The Past, Present and Future of SCORM

Petros PAPAZOGLOU PAPAZOGLAKIS
The Bucharest University of Economic Studies
ppapazoglakis@gmail.com

SCORM is the most widely used standard in the e-Learning world. Passed from various versions until to establish at the 2004 4th edition. Since its early days and until now, SCORM is constantly evolving. At the time of writing, the ADL organization is focusing its efforts in creating the Training & Learning Architecture (TLA). The first station of the TLA architecture is the evolution of a standard capable of tracking the learner's experiences as its name declares "the experience API". In this paper we shall try, apart from noting the crucial points of the SCORM evolution, to investigate the factors that led to the evolution of this ongoing standardization process today, along with mentioning its general functioning principles and its future perspectives.
***Keywords**: e-Learning, SCORM, TLA, Activity Streams, Experience API*

1 Introduction

Since 1999 when the Department of Defense (DoD) of United States initiated the e-Learning standardization and modernization research process and until today SCORM passed through various stages, used standards and technologies already implemented by ADLs precursors (AICC, IMS Global Learning Consortium, IEEE, Ariadne), reached various different versions with specific characteristics and possibilities and continued to evolve. At the time of writing, SCORM is evolving to the so called next generation of SCORM named as the Training & Learning Architecture (TLA). TLA is a work in progress and will define the standardization aspects that shall indicate the path that the future e-learning specifications will follow.

Being (the SCORM) for more than 12 years on the scene and being adapted to carry out the given demands of every time period. Being embraced and followed by the largest e-Learning players of the globe and being durable through time, SCORM can be correctly named as the leading standard in the world of e-Learning.

Now ADL is developing the first component of the TLA architecture, the experience API. This API started to form as an answer to the requests of the e-learning community. Requests that the SCORM 2004, due to its design and functionality could not anymore fulfill. Issues like the wide acceptance of new types of computing devices (Smart Phones, androids, tablets, etc.) from the market. The enrichment of the education process (learners profiling, e-learning assistants-tutors, cognitive adapted activities, educational games, virtual worlds, etc.). The group teaching possibilities and cooperative (or competitive) learning alternatives. The new realizations, technologies and challenges. All of them call for more power, freedom and simplicity to achieve their new era intended goals.

2 SCORM (Sharable Content Object Reference Model)

Is a group of standards, technical specifications and guidelines used in e-Learning. These standards formed nothing else than a technical framework of rules for computer based and Web based learning. This technical framework is a guide, to be followed by the developers creating e-Learning content, in order to create high quality, performance e-Learning content, that can be specially adjusted to the individual needs and also to be reusable in the form of Instructional Objects. By SCORM standardization became possible, the e-Learning content, to be uploaded and used in various LMSes (Learning Management Systems). The single condition for this to happen was both the content and the LMSes to conform to the SCORM standard and specific version of SCORM.

As a reference Model SCORM indicates the required services in order to solve a specific problem and how the needed standards and guidelines can be bond together and used. SCORM aim is the fulfillment of the following properties in every SCORM-based e-Learning environment:

- accessibility: the possibility to track and reach components from various different places and transport them to other places,
- durability: the possibility of one component to resist in various technological changes and to be able to be used again without the need of costly changes (application of new design or configuration),
- interoperability: the possibility to use components from one system into one other compatible one and
- reusability: the possibility to use and include various components in various solutions and ways.

In simple English, the SCORM standard provides to developers a clearly defined technical way, of how to write e-learning content so that this content can be used in various LMSes. Like that, the content becomes easily reusable, portable and accessible.

This reusability of the content offered (when the content complied with the SCORM standard) was the crucial factor that made this standard widely used and popular.

As its name clearly indicates SCORM is nothing else than a Reference Model for the creation of Sharable Content Objects. These objects are the so called SCOs (Sharable Content Objects). The SCOs are the tiniest content formations that can be sent from the LMS to the learner's browser. What differentiates a SCO from the assets (HTML pages, images, videos, audio, flash objects, etc) is that a SCO can communicate and exchange information with one LMS by using the SCORM Application Programming Interface (API). The SCORM API is a full set of library functions that are necessary for the content to communicate with the LMS and are written in standard JavaScript. Taking under consideration the previously mentioned facts the minimum API calls that every SCO must call are the doInitialize() and the doTerminate() for the initialization and termination of the communication with the LMS respectively[1].

