a Dashboard for the teacher to provide him with monitoring tools, but more significantly, we have implemented an innovative LMS-free Adaptive Learning System based on xAPI that does not only produce learning activities, but also consumes them, adapting the content according to the recorded activities of the user. In the future, we plan to extend our application to incorporate activities from various sources, in order to adapt our content analogously, exploiting various Data mining algorithms for content adaptation. We also intend to adopt Open Badges, while incorporating data from other LRSs and integrating with LMSs to make our implementation even more versatile is within our future plans. As adoption of xAPI is extended in learning Industry and other fields, we believe our work will be one more step towards the applications and uses of xAPI.

References

- [1] "ADL." [Online]. Available: <http://www.adlnet.gov/>. [Accessed: 04-Aug-2017].
- [2] "Experience API, Appendix A: Revision History." [Online]. Available: <https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-About.md#Appendix1A>. [Accessed: 21-May-2017].
- [3] "SCORM Overview." [Online]. Available: <http://www.adlnet.gov/SCORM>. [Accessed: 04-Aug-2017].
- [4] K. C. Lim, "Case Studies of xAPI Applications to E-Learning," in *Case Studies of xAPI Applications to E-Learning*, 2015.
- [5] V. Moisa, "Adaptive Learning Management System," *J. Mobile, Embed. Distrib. Syst.*, vol. 5, no. 2, pp. 70–77, 2013.
- [6] C. Mellon, "Best Practices Guide for Content Developers 1," 2003.
- [7] V. J. Shute and B. Towle, "Adaptive E-Learning," *Educ. Psychol.*, vol. 38, no. 2, pp. 105–114, 2003.
- [8] A. Paramythis and S. Loidl-reisinger, "Adaptive Learning Environments and e-Learning Standards," in *Second european conference on e-learning, volume 1, number 2003*, 2003, pp. 369–379.
- [9] C. Chrysoulas, "Building an Adaptive E-learning System Building an Adaptive E-learning System," in *9th International Conference on Computer Supported Education*, 2017, no. April.
- [10] D. Zapata-rivera and V. J. Shute, "Adaptive educational systems," *Adapt. Technol. Train. Educ.*, vol. 7, pp. 4–27, 2012.
- [11] M. Monova-zheleva, "Adaptive Learning in Web-based Educational Environments," *Cybern. Inf. Technol.*, vol. 5, no. 1, pp. 44–55, 2005.
- [12] M. Sharples, J. Taylor, and G. Vavoula, "A theory of learning for the mobile age," *SAGE Handb. E-learning Res.*, pp. 221–247, 2007.
- [13] M. Prensky, "Digital Immigrants, Digital Natives part 1," *Horiz.*, vol. 9, no. 5, pp. 1–6, 2001.
- [14] N. Balacheff, T. De Jong, A. Lazonder, M. Sharples, M. Milrad, and G. Vavoula, "MOBILE LEARNING Small devices, big issues," in *Technology-Enhanced learning: Principles and products*, Springer-Verlag, Berlin, 2008, pp. 1–20.

- [15] M. Sharples, "Mobile learning : research , practice and challenges," *Distance Educ. China*, vol. 3, no. 5, pp. 5–11, 2013.
- [16] K. Ciampa, "Learning in a mobile age: An investigation of student motivation," *J. Comput. Assist. Learn.*, vol. 30, no. 1, pp. 82–96, 2014.
- [17] N. Zaranis, M. Kalogiannakis, and S. Papadakis, "Using Mobile Devices for Teaching Realistic Mathematics in Kindergarten Education," *Creat. Educ.*, vol. 4, no. 7, pp. 1–10, 2013.
- [18] D. Hung, "Theories of Learning and Computer-Mediated Instructional Technologies," *EMI. Educ. Media Int.*, vol. 38, no. 4, pp. 281–287, 2001.
- [19] Y. Park, "A Pedagogical Framework for Mobile Learning : Categorizing Educational Applications of Mobile Technologies into Four Types," *Int. Rev. Res. Open Distrib. Learn.*, vol. 12, no. 2, pp. 78–102, 2011.
- [20] S. Yahya, E. A. Ahmad, and K. A. Jalil, "The definition and characteristics of ubiquitous learning : A discussion," *Int. J. Educ. Dev. Using Inf. Commun. Technol.*, vol. 6, no. 1, pp. 117–127, 2010.
- [21] B. Cope and M. Kalantzis, "Ubiquitous Learning : An Agenda for Educational Transformation," in *6th International Conference on Networked Learning*, 2008, pp. 576–582.
- [22] Tanya Elias, "Learning Analytics," 2011.
- [23] A. Del Blanco, A. Serrano, M. Freire, I. Martinez-Ortiz, and B. Fernandez-Manjon, "E-Learning standards and learning analytics. Can data collection be improved by using standard data models?," *IEEE Glob. Eng. Educ. Conf. EDUCON*, pp. 1255–1261, 2013.
- [24] G. Siemens and P. Long, "Penetrating the Fog: Analytics in Learning and Education," *Educ. Rev.*, vol. 46, no. 5, p. 30, 2011.
- [25] A. Corbi and D. Burgos, "Review of current student-monitoring techniques used in elearning-focused recommender systems and learning analytics . The Experience API & LIME model case study," *IJIMAI*, vol. 2, no. 7, pp. 44–52, 2014.
- [26] "Caliper Analytics® Background." [Online]. Available: <http://www.imsglobal.org/activity/caliper>. [Accessed: 16-Aug-2017].
- [27] "1484.11.1-2004 - IEEE Standard for Learning Technology." [Online]. Available: <https://standards.ieee.org/findstds/standard/1484.11.1-2004.html>. [Accessed: 16-Aug-2017].

