Design and implementation of a Learning Outcomes Authoring Tool

Supporting the Educator to be Learning Designer

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— Educators usually face difficulties while authoring the learning outcomes of a lesson. The well-designed learning outcomes are a key-factor in creating an efficient educational process, and, on the other hand, the dearth of such services is one of the main/common reasons for the failure of, most otherwise, well-established courses. The available LMSs, though, do not sufficiently support the learning outcomes creation. In this paper, the Learning Outcomes Authoring Tool (LOAT), a web based tool that comes to overcome this common addressed issue is presented, along with its architecture. The novelty of LOAT is that it applies emergent web technologies, initially designed for advanced web environments, to support (non-specialist) educators in creating effective learning courses. More over LOAT comes along with a number of pedagogical theories, organized as Taxonomies of Learning Domains. LOAT is designed so that it can be used as a standalone application or as a service into worldwide popular LMSs, such as Moodle, LAMS etc.

Keyword: *learning management system; intelligent tutoring tool; learning outcomes; learning taxonomies;*

I. INTRODUCTION

A number of powerful web-based educational systems [1] originated from intelligent tutoring systems [2] and adaptive hypermedia [3]. The achievements of Information and Communication Technology (ICT) additionally contribute to their development. Further, the creation of educational resources flourishes, as it is supported from the Educational Technology research. Despite the variety of ICT tools, the plethora of resources being available, and the obvious benefits of their use in teaching and learning, many educators lack the necessary computer skills to take advantage of their potential affordances [4]. Additionally, the LMSs hardly support the (non pedagogical-experienced) educators in creating a course according to pedagogical theories. Still, the educators undertake the total effort of orchestrating the learning activity: they decide the content, they plan the method, and they assess the learning outcomes. International trends in education, though, shift from this traditional educator-centered approach to a learner-centered one, focusing on the abilities the learner should acquire by the end of a course. Consequently, one of the crucial requirements of the modern Learning Management Systems (commonly abbreviated as LMSs) is to support this shift. It is necessary to facilitate the (re)writing of the courses in terms of learning outcomes, clearly describing the goals expected from the learners –an approach that would further determine the techniques the educators could use to achieve these goals. As the existing LMSs lack in capabilities that accommodate these needs [1], [5], [6], a focused effort is presented in this paper, coming to fill this gap: the Learning Outcomes Authoring Tool (LOAT), a web tool, based on taxonomies of learning domains [7], [8]. LOAT is based on emergent web technologies and novel practices applied in web application design. The role of this tool is twofold: from the one side, it comes to aid the educators in writing clear/measurable learning activities and from the other, 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. Continuously, the architecture of the proposed Learning Outcomes Authoring Tool (LOAT), that comes to support this task, is described. Finally, a meta-level part of LOAT, an add-on service in current LMSs, is described, along with its use and its future perspectives.

II. THEORETICAL FRAMEWORK

Learning outcomes are statements that describe the objectives a learner should be able to perform after completing a learning activity [9]. Consequently, learning objectives or outcomes are a tool for both educators and learners: the educators can use learning objectives to precisely express what they expect from their learners [10] and, the learners can use them to easily prove and self-assess their achievements. As a result, a clear set of measurable performance definitions reduces the risk of a biased evaluation of the learner. To further highlight the results of the proper selected learning outcomes, let's mention that in case where narrow and limited learning outcomes are used, an education where creativity and imaginative leaps are highly valued is not supported [12].

A well-constructed learning objective must contain three parts: *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 accurately described [11].

Despite the benefits of well-written learning outcomes, it is a usual phenomenon the non-specialist / novice educators to encounter problems [13], [14] in expressing them clearly. This is due to a number of reasons, the most important of them initiates by confusing '*teaching intention*' and '*expected learning objectives*'.

Teaching intention or aims are the broader purposes of the educational process, whereas *objectives* are the specific steps that lead to these goals [15]. *Objectives* are intentions in measurable terms and are stated as intentions of the tutor. *Learning outcomes* are what result from a learning process Some guidelines to create efficient learning outcomes are: **a)** learning outcomes should be *i)* achievable; *ii)* assessable and *iii)* expressed in a language that both learners and other educators could easily understand; **b)** learning outcomes should clearly indicate the nature and/or level/kind of learning method required.

