

Octopus: A Collaborative Environment Supporting the Development of Effective Instructional Design

Manolis Mylonakis, Polyxeni Arapi, Nektarios
Moumoutzis, Stavros Christodoulakis
Laboratory of Distributed Multimedia Information Systems
and Applications (TUC/MUSIC),
Technical University of Crete
73100 Chania, Greece
{manolis, xenia, nektar, stavros}@ced.tuc.gr

Maria Ampartzaki
Department of Preschool Education
University of Crete
Rethymnon, Crete, Greece
mabarjaki@edc.uoc.gr

Abstract— Current technology does not meet the needs and expectations of instructional designers and teachers to collaborate on creating and using effective instructional designs to keep learners motivated and maximize the learning outcome. On the one hand, the IMS Learning Design standard is very complex for instructional designers and teachers to understand and use as a common language. On the other hand current tools, especially those close to the standard are not highly usable. This paper presents Octopus, an environment supporting collaboration between instructional designers and educators for the development of educational templates and scenarios that can be used and reused in different educational contexts. Octopus pays special attention in the users' needs while it stays compatible with standards. Thus, usability studies have been performed from the earliest development steps, while a number of workshops have been organized with the participation of pedagogy experts and teachers. A significant number of educational templates and scenarios have been developed with Octopus as well, within ongoing EU projects.

Keywords—Instructional design; instructional design tools; IMS Learning Design, pedagogical patterns

I. INTRODUCTION

Instructional design is an engineering activity focusing on “what” and “how” to teach learners (with different needs and characteristics) to keep them motivated and maximize the learning outcome. It employs theories, models, methodologies and tools for instruction and applies strategies and techniques derived from them to address specific instructional challenges [3]. Its output is some instructional product to help a learner acquire some knowledge or skill [14]. Consequently, instructional design essentially translates principles of learning and instruction into plans for instructional materials, activities, information resources and evaluation [28] in a way that promotes transfer of sound practices and models in diverse content areas [26].

Instructional designers and teachers share a common goal: making instruction appealing, effective and maximizing the learning outcome. Towards this goal their roles are complementary and integral: Instructional designers are expected to provide generic instructional designs, while teachers deliver instruction after selecting and contextualizing a specific instructional plan to meet the needs of a particular

learner group. However, this type of collaboration is not observed in practice. As a result, teachers are not aware of the sound instructional alternatives and, in many cases, try to play the role of instructional designer themselves leading to poor instructional designs, where a sound pedagogical approach (the “how” to teach) is absent and the focus is given on “what” to teach. This results in ineffective encounters, inefficient activities, and finally unmotivated learners.

Technology could function as a common language between instructional designers and teachers providing effective communication between them as well. Standards could provide a common conceptual model and representation for instructional designs and ensure interoperability and reusability of educational templates and scenarios in diverse systems and contexts. The dominant standard for instructional design is IMS Learning Design (IMS LD) [13]. IMS LD has been criticized many times due to its complexity [11][24][27]. Tools that expose this complexity to the user (e.g. the RELOAD editor [19], CopperAuthor [29] and ReCourse [10]) remain untapped. These tools focus on supporting the standard and put user needs on the background. Supporting standards is definitely important and necessary. However, users (in this case teachers and instructional designers) little care about them. They want tools that will make their lives easier and improve the way they work.

Consequently, appropriate applications exploiting IMS LD while, at the same time, hiding its complexity, are needed to enable instructional designers and teachers use a common language for collaboration, development and sharing of educational templates and scenarios that can be used, adapted and further reused in new educational contexts. To be successful, such tools should put user needs in the foreground. To do so, usability studies can be employed from the beginning of their development.

In this paper we present Octopus, a collaborative environment allowing instructional designers and educators to work together towards their common goal; creating effective instruction. Octopus pays special attention in the user needs and the instructional design process itself, while it stays compatible with standards. Thus, usability studies have been performed from the earliest development steps, while a number of hands-on workshops have been organized with the participation of pedagogy experts and teachers. Improvements

of the tool have been made after the continuous feedback of users in a number of tool releases.

