

Learning Outcomes Design Authoring Tool – The educator is not alone!

Dr. Tania Al. Kerkiri

*Dept. of Applied Informatics, University of Macedonia
156 Egnatia Street, 54006,
Thessaloniki, Greece
E-mail: kerkiri.tania@gmail.com*

Dr. Spyros Papadakis

*School of Science and Technology,
Hellenic Open University,
Patras, Greece
E-mail: papadakis@eap.gr*

ABSTRACT

Creating a course is a demanding process that presupposes the scientific knowhow and various technical skills on effectively delivering content. Contemporary Learning Management Systems (LMSs) do not adequately support the structural appropriateness of content, despite being able to provide the lesson. Moreover, they are not readily capable of determining learning outcomes – the essential factor against which the effectiveness of a lesson is validated. In the current work we propose and describe the Learning Outcomes Design Authoring Tool (as of now LeODAT), which is a web-based tool conceived and construed to support correct coding and authoring of a course by additionally creating descriptors for its learning outcomes. Furthermore, LeODAT comes along with a number of pedagogical theories, organized as Taxonomies of Learning Domains. LeODAT is designed to be used as a standalone application or as a service into worldwide popular LMSs, such as Moodle, LAMS etc.

Keywords: learning management system; intelligent tutoring tool; learning outcomes; learning taxonomies;

INTRODUCTION

Educational approaches have been decisively enhanced by applications of e-Learning, the multiple benefits of which have been extended in every educational level, predominantly assisting with adult learning experiences. As a result, Learning Management Systems (commonly abbreviated as LMSs), i.e., the contemporary means of delivering this type of educational provision, are constantly enriched with new capacities in order to support more, and qualitatively richer services.

While LMSs are advancing, educators advance too in creating numerous courses, more or less based on educational specifications and standards, most popular

among them, IMS 0and SCORM. Such standards contribute to high quality lessons. However, even the most well-designed, inspirational, edifying lesson may not lead to the desired educational results inasmuch as its comprehension, absorption and functional incorporation in the user's cognitive repository is concerned. Most often, lack of sufficient psychopedagogic knowledge on the part of educators themselves is the prevailing factor causing course inefficiency.

We claim that if the tutor, during the course-authoring process, had a template that would guide him/her through the steps s/he should follow to achieve an efficient course, the authoring process would totally improve.

A visible solution to this is the creation of courses on the basis of prepared templates for constructing lesson modules.

A considerable number of e-learning tools have been presented so far in order to support educators create learning plans. Some of the well known are: Phoebe, London Pedagogy Planner, LearningMapR, Compendium, QUT's Learning Design Templates and the LAMS Activity Planner.

Nevertheless, these tools propose course construction designs that do not cover for obtaining meaningful learning outcomes.

Previous research (Kerkiri & Palaiologou, 2009; Papadakis & Giglione, 2009) shown that the existing LMSs lack capabilities that meet these needs. A focused effort is here presented, with an aim to fill this gap: the Learning Outcomes Design Authoring Tool (LeODAT), a web tool, based on taxonomies of learning domains initially introduced by White & Gagné, 1974). LeODAT is based on emergent web technologies and novel practices applied in web application design. The role of this tool is twofold: firstly, it comes to aid the educators in writing clear/measurable learning activities and secondly, it intends to augment the accessibility of the learning theory.

The rest of this paper is organized as follows. Initially, some theoretical concepts, along with the problem that (non-specialist) educators usually face in explaining the design rationale are presented. Consecutively, the architecture of the proposed Learning Outcomes Design Authoring Tool (LeODAT) that comes to support this task is described. Finally, a meta-level part of LeODAT, developed as an add-on service in current LMSs, is described, along with its use and its future perspectives.

THEORETICAL FRAMEWORK

LMSs are web-based applications of various built-in capabilities facilitating online distanced educational interventions.

Contemporary LMSs:

- i) support provisions of educational modules in a variety of forms (e.g., docs, html, ppt presentations, video/audio files, different types of survey and questionnaire- investigations, etc), with respect to the individuality and idiographic preferences of users;
- ii) hold up a collection of tools multifariously assisting with course construction and apt modular assemblage. Such tools include chat rooms, e-mail exchange sites, file sharing, teleconferencing, electronic support of student progress, surfing history archives, student

group coordination, estimates of group learning outcomes, etc).