As shown above, clearly articulated learning outcomes are essential to the effective design of instructional and learning systems. The learning outcome approach is a perspective and a mode of thinking in developing valid sequences of learning activities. Educational models, from the other side, make easily understood the relationships between learning outcomes and the learner's progress, as the latter moves along the 'novice-to-expert' continuum. These sequences can be described as hierarchical taxonomies –a method to simulate the way human knowledge is constructed. New ideas are created by integrating previous ones along with sensory input (perceptions). The dependency of newly created knowledge to the old one proves the hierarchical knowledge construction: every higher level concept is based on lower level information [16]. This is already realized in educational theory, and thus the educational models are designed as taxonomies.

Generally, *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 [17].

An initial effort that set the basis of learning taxonomy creation presented in [17], where a methodology for writing learning outcomes was described. In the same research, Bloom identified a classification of the 3 levels of thinking that takes place during the learning process: cognitive, affective and psychomotor. Within each of these levels Bloom recognized an ascending order of complexity. Nowadays, his taxonomy is worldwide accepted. It is used as a framework upon which one could build prior knowledge and, consequently, develop more complex levels of understanding [13].

In recent years, attempts were made to revise Bloom' Taxonomy [18] with some, apparently minor, but actually significant, modifications. At the same time, several taxonomies, such as **S**tructure of the **O**bserved **L**earning **O**utcome (SOLO) taxonomy, were also proposed. SOLO 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 learning period [18] and Critical Thinking framework [19], [20].

Nowadays a great amount of learning is delivered through LMSs. LMS are a wide range of systems that organize and provide access to online learning services for students, teachers, and administrators. They are capable to offer a wide-range set of services, which, chiefly, include access control, communication tools, and organization of user groups. More advanced ones provide services for learning content provision, administration and management of the learning processes / courses / learning resources. It is therefore precise that these systems are a means to keep up the cognitive-constructivist approach and to electronically support educational techniques [21]. However, they do not offer tools that create learning outcomes based on hierarchical structures [1].

A first effort on creating this kind of service was firstly initiated in [5], [6]. 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 faces several limitations, e.g. in handle dynamic content, multiple learning taxonomies, content in multiple languages etc.

III. An Intelligent Learning Outcomes Authoring Tool (LOAT)

Realizing these shortcomings, we designed and developed an intelligent solution for the LMSs that lead to gain competitive advantages in constructing educational scenarios or sequences of learning activities. Our effort concluded in a cognitive tool, the **Learning Outcomes Authoring Tool** (**LOAT**).

LOAT is designed and implemented as an exemplary paradigm, using mature programming techniques and emergent web technologies. It is a web application that provides a number of features in handling hierarchies. In our proposal, the hierarchies represent cognitive models, and their content specifies learning theories. The navigation between the nodes of the hierarchies aid the educator to understand, and, furthermore, to select the appropriate learning outcome, according to the cognitive level s/he wants to achieve (ie, cognitive, affective or psychomotor [7]).

Since the initial provision of this tool was 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 both in the interface (known as i18n –internationalization) and in the content itself. This is achieved by using the 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 this proposal a dual capacity is 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 LOAT comes along with a detailed extension of content of the pedagogical theories.

A. The architecture of LOAT

Both parts of LOAT implemented as 3-tier client-server architecture (Figure 1).

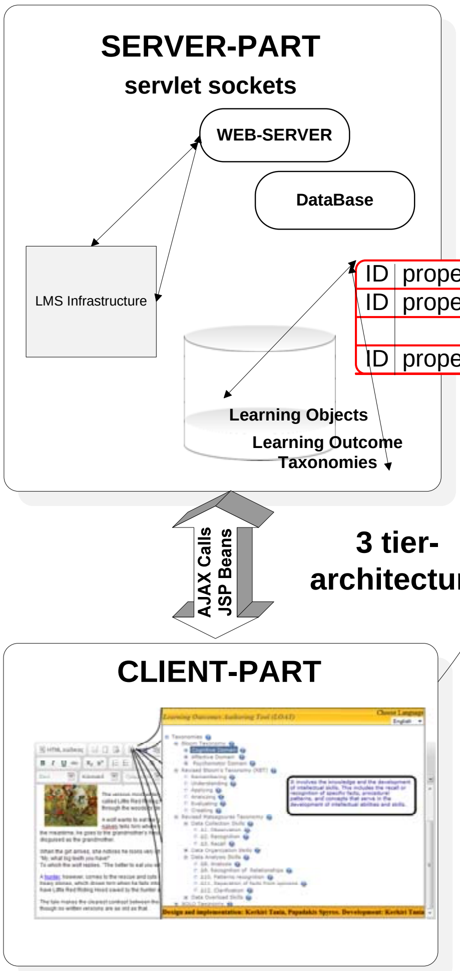


Figure 1. The architecture of the proposed system. The usage of the client-side tool is also depicted.