The structure of this paper is as follows: Section II presents related tools for instructional design. Section III presents Octopus and how it supports the process of instructional design. Section IV presents Octopus architecture. Section V focuses on the use and evaluation of Octopus. Finally, Section VI presents the conclusions and future directions of this work.

II. RELATED WORK

Griffith et al. [11] classify learning design tools along two axes: 1) Close to specification (low-level) - distant from specification (high-level) and 2) General purpose - specific purpose. The first axis has to do with how close the presentation terms and structures of the tool are to the specification, while the second is related with how generic or specialized a tool is [24]. Generic tools give users access to the entire specification, whereas specialized cater for a specific pedagogy.

MOT+ LD Editor [23], LAMS [4] and ASK LDT [27] are high level, generic tools. MOT+ LD Editor is a stand-alone graphical authoring tool, which enables users to graphically design learning designs and learning design templates based on the interconnection of user defined learning activities. However, it requires further understanding of knowledge based concepts before it can be used. LAMS is a web-based graphical authoring tool, which enables users to graphically design LDs and LD templates based on a linear sequence of pre-defined learning activities. MOT+ LD Editor and LAMS underlying model are not IMS LD compliant, but they can export LDs and LD templates to IMS LD format. ASK-LDT is a stand-alone graphical authoring tool, which enables users to graphically design LDs or LD templates based on the interconnection of user defined learning activities characterized with terms from the "DialogPlus Taxonomy of Learning Activities" [9]. It can export LDs and LD templates in IMS LD Level A, B format. However, ASK-LDT mostly uses concepts of IMS LD in its GUI and, along with the complex graphical notation used to classify the different types of activities, makes the learning design process complicated.

"Low level" tools include RELOAD Editor [19], CopperAuthor [29] and ReCourse [10]. RELOAD Editor supports IMS LD Levels A, B and C. However, it is not suitable for instructional designers and educators since it requires the designer to know every detail of the IMS LD elements and concepts to use it. CopperAuthor is a form-based editor supporting IMS LD Level A. As with RELOAD, it is extremely difficult to use it since it requires detailed understanding of the IMS LD specification. ReCourse is a stand-alone authoring tool supporting IMS LD that combines form-based and graphical-based authoring of learning designs and learning design templates based on the interconnection of user defined learning activities. Although its user interfaces are more intuitive than those of RELOAD and CopperAuthor editors, it is still difficult for teachers to use it, especially using its graphical-based interface. ReCourse can import and export LDs and LD templates in IMS LD compatible format.

Open Graphical Learning Modeler (OpenGLM) [15] is a tool distant from the specification (high-level) and specific-purpose, which supports IMS LD levels A and B. The tool was conceived to facilitate non-IMS LD experts in creating, sharing and reusing units of learning. To achieve that, OpenGLM adopts a visual modeling metaphor that conceals the complex and unintuitive elements and structures of IMS LD from the user interface. Although more intuitive than the other visual modeling tools, the results of the evaluation performed in [15] show that the teachers still have problems with transferring concepts from their teaching environment to the concepts of IMS LD and difficulties to assemble a learning design using the visual modeling interface of the tool.

As noted in [8] useful tools for instructional designers and teachers seem to be distant from the specification, in the sense that "real" users should not understand IMS LD and that their designs are just compiled into IMS LD.

Octopus supports IMS LD Level A, while hiding its complexity from its graphical user interfaces. In this sense it is closer to OpenGLM. As ASK-LDT [27], MOT+ and LAMS, it supports the development of educational templates. In addition to all tools mentioned here, Octopus further supports: 1) collaboration between instructional designers and teachers; 2) editing and sharing of educational templates and scenarios including reuse of their parts through drag-and-drop; and 3) connection with external repositories and environments exploiting available designs but also sharing educational practices and scenarios developed within the system with the wider educational community, using their APIs and implementing protocols such as OAI-PMH and OpenSearch. Octopus' focus is to support the user needs, while staying compatible with standards. Thus, the users' requirements were taken into account from the design phase and extensive usability studies have been performed from the earliest development steps, as well as a number of workshops were organized with the participation of pedagogy experts and teachers resulting in a number of tool releases.