A closer inspection of the LMSs capabilities, however, tends to reveal that their current components do not satisfactorily cover for educator support in a psycho-pedagogically correct fashion and this tends to prove detrimental for especially inexperienced educators in the e-learning processes; educators are still burdened with the difficult task of orchestrating the entire learning activity without necessarily having access or being familiar with a host of theoretical tenets that would enhance, accelerate and optimize their creation both conceptually and contextually. They decide upon the content, they plan the method, and they assess the learning outcomes –a crucial multitask process which would ideally be also one of LMSs assignments relieving educators and freeing up their time for deepening their scientific insights on the subjects at hand.

An important element for efficient course construction is for the author to obtain learning outcome milestones beforehand. One such learning outcome milestone that is unwaveringly defined is twofold: it enlightens the educator's clear vision as for the content of their teaching and the user's clear understanding as for what has to be achieved. It follows that a number of consecutive benefits are secured:

- a) the tutor is furnished with a guide to help with building the course;
- b) the risk of a biased evaluation of the learner is reduced through a clear set of measurable performance definitions,
- c) self-assessment is allowed,
- d) a clear sequence of steps is offered to the learner to reach the educational outcome.

To accomplish these, the tutor possesses two distinct means; namely, the Learning Outcomes and the Educational Models –concepts that are elucidated in the following sections.

In Kennedy (2006) and Smith & Ragan (2005) can be found initiatives of Learning Outcomes. Learning Outcomes or Objectives are statements that clearly describe the goals a learner should be able to reach and be able to demonstrate that has conquered after completing a learning activity. A well-constructed learning objective must contain three features: the conditions under which the behavior is performed, a verb that defines the behavior itself and the scale (criteria) to which a student should perform the behavior. If any one of these three components is missing, the learning outcome is not considered

accurately described. Thus, Conole & Fill, (2005) claimed that learning objectives, serve both educators and learners: the educators can use them to precisely express their expectations from their learners and the learners can use them to easily prove and self-assess their achievements. To conclude, the learning outcome approach is a perspective and a mode of thinking in developing valid sequences of learning activities.

Educational models, on the other hand, illustrate the relationships between learning outcomes and actual progress of the learner, as the latter moves along the 'novice-to-expert' continuum: new ideas are created by integrating previous ones along with sensory input (perceptions). Every higher level concept is based on lower level information (Sein, Bostrom, & Olfman, 1999). The dependency of newly created knowledge on the old one proves the hierarchical knowledge construction; hence a hierarchy is a method to simulate the way human knowledge is constructed. This is already realized in educational theory, and thus the educational models are designed as taxonomies.

Generally, a taxonomy is a collection of terms of a vocabulary organized into a hierarchical structure. The construction of taxonomy is the first step towards structuring domain knowledge, and this hierarchical knowledge organization influences all areas of education. An initial effort that sets the basis of learning taxonomy creation is presented by Bloom, Engelhart, Furst, Hill, & Krathwohl, (1956), wherein a methodology for writing learning outcomes was described in accord with the taxonomy of Bloom, a worldwide accredited classification of immense impact in scientific productions. Bloom identified 3 levels of thinking activated during the learning process: cognitive, affective and psychomotor. Within each of these levels, he recognized an ascending order of complexity. His taxonomy provides the framework upon which one could build prior knowledge and, consequently, develop more complex levels of understanding.

In recent years, attempts were made to revise Bloom's Taxonomy (Biggs, & Collis, 1982) with some, apparently minor, but actually significant, modifications. Concurrently, several taxonomies have appeared, such as the Critical Thinking framework by Matsagouras, (2001), and, perhaps most importantly, the SOLO framework, provided by Biggs & Collis (1982). The Structure of Observed Learning Outcome (SOLO) taxonomy provides a systematic method to describe how a learner's performance grows in

complexity while mastering many tasks, particularly the kind of tasks undertaken in a given learning period. Notably, concurrent LMSs do support 'tree structures', or 'forking structures' for creating a course. It is therefore accurate to assert that these systems reflect the cognitive-constructivist approach and are thus methodologically updated and appropriate to electronically support educational techniques (Kerkiri, Paleologou, Konetas, & Chatzinikolaou, 2010). However, they do not offer tools that create learning outcomes based on hierarchical structures (Kerkiri & Palaiologou, 2009).