The SCORM standard is describing three foundation elements as defined in the manuals published by ADL in 2004 version [2] [3] [4]:

- **The Content Aggregation Model (CAM)** [2]: Is setting the guidelines to be followed in order to package the SCOs of the learning content, in a specific way (inside a .zip file). Those packages can be uploaded and used in various LMSes. One package can contain from one SCO until hundreds of them, depending on the individual way of use and design of the content. The content package includes:
  1. A (XML described) manifest file (with name imsmanifest.xml) that defines the following:
     - the names of the assets and the SCOs that are incorporated in the package described as resources
     - how the content is organized in a (tree formation) content structure diagram called organization
     - rules for navigation and sequencing
     - metadata (in other words additional informative data) for the package itself but also for the SCOs and the aggregations. Aggregations are conglomerations of related activities.
  2. The various veritable assets and SCOs which are the real files, actually placed inside the .zip package file.
- **The Run-Time Environment (RTE)** [3]: Sets the necessary requirements to be followed by the developers of LMSes regarding the Run-Time Environment. With the term Run-Time Environment is described:
  1. the content-LMS communication in standardized manner. This is achieved by the use of the SCORM API as well as the definition of information and parameters that can be set or get from/to LMS by a SCO,

2. the steps needed to be followed, in order the content to run,
3. the definition of the data model that is used when information relevant with the learners interference with the content are transferred,
4. the way for the various SCOs inside the content to set and get values by using the SCORM API and the defined data model. Examples of information the SCO needs to get-receive from the LMS can be the name of the learner, the language used, the bookmarking point reached by a learner, etc. Examples of info set-stored in one LMS by one SCO can be e.g. the score, the time consumed, pass/ fail characterization.

- **The Sequencing and Navigation (SN)** [4]: Describes the possible ways that SCORM conformant content can be navigated and the SCOs that are included inside the content can be sequenced by the learner. This can happen following specific learner or system driven navigation and sequencing evens. This sequence and navigations definitions are usually set during the design time to form the order of execution of a group of pre-formed activities. The sequencing actually defines the order that every SCO inside an aggregation of SCOs shall be executed.

The SCORM 2004 introduced except from the learner driven SCO presentation by user choice, the flow navigation control mode where the order of the content is defined in advance by the designer but also the Choice Exit Control Mode where the user can choose a SCO of his preference from a constantly changing possibility of SCOs. For example starting with SCO choices initially hidden that is gradually presented during the learning process.

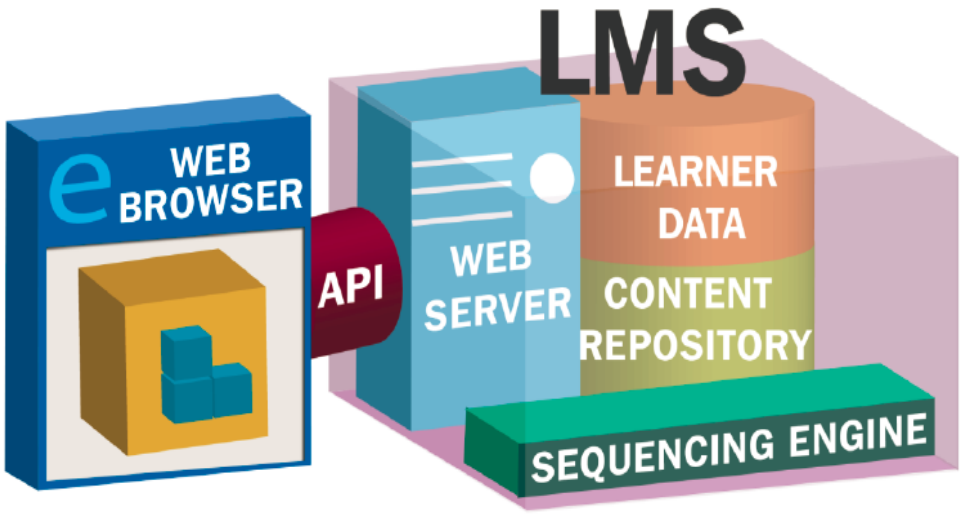

Fig. 1. The components of SCORM [1]

In other words the structures defining SCORM are:

- the Navigation and sequencing part,
- the organization and general construction of the content forming a course and
- the LMS communication and management.

A schematic representation of the various constituent parts incorporated in the SCORM standard are presented in the Figure 1.

3 History

SCORM life was initiated at January 1999 through the Executive Order 13111 given by the president of the United States, which was instructing the DoD of USA to create e-Learning specifications and standards for e-learning, in order them to be used both in the government institutions and private enterprises [16].