- [28] D. Griffiths and T. Hoel, "Comparing xAPI and Caliper," *Learn. Anal. Rev.*, no. 7, pp. 1–27, 2016.
- [29] V. Lukarov, M. A. Chatti, H. Thüs, F. S. Kia, A. Muslim, C. Greven, and U. Schroeder, "Data Models in Learning Analytics," in *DeLFI Workshops*, 1014, pp. 88–95.
- [30] "Initial xAPI/Caliper Comparison." [Online]. Available: <https://www.imsglobal.org/initial-xapicaliper-comparison>. [Accessed: 04-Aug-2017].
- [31] A. Berg, M. Scheffel, H. Drachsler, and M. Specht, "Dutch Cooking with xAPI Recipes The Good , the Bad , and the Consistent," in *IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*, 2016, pp. 234–236.
- [32] I. Ruano, P. Cano, J. Gámez, and J. Gómez, "Advanced LMS Integration of SCORM Web Laboratories," *IEEE Access*, vol. 4, pp. 6352–6363, 2016.
- [33] K. Murray, P. Berking, J. Haag, and N. Hruska, "Mobile Learning and ADL ' s Experience API Overview of the xAPI," *Connect. Q. J.*, vol. 12, no. 1, pp. 45–49, 2012.
- [34] "Experience API/ Tin Can API." [Online]. Available: <http://experienceapi.com>. [Accessed: 04-Aug-2017].
- [35] "The xAPI Overview." [Online]. Available: <https://www.adlnet.gov/xAPI>. [Accessed: 04-Aug-2017].
- [36] "xAPI-Spec." [Online]. Available: <https://github.com/adlnet/xAPI-Spec/>. [Accessed: 21-May-2017].
- [37] C. Glahn, "Using the ADL experience API for mobile learning, sensing, informing, encouraging, orchestrating," in *7th International Conference on Next Generation Mobile Applications, Services, and Technologies*, 2013, pp. 268–273.
- [38] "Anatomy of a Experience API Statement." [Online]. Available: <https://tincanapi.com/ebook/>. [Accessed: 04-Aug-2017].
- [39] "Tin Can API Registry." [Online]. Available: <https://registry.tincanapi.com/#home>. [Accessed: 04-Aug-2017].
- [40] "Experience API Ecosystem." [Online]. Available: <https://experienceapi.com/ecosystem/>. [Accessed: 04-Aug-2017].
- [41] "Experience API Adopters." [Online]. Available: <https://experienceapi.com/adopters/>. [Accessed: 29-Aug-2017].
- [42] T. Poeppelman, M. Hruska, R. Long, and C. Amburn, "Interoperable Performance Assessment for Individuals and Teams Using Experience API," in *Second Annual GIFT*