An initial effort on creating this kind of service was introduced in Papadakis, & Giglione, (2009). The "Cognitive Skill-based Learning Objectives Wizard (CSLO-Wizard)" was proposed as an add-on in LAMS. Although CSLO-Wizard is a useful attempt, it is very primitive and bares several limitations, including that it cannot handle dynamic/multilingual content; it cannot handle multiple learning taxonomies; and it does not provide capabilities in administering the content and the access of users upon it. A substantial effort was made in Kerkiri, & Papadakis, (2010) which concluded to the Learning Outcomes Authoring Tool (LOAT). LOAT handles hierarchies by emphasis on creating learning outcomes. In this article we extend LOAT, and provide our Learning Outcomes Design Authoring Tool, (LeODAT), which supports the total course creation. Moreover, we have developed this tool as an add-on in the very popular LMS Moodle v2.0, as well as, in LAMS, v.2.

LEARNING OUTCOMES DESIGN AUTHORING TOOL (LeODAT)

According to prominent pedagogic theoretical tenets (Moon, 2002) a lesson plan has to be well organized and comprised by: Lesson title, Scenario, Target group(s), Time and timing as well as Application schedule. At a consecutive refinement stage of the learning plan design each schedule has to be broken down into small units, for each of which a pilot micro-teaching process has to be followed, in order to highlight the methodological facets for transmitting the whole module.

Each pilot application has its own forking structure, comprised by: Section title, Duration, Aim/Scope, Scheduling.

In turn, within each schedule one can identify 3 basic parts: Introduction, Main theme, Conclusion(s). Furthermore, within each part 4 characteristics are predominant:

- i) part title,
- ii) estimated duration,
- iii) activities and educational tactics enhancing the content; and finally
- iv) educational means and materials for apt usage.

The pedagogical theories clearly define that the forking structure is the most appropriate perceptual structure to facilitate mind mapping and augment micro-teaching performance.

In our model, the forking structure is dual:

- a) regarding the course creation: represents the lesson organization, as seen in Figure 1., which also demonstrates the capabilities of the administrative part of the LeODAT; and
- b) regarding the specification of the learning outcomes: represents pertinent cognitive models while their content denotes the underlining learning theories.

By pilot simulations of the functional feasibility of the workability and practicability of the forking structure (Moon, 2002) was also verified, in that it is patterned in a way and can be offered as a template instructing the educator while organizing his/her lesson(s).

We, thus, designed and developed a solution for the LMSs that support the methodological steps that have to be followed for reaching these guidelines. Our effort concluded in the Learning Design-Authoring Tool (LeODAT). Within LeODAT a template is created for securing the hierarchical architecture.

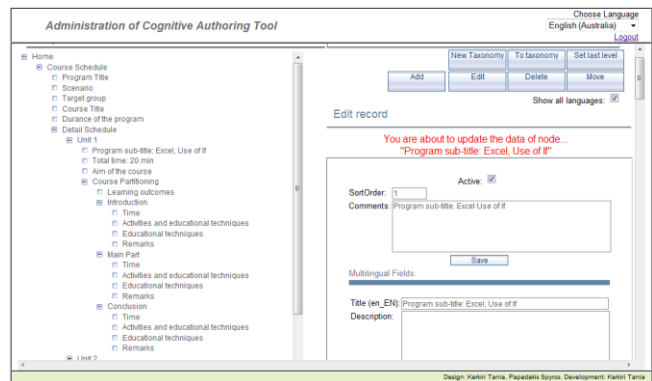
In the last level of the creation of a course, when the actual micro-teaching performance is created, the LeODAT is expanded to support the lesson focused on achieving specific learning outcomes. To manage this outcome, a number of learning taxonomies were embedded.

As each learning taxonomy is structured as a hierarchy, any of them can be described through LeODAT. Consequently, we have created patterns for each known learning taxonomy, e.g. up to date we have mapped SOLO, Matsagouras, Blooms' and revised Bloom's, and, lastly, Critical Thinking.

No limitation on the number of the taxonomies exists.

In the last level of the LeODAT a number of patterns for these learning taxonomies exist, each one having its copy in each available language of the system, which, in turn, locks a unique and specified identity for each node.