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, controlling the data flow [22]. The communication between the two parts is made through javabeans (JSP technology) and AJAX-calls [23].

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.

The main principles that lead the design and implementation details of this tool are [24]:

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. The database is built using InnoDb tables, which further contribute to the database consistency. 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, the latter is aided by these pedagogical data while constructing their lessons.

3) To further facilitate the development and to make LOAT 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 based on css classes. 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 [25]. 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.

4) Finally, confidential information of the application, such as database passwords, are kept in protected external files, simplifying, in the same time, the configuration of the application.

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

B. The functionality of LOAT

The *administrator's* interface is divided into two parts, shown in Figure 2.

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*

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.

Case 3: Finally, a third option is available: 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.

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.

Let's mention that, in any of the above three cases, LOAT creates the same hierarchy for every language of the system, (even if only the English description for the data of a node is provided). This means that each node of taxonomy has the same Id in every language. Consequently, this creates a template for any educational taxonomy regardless language. While LOAT is used it gives the same identification for the same learning outcome of any learning resource.

4) *Edit a node*

Edit: the detailed data of a node, along with their multilingual 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 are 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*

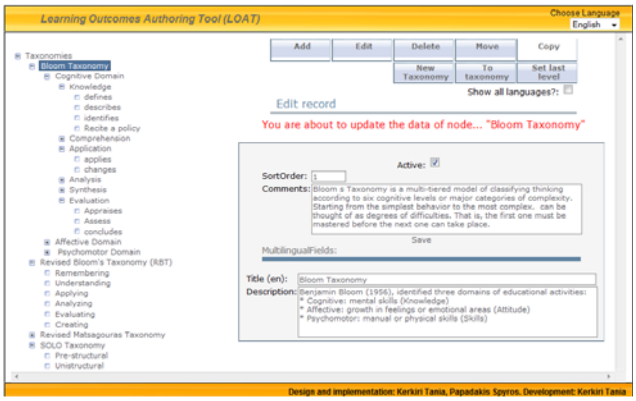
Move: moves a node in another branch, along with its subsequent nodes –still preserving the hierarchy of the branches. Cycles are not allowed. Consequently, it is not allowed to move a node under a sub-node.

7) *Convert a node to a new txaonomy*

Convert node to taxonomy: converts a branch 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. **a)** 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, **b)** 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, **c)** in a meta-level application, each node that has been defined as final, will be treated as a leaf.



Edit record

You are about to update the data of node... "Bloom Taxonomy"

Active: ☑

SortOrder: 1

Comments: Bloom s Taxonomy is a multi-tiered model of classifying thinking according to six cognitive levels or major categories of complexity. Starting from the simplest behavior to the most complex. can be thought of as degrees of difficulties. That is, the first one must be mastered before the next one can take place.

MultilingualFields:

Title (en): Bloom Taxonomy

Description: Benjamin Bloom (1956), identified three domains of educational activities:
* Cognitive: mental skills (Knowledge)
* Affective: growth in feelings or emotional areas (Attitude)
* Psychomotor: manual or physical skills (Skills)

Figure 2. The LOAT administrator environment

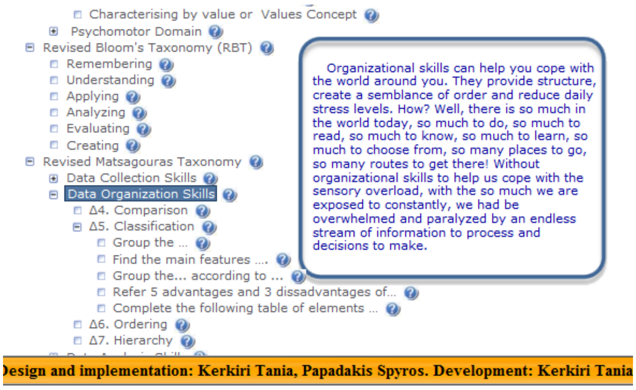
In Figure (2) a screenshot of the administration interface of LOAT is presented. LOAT today, it is available as a standalone application. As it can be seen in this figure, fields only in English language are presented –due to the fact that English was selected as the default language of the system.