This order was mainly initiated by the need to standardize the e-learning world, thus pro-

vide ways for conformity, reusability, durability, portability, interoperability and accessibility (of the content) to exist. For this effort to be realized the DoD of United States launched the Advanced Distributed Learning Initiative (ADL).

The SCORM specification is a product of the ADL initiative. Since 1999 various versions of SCORM have been developed with the most resent one to be the SCORM 2004-4th Edition that was released in 2009.

The main problem that needed to be solved through standardization was that the quality content has always been very expensive, time demanding and a process difficult to be realized. In that sense, were needed ways to allow content to be reusable thus to allow an easier construction of a new e-learning activity by using mainly existing e-Learning components. Like creating Lego® e-Learning content object pieces which can be used latter as a part of other learning activities.

The main SCORM landmarks are [6]:

✓ **SCORM 1.1 (January 2001)**

The initial trial outcome of the SCORM. Was mainly focused in realizing various unresolved matters. The trial implementations showed that SCORM 1.1 was not functional, and the basic aim of interoperability was still not achieved. The result of the SCORM 1.1 implementation was mainly used as income to the following versions of SCORM.

✓ **SCORM 1.2 (October 2001)**

It is actually the first valuable release of the SCORM. This can be said because SCORM version 1.2 was the first version that allowed conformance tools to exist, thus allowing the conformance of the content to be checked. The SCORM 1.2 realization indicated that content can be created in ways that can be portable and interoperable.

In this sense, one leading LMS vendor, of that time, could start to see that the required cost and time of new content enrichment or creation, dropped almost to zero by using already implemented SCORM conformant SCOs. Any remaining issue, at that period, was a result of not completely conforming to SCORM, or the result of a wrong interpretation of a SCORM feature that still was not completely resolved.

✓ **SCORM 2004 (October 2001 until March 2009)**

The SCORM 2004 version was by large a better SCORM version comparing with SCORM 1.2 version, because it managed to reduce the ambiguities of the previous SCORM specification and by that way, managed to make SCORM to comply with stable IEEE standards. Also the SCORM 2004 API, incorporated support for a big range of languages to be used in ECMAScript (more widely known by the name JavaScript).

Additional improvements are realized also in the field of content navigation and sequencing. These improvements made SCORM 2004 an important point in the evolution of the e-Learning standards. They allowed dynamically adaptive sequencing techniques to become reality, thus allowing a more learner centred approach realization. The SCORM 1.2 allowed the content to be portable among various LMSs but (in the sequencing part) was letting the user to choose which part of the content to run and at what order. The SCORM 2004 is moving one step ahead and defines dynamic or premeditated ways of a learner to follow adjusting to its skills or potentials or to a content creator, activity-centred order.

Furthermore SCORM 2004 4th edition made possible the data exchange and sharing between the SCOs (something that until that moment was not possible) thus allowing ways for them to communicate.

At 2009 ADL published three specification manuals for the description of the SCORM specification and more precisely for the description of the three SCORM foundation components:

1 SCORM 2004 4th Edition - Content Aggregation Model (CAM) [2]
2 SCORM 2004 4th Edition - Run-Time Environment (RTE) [3]
3 SCORM 2004 4th Edition -Sequencing and Navigation (SN) [4]

In the 2004 edition are introduced the terms certification and compliance of the content with the SCORM standard (not only con-

formance that was achieved in SCORM 1.2 version) and specialized tools were created to check the compliance-adoption and finally provide the SCORM certification (from one independent 3rd party) to the content or systems, reassuring that the given products fully comply and follow the ADL specifications. These tools are published and provided (free of charge) by the ADL (e.g. the Conformance Test Suit) for the detailed evaluation of content packages, LMSes and SCOs conformance, and can be used by anybody who wishes to test the compliance of one product created (LMS, SCO, content package), with a given SCORM version.

4 Why SCORM Needs to Evolve

SCORM can be characterized as a successful standard [10] that managed to fulfill its purpose with relatively great success. That is to provide a common, free, e-learning standard for massive use, as a common reference for the e-Learning content. This is the main reason that made SCORM to remain on the scene and evolve all these years.

The problem is that since 2009 a lot of things changed or evolved. Various new technologies became popular. New consumer products and realizations came out. New forms of e-learning were introduced.