Figure 1. The LeODAT administrator environment



As we came to realize the LMSs shortcomings for the learning communities, we designed and implemented LeODAT, using mature programming techniques and emergent open-source web technologies. LeODAT is a web application that provides a number of features in handling hierarchies. Since the initial intention for providing this tool was for it to become a plug-in in a number of different LMSs, it was designed according to the standards followed by the most popular ones. These standards include techniques and technologies, as:

- 1) multiple language support in both the interface (known as i18n) and the content itself, as achieved by using existing locales of the underlined LMSs,
- 2) a client-server architecture, that distinguishes the users' actions from the data which are actually located on the server-side of the proposed infrastructure,
- 3) the use of open-source tools that can be used to conduct existing LMSs, such as web-servers (e.g. Apache-Tomcat), databases (MySQL), remote management database tools (MySQL Administrator Tool), and web programming languages.

Finally, technologies of the Web 2.0 play an important role in this tool by extending the web-based e-Learning communities with new capabilities.

In LeODAT a dual capacity was implemented:

Firstly, an administration web tool was created that handles the taxonomies of cognitive theories. These tool is available to a group of educators, having pedagogical knowledge and administration capabilities, and,

Secondly, a web tool was provided that makes this content available to all educators during the creation of a course.

This part of LeODAT comes along with a detailed extension of content of the pedagogical theories.

The architecture of LeODAT

Both parts of LeODAT implemented as 3-tier client-server architecture. Each capability of the client-part is supported by a corresponding socket in the server-part. The server-side business logic and data processing is accomplished through java technologies. In the client-side, standard web technologies are used, i.e. html, jsp and javascript, to control the data flow (Papadakis, & Kerkiri, 2010). The communication between the two parts is made through beans, JSP technology and AJAX-calls (Musser, 2007).

For future extensions of this tool, the data that are fetched from the server are located in intermediate xml-files. From these files, they are properly decoded and built the client-side interface on-the-fly.

According to Fields, Kolb, & Shawn, (2001), the main principles that lead the design and implementation details of this tool are:

- 1) The data errors are mainly trapped in the front-end, via extended check-ins, before the data are sent to the database –thus preserving the database integrity and preventing the network/server/database from useless usage. Although the structure of the taxonomy is hierarchical, a relational database inheritably should be used to prevent the compatibility to the existing LMSs. Consequently, several functionalities were implemented in order to simulate the methods used by such systems in traversing the nodes of hierarchical databases and xml-files.

- 2) Two kind of users are foreseen, having distinct roles: the administrators and the end-users (namely, the creators of the lesson). This is achieved by clearly distinguishing their authorities in the system, through different access-points: the administrator's module is separated from the educator's module. The former handles the educational taxonomies, and the latter is aided by these pedagogical data while constructing their lessons.

- 3) To further facilitate the development and to make LeODAT able to incorporate into various LMSs (each having its own design), the application logic was also distinguished from the interface of the application. This achieved using the Model-View-Controller (MVC) architecture, which isolates the application logic from the input and presentation, permitting independent development, testing and maintenance of each.

Specifically, the business logic is implemented by using client-server routines written in javascript and jsp, or server-side java-programs, while the view is totally based on style sheets. Moreover, major attention

was paid so that to achieve a uniform and user-friendly behavior all over this tool. The design of the interface is conducted by HCI-principles 0. Moreover, a number of multilingual messages were foreseen, firstly, to guide the users, and on the other, to inform about the consequences of their actions.

The functionality and main features of this tool is described in the following sub-section.

The functionality of LeODAT

The administrator's interface is divided into two parts, which are shown in Figure 1.

On the left-part, a tree-structured view of the educational taxonomies is presented. The right part is also divided into two other parts:

- a) a standard menu, which is located on the top. This menu defines the features of the system concerning the management of taxonomies and their nodes. Each button of the menu clearly corresponds to an available action, and
- b) a frame, which is located below this menu. This frame is flexible and its appearance totally depends on the action being performed. Its functionality is also dependent on this action.

In each node of the taxonomy two events are attached, both implemented by AJAX-calls. The first one is attached to the preceding icon of each node. This event retrieves the current node's sub-nodes (these correspond to the records of the database having the same parent-ID), expanding / collapsing these sub-nodes, accordingly. The second event is attached to the node itself. This last event retrieves detailed information for the specific node.

The available functions on the data, as specified by the menu buttons, are:

1) Language selection

It changes the language in which the interface is shown, as well as, the language the data is presented. The appearance of the frame on the right part depends on the selected language, as well. It is affected as follows:

Case 1: If English is the default language of the user, then input fields for a single node are created just for the English language.