III. OCTOPUS AND THE INSTRUCTIONAL DESIGN PROCESS

Action psychology states that human behavior is directed towards plan-based accomplishment of goals. Plans are hierarchically arranged. Background knowledge and the environment interact in plan creation and execution. Considering goal priorities and dependencies when deciding what to learn and how to coordinate multiple learning strategies improves the effectiveness of learning in a system with multiple goals. Learning strategies, represented as methods for achieving learning goals, can be chained, composed, and optimized, resulting in dynamic learning plans that are pursued in a flexible manner [25].

Octopus is a collaborative environment facilitating instructional designers to develop learning strategies in the form of pedagogical patterns (or educational templates), as prescriptions for designing instructional products to optimize the learning outcome. Pedagogical Patterns capture best practice in particular educational domains, assisting teachers to outline the strategies to surpass common difficulties and problems, such as how to motivate students, how to introduce

new concepts or how to sequence activities [7]. The teachers, based on these patterns, can create a number of educational scenarios for various educational contexts. There is an obvious relation with architectural design patterns, as described by Alexander [1], where “a pattern language gives each person who uses it, the power to create an infinite variety of new and unique buildings, just as his ordinary language gives him the power to create an infinite variety of sentences”. Fig. 1 presents an educational template for the problem-based learning approach, while Fig. 2 presents an educational scenario developed using this template.

Sharing of educational templates and scenarios is supported in workspaces, as well as reusability of educational scenarios and parts of them (e.g. activities, activities structures) in other scenarios with an easy way from within the Octopus. This adds a social dimension in the process of instructional design. Instructional designers and educators can easily find and use learning activities or structures of them residing in templates or scenarios developed by other users.

Octopus pays special attention in the users’ needs, while staying compatible with standards. Octopus supports IMS LD Level A since it seems to be the most understandable and suitable for instructional designers and teachers, if they know the correspondence of its concepts with the concepts that they are familiar with [11][24][27].

IMS LD Level A models activities, roles and environments. Activities can be grouped into activities structures and executed into specific environments. An environment is formed by learning objects and services provided to users during activity execution. Users are classified into roles (learners, teachers, tutors, etc.). Octopus presentation model hides IMS LD complexity from its graphical user interfaces and, where needed, it uses more intuitive concepts in place of the IMS LD ones. The decisions made were based on the feedback from instructional designers and teachers from the earliest development steps and later in a number of usability studies and workshops (Section V). Fig. 3 shows the Octopus presentation model compared to the model of IMS LD Level A. The concepts used in the GUI are presented with bold line. The names of IMS LD concepts that have been replaced are presented with dashed line. The concepts presented with dash line are hidden from the GUI.

Octopus allows metadata description of educational templates/scenarios and their integral parts (e.g. activities, resources) using Application Profiles of LOM metadata, supporting this way the needs of different communities [20]. Connection with external repositories and environments is also supported, both for finding and reusing available resources, but also for sharing educational templates and scenarios developed in Octopus with the wider educational community.

The process of the creation and editing of educational templates and scenarios is presented in Fig. 4. This can be divided into the following three phases: (a) Identifying an educational template/scenario to edit, (b) Editing an educational template/scenario, (c) Saving/Deleting/Exporting (to IMS LD or SCORM) a template/scenario.