The massive market acceptance of new types of computing devices was one of the problems that were not taken under consideration when SCORM 2004 4th edition was released. At that time (when SCORM final version was introduced) SCORM 2004 was focusing on standardization issues used by (the old fashioned now) desktops and laptops and was not foreseen the imminent invasion of smaller devices (like the tablets and mobile phone versions that exist today). These devices have different characteristics and possibilities. Smaller screens, touch screens, various native OS (android, iOS, windows mobile, etc.) wireless fast internet connection possibilities (4G, Wi-Fi) and various extra gadgets (accelerometers, video cameras, GPS receivers etc.).

Furthermore SCORM was implemented under the concept that the content shall always be used, in close cooperation with an LMS (Learning Management System) and a browser. Not independently how for example native mobile applications demand. In addition to that there are specific types of content that cannot be specified in advance inside a strictly formatted .zip package, as SCORM 2004 defines (E.g. native mobile-pc applications, Rich Internet applications, heavy out of browser simulations) and the idea of packaging content (as e-learning community indicates) is becoming more and more old fashion[13]. In addition to all the previous facts, the cost of SCORMifying content is a costly process that demands effort and time. This is an oversized burden for the medium to small organizations.

Also, SCORM 2004 was not designed for team or group learning. It was aiming on a single learner tracking without the possibility for information exchange among a team or a group of users. Actually SCORM was not allowing not even tracking from SCO to SCO, until the implementation of the 4th edition of the SCORM 2004. New education concepts like the team based, social, competitive and collaborative learning, move one step ahead and demand the share of info, from a SCO to a group of users. Model not possible to be implemented with the use of SCORM 2004. Moreover, when we want to achieve a personalized way of learning, the tutor must be able to define (configure) a number of profiling customizations (level of difficulty per person, help scaffolding system enabled or not, etc). Also, general configurations must be applied (for example, number of attempts allowed for a given content, visibility or not of results, Access control, spread or not of informative events from every user to the rest of the participating users of the group, etc).

One other problem was that even though, SCORM 2004 could track enough but limited results and experiences, could not track a big differentiation and variety of experiences. Also, these experiences, could not be easily exposed or shared across various stakeholders in and out of the LMS system in order these experiences (in the form of data) to be informative or further processed (provide fur-

ther info representations, conclusions, statistics, graphs and results).
The SCORM security and authentication is one more problematic point. SCORM relies exclusively to the security offered by the LMS and cannot secure or authenticate itself. This fact is an obstacle for a service based mechanism approach. Also, the luck of content security and authentication can lead to many unpleasant surprises e.g. the hacking of the client-side content and the exposure of the correct answers in one test exercise.
Finally, the sequencing as implemented in the 2004 SCORM version although it allows almost any sequence possibility among the SCOs still is very complicated and demanding job. In 1.2 version was a lot simpler but the possibilities were only limited to user chosen sequencing. The complexity of 2004 sequencing led many developers, to implement sequencing internally (using SCORM 1.2).
All these factors blended with the popularity of the open source notion, changed the e-learning's community needs and demands, asking the e-learning scene to change facing the new requirements.

5 The Training and Learning Architecture (TLA)

The problems previously mentioned, led to the continuation of the research concerning the e-learning standards and protocols. The aim now is the creation of the so called next generation of SCORM under the code name Training and Learning Architecture (TLA).

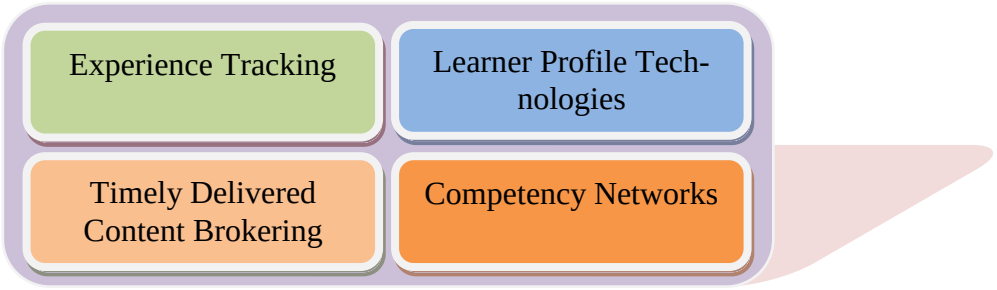

Fig. 2. The components of Training and Learning Architecture (TLA)

The first step of this ongoing process of creating the future edition of SCORM was the Tin Can API (latter known by the name experience API) that has as a purpose the tracking of the various experiences of the learner (when using the content). This experiences tracking can be achieved by the use of a LRS (Learning Record Store) component in order to record to and retrieve from these experiences.
The Components of the Training and Learning Architecture (TLA) are shown in the figure 2.
As we see in the figure 2 the Training and Learning Architecture (TLA) is a general term, incorporating various different projects that shall take place as part of this ongoing process of evolution.