Case 2: If English is not the default language of the system, then fields for two languages are created, both English and the native language of the user. Finally, a third option is available:

Case 3: If the user wants to handle the data of a node in every available language of the LMS, then input fields in all these languages are created.

In an open net community, as the e-Learning communities, LeODAT operates as follows: users with administrative capabilities prepare the content of the taxonomy in their native language. They may use it as a prototype in order to translate it in their own spoken language. They may also expand on the content of the original taxonomy, thus offering additional knowledge to the users within the community.

2) Create a new taxonomy

New taxonomy: creates the root node of a new hierarchy, and initializes its parent Id to 0.

3) Add new nodes

Add: adds a new node in the hierarchy under the selected node.

In any of the cases described above in ‘**Language selection**’ functionality, the user has to provide data at least for the English language. Even if the input fields in the other languages are not filled in, this tool still fills them with the English data. This means that the English language is the common basis for all the users of this tool. The users can use the English description of the pedagogical content of the taxonomies, and translate it into their own language.

One of the **main features** of LeODAT is that in any of the above three cases, it creates the same hierarchy for every language of the system, (even if only the English description for the data of a node is provided). LeODAT gives the same identification for the same learning outcome of any learning resource. This means that each node of the taxonomy has the same Id in every language.

Consequently, this creates a template for any educational taxonomy regardless language. This means that every taxonomy has the same structure in every language.

As a result, all courses acquire one and unique structure in the LMS constellation, and each node one and unique identity, regardless of the original language input.

4) Edit a node

Edit: the detailed data of a node, along with their multi-lingual descriptions, are presented for editing. Once more, fields for **i)** a single language (English only), **ii)** two languages (the user’s native language,

plus English), or **iii)** the total languages of the system are presented, as described above. The user, though, is allowed to change only the data of the language s/he desires –the rest of the data is not affected.

5) Delete a node

Delete: deletes a node along with its sub-nodes, after clearly warning the user about the consequences of his/her action.

6) Move a node

Move: moves a node in another branch, along with its subsequent nodes –still preserving the hierarchy of the branches. Cycles are not allowed. Consequently, it is not allowed to move a node under a sub-node.

7) Convert a node to a new taxonomy

Convert node to taxonomy: converts a branch of the tree to a new taxonomy. This function still maintains the hierarchy of the nodes.

8) Set the level of nodes return an output

Set last level: defines the lower level of the taxonomy. The depth is limited to 255. The lower level is the same for each node of the taxonomy. The way this last level is used may vary e.g.

i) it can force the user to provide the same number of levels in each branch of the taxonomy. This assumes that each educational taxonomy has the same depth in every cognitive skill,

ii) no more levels are allowed after this last level, neither in the creation of a new node nor in the movement of an existing one,

iii) in a meta-level application, each node that has been defined as final, will be treated as a leaf.

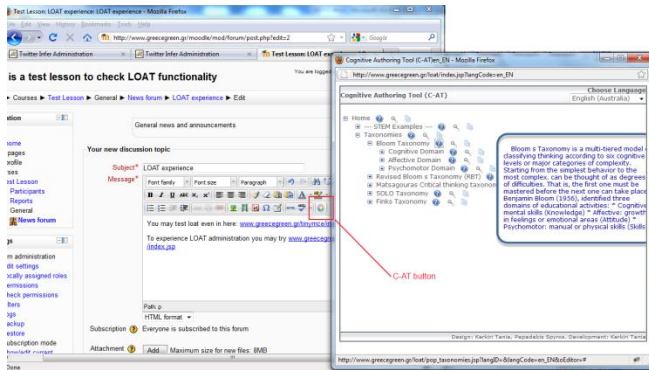
In Figure (2) a screenshot of the administration interface of LeODAT is presented. Στο συγκεκριμένο σχήμα γίνεται επίδειξη της δομής του μαθήματος.

LeODAT IN ACTION

LeODAT runs as a standalone web application handling hierarchical organized data. Moreover, a *meta-level view* of LeODAT data was implemented highlighting its actual capabilities. This part is available as a plugin in popular html WYGIWIG editors (see Figure 1), such as FCK and tinyMCE, which are successfully embedded in a variety of web-applications –including LMSs.

In Figure (2) we can see LeODAT as a plugin in Moodle, v.2.0. The tool is also available in LAMS.