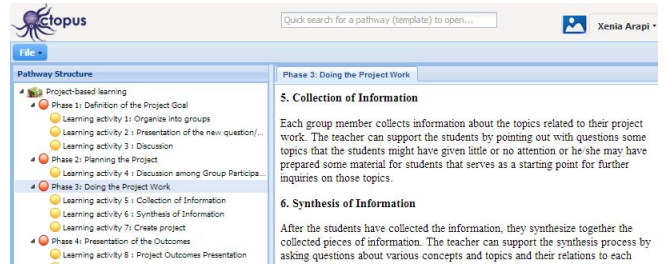


Fig. 1. Example of an educational template for project-based learning

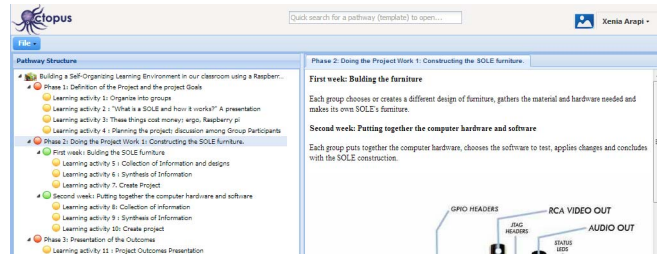


Fig. 2. An ed. scenario based on the project-based learning template

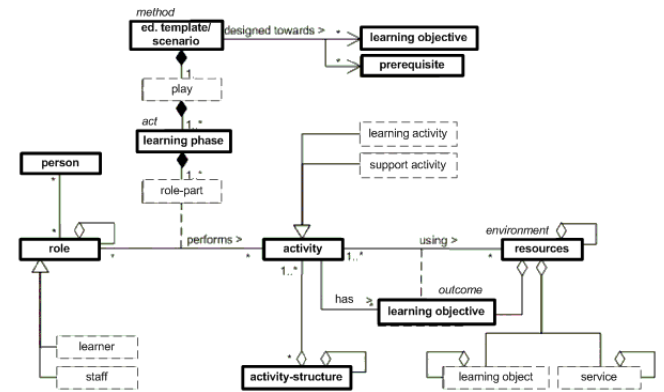


Fig. 3. Octopus presentation model vs. IMS LD Level A model

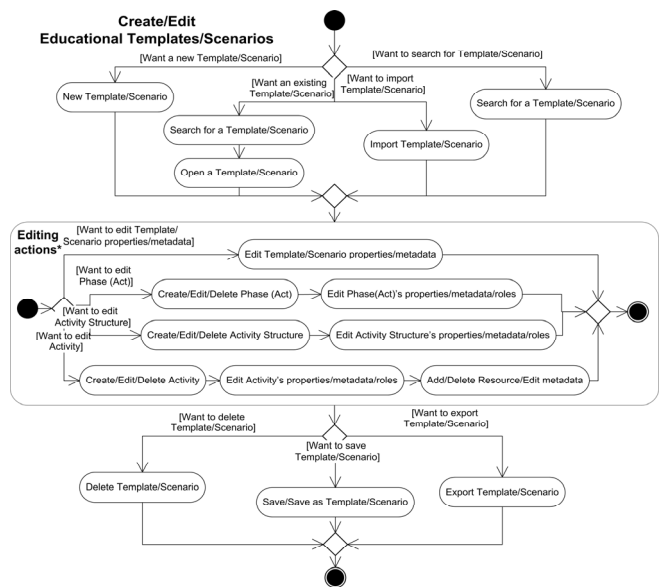


Fig. 4. Educational templates and scenarios authoring process

A. Identifying an educational template/scenario to edit

This phase may include the creation of a new educational template/scenario or search/open/import an available one. In the case of a new educational scenario it may also include search and use of an educational template for the development of the educational scenario.

B. Editing an educational template/scenario

The educational template/scenario development phase includes the following steps:

1) Defining the scope and learning objectives of the educational template/scenario according to the current educational context. Defining any prerequisites. Editing the properties/metadata of the educational template/scenario including those values.

2) Dividing the upper learning objective into sub-objectives and define a number of learning phases and activities needed to achieve these objectives. Structuring and describing those activities (e.g. insert new nodes, delete nodes or change their position, edit description). Search and reuse of learning activities or structures of them from other templates/scenarios developed and shared by other users is supported with a simple drag and drop function (Fig. 5). Editing the properties/metadata of learning phases and activities. Assigning roles to phases, activities.