- Users Symposium, Orlando, FL, 2015, pp. 1–8.*
- [43] M. Hruska, C. Amburn, R. Long, and T. R. Poeppelman, “Experience API and Team Evaluation : Evolving Interoperable Performance Assessment,” in *The Interservice/Industry Training, Simulation \& Education Conference (I/ITSEC)*, 2014, pp. 1–11.
 - [44] “ADL LRS.” [Online]. Available: <https://www.adlnet.gov/adl-lrs>. [Accessed: 12-Jul-2017].
 - [45] “TinCan Javascript Library.” [Online]. Available: <http://rusticsoftware.github.io/TinCanJS/>. [Accessed: 04-Aug-2017].
 - [46] “adlnet/xAPIWrapper: Wrapper to simplify communication to an LRS.” [Online]. Available: <https://github.com/adlnet/xAPIWrapper>. [Accessed: 04-Aug-2017].
 - [47] “xAPI Dashboard.” [Online]. Available: https://github.com/adlnet/xAPI-Dashboard/blob/master/API_dashboard.md. [Accessed: 04-Aug-2017].
 - [48] “adlnet/xapi-youtube.” [Online]. Available: <https://github.com/adlnet/xapi-youtube>. [Accessed: 04-Aug-2017].
 - [49] D. Burgos, “L . I . M . E . A recommendation model for informal and formal learning , engaged,” vol. 2, pp. 79–86, 2013.
 - [50] “BootStrap.” [Online]. Available: <http://getbootstrap.com/>. [Accessed: 04-Aug-2017].
 - [51] “H5P.” [Online]. Available: <https://h5p.org>. [Accessed: 04-Aug-2017].
 - [52] “xapi-statement-viewer.” [Online]. Available: <https://github.com/adlnet/xapi-statement-viewer>. [Accessed: 04-Aug-2017].
 - [53] “H5P and xAPI.” [Online]. Available: <https://h5p.org/documentation/x-api>. [Accessed: 04-Aug-2017].
 - [54] “wp-h5p-xapi.” [Online]. Available: <https://wordpress.org/plugins/wp-h5p-xapi/>. [Accessed: 04-Aug-2017].
 - [55] J. Kevan and P. R. Ryan, “Experience API : Flexible , Decentralized and Activity-Centric Data Collection,” *Technol. Knowl. Learn.*, vol. 21, no. 1, pp. 143–149, 2016.
 - [56] K. Kitto, S. Cross, Z. Waters, M. Lupton, and G. Street, “Learning Analytics beyond the LMS : the Connected Learning Analytics Toolkit,” in *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge2*, 2015, pp. 11–15.
 - [57] A. Benedek, “Learning Design Versus Learning Experience Design: Is the Experience Api Making the Difference?,” in *Edulearn13: 5th International Conference on*

Education and New Learning Technologies, 2013, pp. 2609–2621.

- [58] “Nine Practical Uses of the Tin Can API (xAPI).” [Online]. Available: <https://www.youtube.com/watch?v=8LFhDdqQ13A>. [Accessed: 04-Aug-2017].
- [59] K. Papadokostaki, G. Mastorakis, S. Panagiotakis, C. X. Mavromoustakis, C. Dobre, and J. M. Batalla, “Handling Big Data in the Era of Internet of Things (IoT),” in *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, C. X. Mavromoustakis, G. Mastorakis, and C. Dobre, Eds. Cham: Springer International Publishing, 2017, pp. 3–22.

List of Abbreviations

ADL	Advanced Distributed Learning
BAA	Broad Agency Announcement
DoD	Department of Defense
HPML	Human Performance Measurement Language
IRI	Internationalized Resource Identifier
JSON	JavaScript Object Notation
LETSI	Learning-Education-Training Systems Interoperability
LMS	Learning Management System
LRS	Learning Record Store
MOOCs	Massive Open Online Courses
PLE	Personal Learning Environment
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
URI	Uniform Resource Identifier
UUID	Universally Unique Identifier
xAPI	Experience API

References of xAPI adopters

The homepages of the xAPI adopters presented in this Thesis are listed below, in the order presented:

Software	Webpage
Blackboard	http://www.blackboard.com/
DoceBo	http://www.docebo.com/
Litmos	http://www.litmos.com/
Moodle	https://moodle.org/
Sakai	http://www.sakaiproject.org/
TalentLMS	http://www.talentlms.com/
TOPYX	https://topyx.com/
Captivate® (7+)	http://www.adobe.com/products/captivate.html
Storyline & Presenter	http://www.articulate.com/products/storyline-overview.php
Exam Builder	http://www.exambuilder.com/
dominKnow Platform	http://www.dominknow.com/
H5P	http://h5p.org/
AEL Data	http://www.aeldata.com/
Curatr	http://curatr.co.uk/
eLearning Studios	http://www.e-learningstudios.com/
Flipick	http://www.flipick.com/
HiHaHo	https://www.hihaho.com/tincan
PulseWeb	http://www.pat.nl/english/pulseweb-e-learning
Rapport	http://www.pal-tech.com/
SquirrelThat (Kalleo People Group)	http://www.squirrelthat.com/
Tappestry	http://tappestryapp.com/
JobAider	http://www.peregrinepg.com/
Knowledge Guru™	http://www.theknowledgeguru.com/

SurveyGizmo	http://www.surveygizmo.com/
TREK	http://www.cognitiveadvisors.com/
WebRatio	http://www.webratio.com/