Figure 2. LeODAT as a plugin in Moodle



This part of LeODAT is mainly addressed to the end course-creator.

It comes along with a great pool of pedagogical data that describe the learning theories, and a great collection of verbs that handles learning outcomes. The contribution of the learning community fulfilling this pool is precious.

Its capabilities are actually emerged while the educators create their lessons, as follows: the educator has all the pedagogical frameworks available, each one described by its taxonomy. While s/he navigates over the nodes of the taxonomies receives “hints” in order to build up learning activities. This is achieved through a window that pop-ups by the ‘mouse blur’ event –which is attached to the exclamation mark of each node– and triggers a server-call through an ajax-call. This window presents the cognitive theory described by the underlined node.

More details of the selected pedagogical theory are shown while the tree expands.

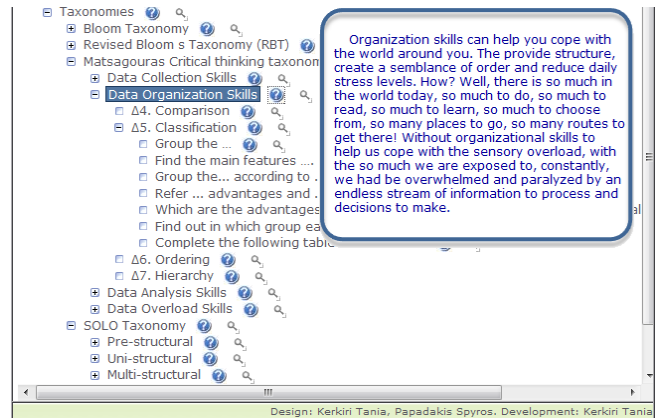
In the last level, the educator finds out which learning outcome to use so that to achieve the proper objective.

Let’s see an example of how this really works:

Imagine the educator creating his/her lesson using LeODAT. S/he can choose the pedagogical framework (a taxonomy indeed) s/he wants to follow and then to navigate through its nodes –already available as part of our effort. Let the educator choose ‘**Revisited Matsagoura’s Taxonomy**’. After that, s/he can even deepen in the levels of this theory. S/he can decide the cognitive skill s/he wants to improve, e.g. ‘**Data Organizational Skills**’.

When s/he wants assistance, s/he can open the pop-up window that presents the underlined cognitive theory – this is depicted in Figure (3).

Figure 3. LeODAT in action



Even deeper, in the last level, s/he can *find out* and, moreover, *select* the appropriate verbs/expressions (in fact, ‘learning outcomes’) that are proposed by this theory to achieve this specific outcome. While the educator selects the node, LeODAT transfers the selected text into the editor.

After that, we can easily claim that LeODAT inspires the educators while creating their lessons.

New perspectives

The way in which the course creation process is structured in the LeODAT and the way in which we organized its underpinning components lead already to the next step of further enhancing its functionality: its future version will be able to provide a collection of features concerning the conceptual aspects of the course and pertaining to descriptions of its various learning outcomes.

This new capability is implemented based on technologies emerging from Web 2.0. This functionality will be supported through metadata. More specifically, we stipulate an advanced structural format that will store metadata derived from the lesson processes and will successively be integrated in the lesson mapping so as to function as descriptors of the learning goal(s) as arrived at in accord with the selected educational taxonomy.

The purpose of this metadata is three-fold:

a) to uniquely identify each learning outcome, and describe it along with its specific educational purpose, and furthermore in a multilingual way,

b) to link the learning resources to their learning outcomes, and

c) to be used in searching for learning resources that fulfill the specific learning outcomes. In addition, a search mode is designed to be incorporated, in order to facilitate tracing and back-tracing of lessons.

An important advantage of our work is that this mechanism will make it possible to retrieve educational modules regardless of the original language in which they were initially presented. The ensuing benefit is that the way in which we have built the infrastructure supporting lesson creation, i.e., its coding regardless of language and specific cognitive field, constitutes a valuable tool for the entire educational community which serves in the easy tracking, retrieving, modifying, adapting and re-using endlessly a large variety of learning design.

Due to LeODAT is designed to be used in current LMSs, and is already available as an add-on on popular LMSs (e.g. Moodle, LAMS) through the HTML-editors they embed it could be a bridge among them.