Moreover, *a meta-level view* of LOAT data was implemented highlighting its actual capabilities. This part is addressed to the end course-creator. Its capabilities are actually emerged while the educators create their lessons, as follows: while the educator navigates over the nodes of the taxonomies, a server-call is executed on 'mouse blur' event –

which is attached on the exclamation mark of each node–triggering an Ajax-call. While this call executed, a window pop-ups presenting the cognitive theory described by the underlined node. More details of the selected learning theory are shown while the tree expands. In the last level, the educator finds out which learning outcome to use so that to fulfill the proper objective. The interface of this part of LOAT is shown in Figure 1.

Let's see an example of how this really works. Imagine an educator that uses LOAT to create his/her lesson. This educator navigates through the taxonomies, already provided as part of our effort. As depicted in Figure (3), the pop-up window clearly shows the theoretical frame of the cognitive theory it represents. 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**'.

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, LOAT writes the selected text into the editor.


□ Characterising by value or Values Concept
⊞ Psychomotor Domain
⊟ Revised Bloom's Taxonomy (RBT)
□ Remembering
□ Understanding
□ Applying
□ Analyzing
□ Evaluating
□ Creating
⊟ Revised Matsagouras Taxonomy
⊞ Data Collection Skills
⊟ Data Organization Skills
□ Δ4. Comparison
⊟ Δ5. Classification
□ Group the ...
□ Find the main features ....
□ Group the... according to ...
□ Refer 5 advantages and 3 dissadvantages of...
□ Complete the following table of elements ...
□ Δ6. Ordering
□ Δ7. Hierarchy

Organizational skills can help you cope with the world around you. They provide structure, create a semblance of order and reduce daily stress levels. How? Well, there is so much in the world today, so much to do, so much to read, so much to know, so much to learn, so much to choose from, so many places to go, so many routes to get there! Without organizational skills to help us cope with the sensory overload, with the so much we are exposed to constantly, we had be overwhelmed and paralyzed by an endless stream of information to process and decisions to make.

esign and implementation: Kerkiri Tania, Papadakis Spyros. Development: Kerkiri Tania

Figure 3. LOAT in action

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

C. *New perspectives*

The administrative part of LOAT is already available as a standalone application, and it soon will be available in various LMSs. This part allows the collaboration of educators in providing its content. A second view of it is already available as a plugin in popular html WYGIWIG editors (see Figure 1), such as FCK, which are successfully embedded in a variety of web-applications –including LMSs.

A future functionality is also designed. This new capability is implemented based on technologies emerging from Web 2.0. To support this functionality a new structure is designed, containing metadata. 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, **c)** to be used in searching for learning resources that fulfill the specific learning outcomes.

LOAT is implemented so that to undertake all the effort needed to properly maintain this metadata. Our final goal is to provide a web application that shifts to an emergent web paradigm, exploiting the value of people's participation in the implementation of common learning objectives in the form of learning outcome.

IV. Conclusion

For each module / lesson, educators need to write learning outcomes that match the module or lesson content, teaching strategies and assessment. Learning outcomes specify the intended endpoint of the engagement to specified learning activities. The greater the synergy between learning outcomes, teaching strategies and assessment techniques, the more successful the learning process is likely to be.

In this article, a learning outcomes editor, the Learning Outcomes Authoring Tool, is presented. This is a web-tool driven from cognitive theories and based on emergent web-technologies aiming to facilitate the learning outcomes creation. This tool soon will be available as an add-on on various popular LMSs, such as Moodle, LAMS, etc. Due to LOAT is designed to be an add-on in LMSs, it could be a bridge among them. Our effort was oriented so that LOAT to become a real assistant for the learning community. Technology does not teach students, effective educators do. Our goal is to support the educator to be an effective Learning Designer. As educators get involved in authoring learning activities, it is imperative (for ultimate success of all participants) that educators share examples and stems of well-designed learning outcomes that meet demanding behavioral criteria.