5.1 The experience API (Tin Can API)

The initial phase for the construction of the TLS architecture is oriented in the tracking of e-learners experiences and is done by introducing the following new components:

- runtime API,
- data model,
- data model format/syntax,
- communication transport/store/retrieve/ security methods.

This initial effort done led to the first Tin Can API draft in September 2011. Latter to the 0.9, 0.95 and 1.0 Tin Can API specification release (June 2012, August 2012, April 2013) [5].
The important thing is that ADL through a Public inquiry, managed to gather from the e-learning community (both at public and private sector) their feedback, views, perspectives, and even votes for the establishment of top requirements that should be set as fea-

tures implemented in the Experience API. These requirements can be summarized into the following [13]:

1. **Tracking of applications that run outside the environment of a browser, package or LMS.** This allows many possibilities. For example the implementation of native mobile applications with possibilities to track almost anything. Or the tracking of Rich Internet Applications (like Adobe Flash, JavaFX, and Microsoft Silverlight) that can also run without the need of a browser or LMS. Old fashion methods like packaging are not regarded anymore necessary and are not demanded.
2. **Content to be tracked when non-permanent connection to the internet exists or limited network access.** Handy for scenarios where mobile users connect to the internet for limited periods, download content, use it and reconnect to send their activity's results and statistics.
3. **Tracking of distributed content and system models.** SCORM allowed the tracking of specific events (start time, end time, score, pass/fail, etc.) from a source to one LMS. What now is requested is the possibility to track almost any experience from any application or distributed content place and from many different distributed systems.
4. **SCORM working as a web service**. This requirement allows content security and authentication as well as distributed content tracking possibilities.
5. **Robust way of saving information and retrieving them.** By that requirement we can track information, save them in one experiences record store and later retrieve and process the results and the various experiences recorded. Doing so, it's becoming possible to exclude useful statistical results, combined info and any useful conclusions derived from the experiences tracked. These data can be shared among students and the instructor in a secure way to support various educational models and scenarios (e.g. competition among learners) [9].
6. **Communication mechanism with updated features**. This requirement includes more simplicity and freedom so that more complicated scenarios and bigger variety of activities (almost anything) to be possible to be tracked.
7. **Social, group, mobile, games and cooperative learning scenarios possible**. A requirement that makes possible scenarios that need exposure of data to 3rd parties like the activity mentor and the learners (group, team, social learning) but also keeps the evaluation data (test, quiz, assessment, etc) secure. By allowing this kind of scenarios we motivate users to engage more [9].
8. **Power joined with simplicity and low cost.** Track everything everywhere in a simple manner and low cost. Improved and simplified sequencing or no content sequencing**.** Bigger freedom and faster ways of implementation.
9. **Tools, Manuals, best practices and open source software offered.** These recommendations will allow the implementation of the future e-learning to be realized in a rapid and simple way, with low cost and also available to smaller organizations and enterprises.

All the previous mentioned requests, had as a result the formation of the first component of the Training and Learning Architecture which is the experiences tracking part with the code name 'experience API'.

The experience API, in a nutshell, is one simple internet service that permits the transfer and storage of phrases that can be used to declare facts, actions and experiences, but also any kind of activity information, formed specifically. These specially formed phrases are called statements. The statements are formed as objects and give to the learner's the possibility to announce to the tracking system service their experiences and results. These experiences can come from every possible kind of content, platform, from SCORM legacy content from in and out of browser solutions.

Statements are formed by three components the Actor, the Verb and the Object and are forming simple but concrete experience statements in a Actor-Verb-Object syntax (or in a "I(actor) did(verb) that(object)" order). One example of the simplest statement that can be formed is the following:

```
{
    "id": "876543210-4321-8765-1234-123456789012",
    "actor":{
        "mbox":"mailto:petros@ase.ro"
    },
    "verb":{
        "id":"http://adlnet.gov/expapi/verbs/created",
        "display":{
            "en-US":"created"
        }
    },
    "object":{
        "id":"http://example.ase.ro/xapi/example/activity"
    }
}
```

Examples of such formed statements can be:

```
George scored "75% on The Second Balkan War, Assessment"
Emilia completed "The Advanced English Stage 1, Training"
John answered "the question 1, with true"
```

The Statements are transported and stored in a special database called Learning Record Store (LRS) in a secure manner. The LRS is not an LMS (A group of specifically defined robust services). The LMS only Implements the tracking service but in a more rich way than SCORM does. The data in the LRS are property of each user but also can be shared among them and their tutor from and to every possible device. Furthermore experiences from various different systems can be bond together and used to form for example more complete profiles by using statistical data from the various tracked experiences.