Ultimately, our vision is to de-stress the educators by covering for mundane technicalities of course launching and even for conceptual aspects of course content and assessment, in order to unswervingly focus on what their status entails, to enhance substantial educational expansions and novel knowledge production. Our midway goal is therefore to verify the usefulness of our new tool, as already available in various LMSs, with as many educators as possible, with a view to test and possibly expand its feasibility in a wide range of edifying subjects. Our ambition is also to corroborate our conviction that course creation based on pedagogically correct educational templates which exploit taxonomy principles to the ultimate degree in providing meaningful services to learners, does indeed ameliorate the very learning processes.

CONCLUSION AND FUTURE WORK

This article presents the structural components and the underlying rationale of the Learning Design Authoring Tool, by acronym denoted as LeODAT –a new tool for LMSs inclusion, which:

- a) assists with the architectural organization of a comprehensive course especially for e-learning purposes;
- b) supports its targeted creation so that it fulfills specific educational needs on defining learning objectives

c) frees from language and content restrictions for creative course launching and

d) boosts educators in the confidence of its exploiting sound pedagogical principles and an accredited taxonomy for both contextual and conceptual structuring.

As the greater the synergy between learning outcomes, teaching strategies and assessment techniques, the more successful the learning process is likely to be, we easily can understand the necessity of this tool.

In its capacity to handle the above mentioned complex issues the LeODAT succeeds in

- a) offering the ability to trace, tackle and retrieve complete lessons and/or partial educational actions related to the learning outcomes and by respect of idiographic particularities of both educational subgroups and individual users and
- b) facilitating interoperability and reusability of learning units.

Up to now, our effort was oriented in:

- a) designing, creating and debugging LeODAT,
 - b) creating a means to integrate it in most popular LMSs
- and, finally,
- c) in collecting a number of pedagogical theories supporting learning outcomes.

We are now prepared for its dissemination, so that to prove that LeODAT is a real assistant for the learning community. As it is imperative (for ultimate success of all participants) the educators must use well-designed learning outcomes that meet demanding behavioral criteria, while getting involved in authoring learning activities, the LeODAT is a real tool for them to be effective Learning Designers.

Moreover, LeODAT is an additional asset to any LMS. The design of this tool is motivated from the desire to aid the learning community during the creation of a lesson and also being part of collaborative LMSs which may challenge to more effective and efficient knowledge management.

REFERENCES

Advanced Distributed Learning, available at <<http://www.adlnet.gov>> [Last Visit] Sep 2011

Babiuk, G., (2005), *Full Bag of 'Tech Tools' enhances the reflective process in Teacher Education*. In C. Crawford et al. (Eds.), *Proceedings of Society for Information Technology and Teacher Education International Conference*. Chesapeake, VA: AACE, 2005, pp. 1873-1877.