Consequently, LOAT 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

[1] T. Kerkiri, and A.-M. Paleologou, "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.

[2] J. Psotka, and S. A. Mutter. "Intelligent Tutoring Systems: Lessons Learned". Lawrence Erlbaum Associates. ISBN 0805801928, 1988.

[3] P. Brusilovsky, "Adaptive and Intelligent Technologies for Web-based". Proc. Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, 4, 1999, pp. 19-25 Doi: http://www2.sis.pitt.edu/~peterb/papers/KI-review.html.

[4] G. Conole, and M. Dyke, "What are the affordances of Information and Communication Technologies", ALT-J, 12.2, 2004, pp. 113-124.

[5] S. Papadakis, and E. Giglione, "Enhancing critical thinking by providing cognitive skill-based question wizards in LAMS activities". Proc. 2008 European LAMS Conference: Practical Benefits of Learning Design, June 2008. doi:

http://lams2008.lamsfoundation.org/pdfs/Papadakis-Ghiglione_EurLAMS2008%20_FINAL_.pdf.

[6] S. Papadakis, and E. Giglione, “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.

[7] L. Anderson, and D. Krathwohl, “A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives”, New York: Longman, 2001.

[8] R. T. White, and R. M. Gagné, “Past and future research on learning hierarchies” Educational Psychologist 11.1, (1974). 2010

[9] D. Kennedy, “Writing and using learning outcomes: a practical guide”. Cork: Quality Promotion Unit, University College Cork, 2006.

[10] G. Conole, and K. Fill, “A learning design toolkit to create pedagogically effective learning activities”. Journal of Interactive Media in Education, 2005, vol. 1.

[11] B. Kizlik, “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.

[12] S. Adam, “Using Learning Outcomes. A consideration of the nature, role, application and implications for European education of employing ‘learning outcomes’ at the local, national and international levels”, unpublished. Report presented at the 2004 United Kingdom Bologna Seminar, Heriot-Watt University, Edinburgh, Scotland, 2004.

[13] M. Kordaki, S. Papadakis, and T. Hadzilacos, “Learning Design: the views of Prospective Computer Professionals”. Proc. Informatics Education Europe II Conference., Nov, 2007, pp. 2-11.

[14] G. Babiuk, “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.

[15] J. Moon, “The Module and Programme Development Handbook”. London: Kogan Page Limited, 2002.

[16] M. K. Sein , R. P. Bostrom, and L. Olfman, “Rethinking end-user training strategy: applying a hierarchical knowledge-level model”, Journal of End User Computing, vol. 11(1), 1999, p.32-39.

[17] D. S. Bloom, M. D. Engelhart, E. J. Furst, W. Hill, and D. Krathwohl, “Taxonomy of educational objectives”, vol. 1, New York:McKay, 1956.

[18] J. Biggs, and K. Collis, “Evaluating the Quality of Learning: the SOLO taxonomy”, New York: Academic Press. 1982.

[19] E. Matsagouras, “Strategies of teaching”. Athens: Gutenberg, 1998.

[20] E. Matsagouras, “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.

[21] T. Kerkiri, A.-M. Paleologou, D. Konetas, and K. Chatzinikolaou, “A learning style – driven architecture build on open source LMS’s infrastructure for creation of psycho-pedagogically – ‘savvy’ personalized learning paths”, (Ed). Safeeullah Soomro, E-learning experiences and future, ISBN: 978-953-307-092-6, Publisher: INTECH ISBN 978-953-7619-23-7, 2010.

[22] D. K. Fields, M. A. Kolb, and B. Shawn, “Web Development with JavaServer Pages”, 2nd Edition, ISBN: 193011012X, 2001.

[23] J. Musser, Ajax Web2.0 Principles And Best Practices, Published by O'Reilly media, Inc 1005, Gravenstein Highway North Sebastopol, CA 95472, 2007.

[24] A. Shuen, Web 2.0: A Strategy Guide: Business thinking and strategies behind successful Web 2.0 implementations, Ed. Simon St. Lauren, Published by O'Reilly media, Inc 1005 Gravenstein Highway North Sebastopol, CA 95472, 2008.

[25] A. Maroudas, C. Katsanos, and N. Avouris, “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».