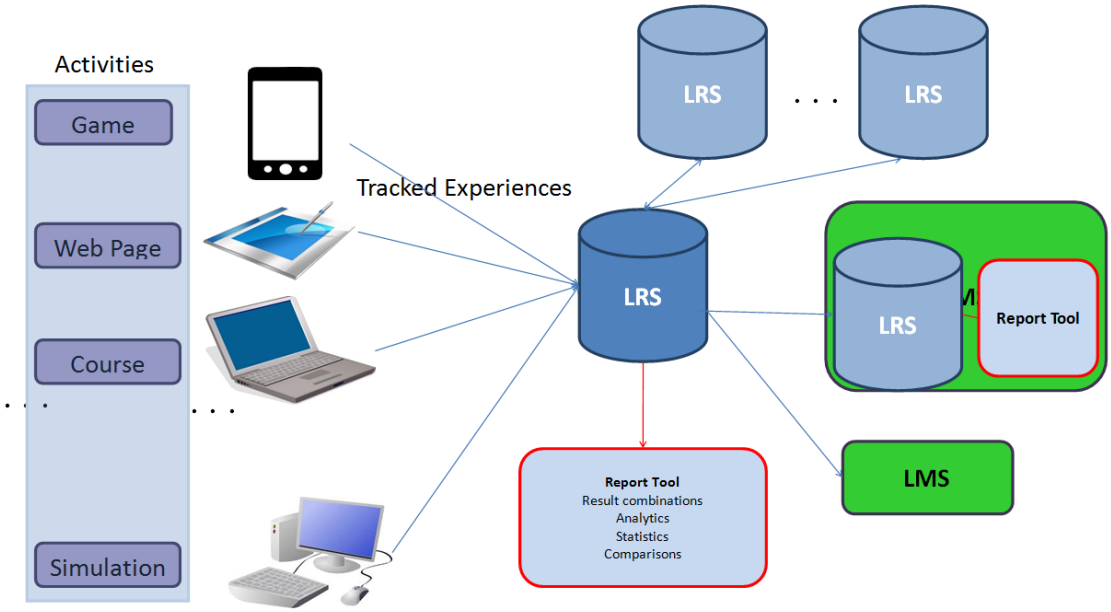

Fig. 3. A schematic representation of the experience API functionality

The experience API is designed to be used as a service by the activity implementers in order them to track the experiences of the users using their activities (e.g. when started one activity, at what level they are at every moment, how much they scored, etc.). By that way they can track every information needed, calculate useful statistical data and information regarding the experiences tracked, but also share these info, to the user, other users of one group or their tutor, depending on the scenario used.

For this specification to work as intended a specific data model is provided to be used and also are provided the necessary components that are needed, in order this experience's tracking system to work.

A schematic representation of the experience API functioning can be seen in the figure 3.

To sum up the experience API offers ways and methods:

- to define and form statements
- to transfer, store and retrieve statements (formed as objects) in a LRS
- to securely exchange information between a LRS and specific content
- to securely exchange experiences among various distributed LRSs.

6 Activity Streams

One Activity stream is a list of informative announcements concerning the activities of one individual. Such implementations have already been popular in various similar uses. The "Activity Streams" is a specification defining open activity stream protocols, which are used to record and announce activities taken in social web applications and web services in general. Contributors of the JSON Activity Streams 1.0 draft are IBM, Facebook, Google, MySpace, Microsoft, VMware [15].

The Experience API is following the basic principles of Active Streams [10]. Like the activity and statement similarities and their foundation syntax (in both cases of 'actor + verb + object'). Their difference is that the Active stream specification is focusing on the tracking of the actions of a publisher, whereas the experience API is focusing mainly on the results of a learner [14]. Furthermore with the Tin Can API we can add definitions which are some kind of metadata form and also further information concerning the use of the activity in the "context" object.