- Biggs, J. & Collis, K., (1982), *Evaluating the Quality of Learning: the SOLO taxonomy*, New York: Academic Press. 1982.
- Bloom, D. S., Engelhart, M. D., Furst, E. J. Hill, W. & Krathwohl, D. (1956), *Taxonomy of educational objectives*, vol. 1, New York:McKay, 1956.
- Buzza, D., Richards, L., Bean, D., Harrigan, K., & Carey, T., (2005). *LearningMapR: A prototype tool for creating IMS-LD compliant units of learning*. Journal of Interactive Media in Education (17).
- Cameron, L., (2008), *Developing a pedagogical planner*, In S. Walker, M. Ryan & R. Teed (Eds.) *Designing for Learning: Post-Conference Reflections*. University of Greenwich, London: University of Greenwich, pp. 53-66
- Conole, G., & Fill, K., (2005), *A learning design toolkit to create pedagogically effective learning activities*. Journal of Interactive Media in Education, 2005, vol. 1.
- Fields, D. K., Kolb, M. A., & Shawn, B., (2001), *Web Development with JavaServer Pages*, 2nd Edition, ISBN: 193011012X, 2001.
- Kennedy, D., (2006), *Writing and using learning outcomes: a practical guide*. Cork: Quality Promotion Unit, University College Cork, 2006.
- Kerkiri, T., & Paleologou, A.-M., (2009), *Do open source LMSs support personalization? A comparative evaluation*, Proc. Second World Summit on the Knowledge Society, (WSKS 2009), SPRINGER Communications in Computer and Information Science (CCIS) Series official proceedings, Crete, Greece, 2009.
- Kerkiri, T., Paleologou, A.-M., Konetas, D., & Chatzinikolaou, K., (2010), *A learning style – driven architecture build on open source LMS's infrastructure for creation of psycho-pedagogically – 'savvy' personalized learning paths*, (Ed). Safeullah Soomro, E-learning experiences and future, ISBN: 978-953-307-092-6, Publisher: INTECH ISBN 978-953-7619-23-7, 2010.
- Kerkiri, T., Papadakis, S., (2010), *Design and implementation of a Learning Outcomes Authoring Tool: Supporting the Educator to be Learning Designer*. In IEEE CPS International Conference on Intelligent Networking and Collaborative Systems (INCoS 2010), November 24th -26th, Thessaloniki, Greece. 2010
- Kizlik, B., (2010), *Education Information for New and Future Teachers, 'Six Common Mistakes in Writing Lesson Plans (and what to do about them)*, unpublished. <http://www.adprima.com/mistakes.htm>, 2010.
- Maroudas, A., Katsanos, C. & Avouris, N. (2009), *Human-Computer Interaction: Investigating users' attention mechanisms in Website interaction*. Proc 3rd Conference of Electrical and Computer Engineering Students, Thessaloniki, Greece (in Greek), 2009. «Επικοινωνία Ανθρώπου Υπολογιστή: Μελέτη Φαινομένων Εστίασης της Προσοχής των Χρηστών σε Ιστοσελίδες στο Διαδίκτυο».
- Masterman, L., (2008), *Phoebe Pedagogy Planner Project: Evaluation Report*. JISC Design for Learning programme. University of Oxford, 2008. Retrieved 20/02/11, from www.jisc.ac.uk/media/documents/programmes/elearningpedagogy/P_hoebeevaluationreportsept08.pdf
- Matsagouras, E., (2001), *Teaching Critical Thinking in the Greek School: An Infusion Program and its Effectiveness*, Journal of Cognitive Education and Psychology, vol. 1(3), 2001, pp. 303-319.
- McAndrew, P., Weller, M., & Barrett-Baxendale, M., (2006), *Learning design and service oriented architectures: A mutual dependency?*, *Journal of Learning Design*, 1(3), 51-60. <http://www.jld.qut.edu.au/>
- Moon, J., (2002), *The Module and Programme Development Handbook*. London: Kogan Page Limited, 2002.
- Musser, J., (2007), *Ajax Web2.0 Principles And Best Practices*, Published by O'Reilly media, Inc 1005, Gravenstein Highway North Sebastopol, CA 95472, 2007.
- Okada, A., & Buckingham Shum, S., (2006), *Knowledge Mapping with Compendium in Academic Research and Online Education*. 22nd ICDE World Conference, 3-6 Sept.2006
- Papadakis, S. & Kerkiri, T., (2010), *The Learning Outcome Authoring Tool (LOAT) integration into LAMS: Scaffolding author to write learning outcomes in any LAMS Activity Tool*, In L. Cameron & J. Dalziel (Ed.) Proceedings of the 2010 5th International LAMS & Learning Design Conference "Sharing Great Ideas". Sydney, Australia 2010, <http://lams2010sydney.lamsfoundation.org/proceedings.htm>
- Papadakis, S., & Giglione, E., (2009), *Facilitating teacher/learning designer to formulate Learning Objectives (LO) using a Cognitive Skills based LO-Wizard in LAMS*, The International Conference on Artificial Intelligence in Education (AIED2009), Brighton, UK, 2009.
- San Diego, J. P., Laurillard, D. Boyle, T., Bradley, C., Ljubojevic, D., & Neumann, T., (2008), *Towards a user-oriented analytical approach to learning design*, ALT-J: Research in Learning Technology 16 (1) 2008, pp. 15–29.
- Sein, M. K., Bostrom, R. P., & Olfman, L. (1999), *Rethinking end-user training strategy: applying a hierarchical knowledge-level model*, Journal of End User Computing, vol. 11(1), 1999, p.32-39. Sharable Content Object Reference Model (SCORM), available at <<http://scorm.com/scorm-explained/>> [Last Visit] April 2011
- Smith, L. P., & Ragan, T.J., (2005), *Instructional Design* (3rd edition). Hoboken John Wiley. 2005
- White, R. T., & Gagné, R. M., (1974), *Past and future research on learning hierarchies*, Educational Psychologist 11.1