7 The Present and the Future

Except from the experience tracking part, that is still forming, the ADL and the research community shall focus into the creation of other components that shall form the Training and Learning Architecture as a group of services for providing various different tasks. Services that shall focus to the following sectors [8]:

- **Content brokering.** Through content management software, shared repositories and smart components to provide shared content adjusted to the learner's profile and available device or multiple devices.
- **E-learners profiling and categorization**. User profiles where can be stored and retrieved accordingly useful information to provide a more learner centered education experience, focused in the educational needs and preferences of every individual learner. Examples of information stored in the profiles can be various user selections and preferences, user levels, personal info, skills, interests, learner categorization and tutor's configuration of various parameters but also any other information that can help the personalization and further assistance, through knowledge of this crucial information. These information shall be used as an informative input to the e-learning content.
- **Competency Networks creation**. Systems and content connected in such way to achieve the learning objectives, environment and competency asked). A framework of learners (nodes) and tutors (mentors) creating a network that can train and interact together using various different environments (like virtual worlds, casual e-learning games, serious games and simulations), and is evaluated and tracked throughout multiple different

devices and platforms.

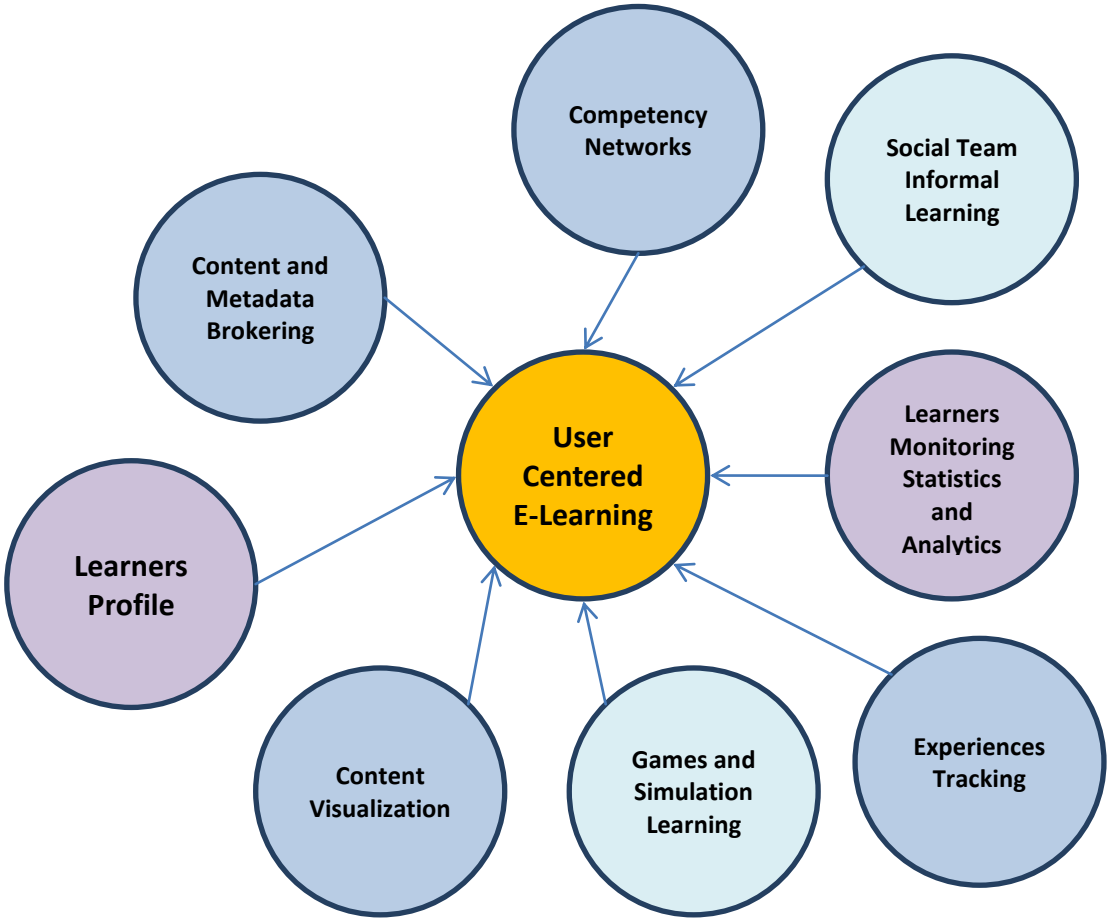

Fig. 4. The future of e-Learning

The aim is the creation of assisting environments, tailored for the personal needs of every user. Also to provide the learner with the learning materials needed and in the most appropriate form to suit the learners profile but also to achieve further communication and stimulation of the learners interest. E.g. one of the aims is the creation of the so called 'Personal Assistant' that shall respond appropriately to the learners needs providing a personalized way of learning using various technologies, platforms and devices and by providing to the learner constant communication and encouragement.

Furthermore the learner of the new era shall receive personalized way of learning adjusted to his/her specific needs and cognitive adaptability demands. This shall be achieved by positioning the learner in the center of the future learning scenery (figure 4).

8 Conclusions

SCORM is evolving into the Training and Learning Architecture adjusting to the demands of the present and future. The first step (the experience API) of this ongoing process is showing us that the future of e-learning shall be a lot more simple, adjustable, modular and powerful. By that way e-learning developers shall have to their disposal open standards, open shared repositories and open source tools to use and to provide new ways of teaching and learning. New types of activities shall appear enriched with more engaging elements (smart tutors, cooperative learning, combined experiences learner's profiles, immediate mentor interaction systems, etc.) leading to a team-group-social e-learning network realization model. Content enriched with game elements, games

scenarios, and with e-learners profiling. Giving to the e-learners, content tailored to their cognitive needs, in various engaging forms and used in every possible device.

References

[1] SCORM Users Guide for Programmers - Version 10 - DoD and ADL - September 15, 2011
[2] SCORM 2004 4th Edition - Content Aggregation Model (CAM) Version 1.1 - Paul Jesukiewicz, Director - Advanced Distributed Learning Initiative Office of the Deputy under Secretary of Defence for Readiness and Training, Policy and Programs - August 14, 2009
[3] SCORM 2004 4th Edition - Run-Time Environment (RTE) Version 1.1 - Advanced Distributed Learning Initiative Office of the Deputy under Secretary of Defence for Readiness and Training, Policy and Programs - August 14, 2009
[4] SCORM 2004 4th Edition - Sequencing and Navigation (SN) Version 1.1 - Advanced Distributed Learning Initiative Office of the Deputy under Secretary of Defence for Readiness and Training, Policy and Programs - August 14, 2009
[5] Sharable Content Object Reference Model Wikipedia, Available at: http://en.wikipedia.org/wiki/Sharable_Content_Object_Reference_Model
[6] In the Eye of the SCORM (An introduction to SCORM 2004 for Content Developers) - Claude Ostyn - Update 0.9-8.8– March 2007
[7] SCORM 2004 4th edition – ADL site - http://www.adlnet.org/scorm/scorm-2004-4th
[8] The Training and Learning Architecture - ADL site, Available at: http://www.adlnet.gov/introducing-the-training-and-learning-architecture-tla
[9] Gamification by design- Gabe Zichermann and Christopher Cunningham - O'REILLY - 2011
[10] The Next Generation of SCORM: Innovation for the Global Force - Jonathan Poltrack, Nikolaus Hruska, Andy Johnson, Jason Haag - ADL-2012
[11] The Experience API – Version 0.95-ADL-Sep 2012
[12] Tin Can API (REST binding)-ADL-Oct 2011
[13] An ADL Perspective on Next Generation SCORM Requirements as Derived from Project Tin Can-ADL-Jan 2012
[14] Tin Can API Comparison with Activity Streams - Andy Johnson - ADL Site - http://www.adlnet.gov/tin-can-api-comparison-with-activity-streams
[15] JSON Activity Streams 1.0- J. Snell(IBM), M. Atkins(SAY Media), W. Norris(Google), C. Messina(Citizen Agency, Google), M. Wilkinson(MySpace, Facebook, VMware), R. Dolin (Microsoft) - May 2011 http://activitystrea.ms/specs/json/1.0/#activity
[16] ADL Overview, Available at: http://www.adlnet.gov/overview
[17] Tin Can API homepage, Available at: http://tincanapi.com/

Petros PAPAZOGLOU PAPAZOGLAKIS graduated in 1999 from the Technological Educational Institute of Piraeus, department of Electronic Computer Systems Engineer. In 2012 graduated from the IT&C Security Master at the Faculty of Cybernetics Statistics and Economic Informatics of the Bucharest University of Economic Studies and since October 2012 is continuing his studies at the same university as a PhD student. During his occupation at the Greek National Center of Scientific Research, and the Technological Educational Institute of Piraeus, he worked in various research projects in National and European level.