3) Associating resources to activities and editing their metadata. Search, retrieval and use of resources residing in external repositories, such as Europeana, Natural Europe, SlideShare, YouTube etc. is also supported exploiting their APIs (Fig. 6).

C. Saving/Deleting/Exporting (to IMS LD or SCORM) an educational template/scenario

This phase may include saving, deleting or exporting an educational template or scenario for transfer and use in other environments. The system supports exporting of educational templates/scenarios to IMS LD Level A or SCORM format.

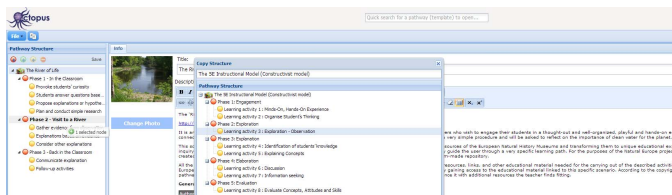


Fig. 5. Finding and reusing activities/activity structures

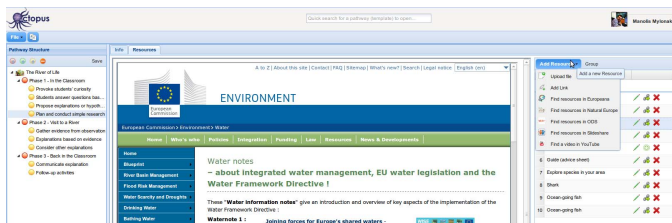


Fig. 6. Adding resources from local disk or the web. Search/retrieval and use of resources residing in external repositories such as Europeana, Natural Europe, SlideShare, YouTube etc. is also supported exploiting their APIs

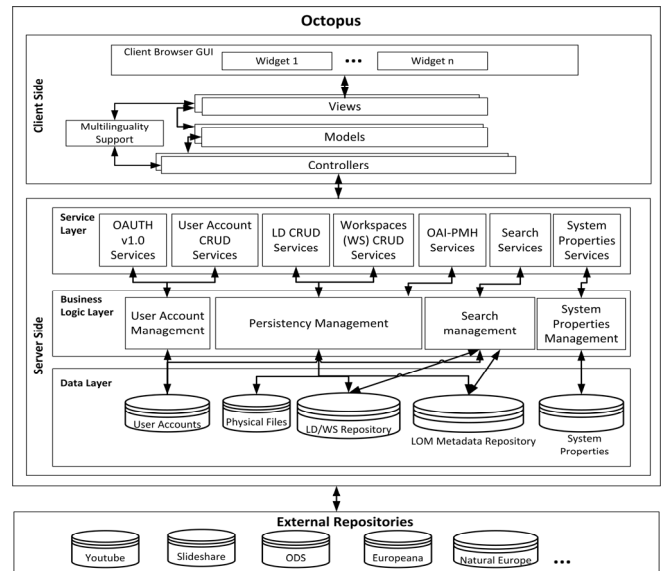


Fig. 7. Octopus Architecture

IV. OCTOPUS ARCHITECTURE

Octopus follows a client-server architecture that is presented in Fig. 7. The components of this architecture are described in the following sections.

A. Client Side

The client side component follows a MVC-like pattern with models, views and controllers, undertaking the communication between the client side as well as the server side component. *Models* are collections of data that are used and presented by the views which could be panels, trees, grids, forms, buttons and windows etc. A *View* is made with HTML, CSS, JavaScript and templates. Combinations of different views are used to produce different *Widgets* to be delivered to the end user through the browser. The *Controller* updates the views when the model changes, adds event listeners to the views and updates the models when the user manipulates the views. The data interchange between the client and the server side component is made through asynchronous AJAX calls and JSON messages. The *Multilinguality Support* module handles the translation of the user interface elements.

The widgets of the client side utilize a common web browser in order to present their functionality: (a) The *User Accounts Management* widget, responsible for the management of the user accounts, (b) the *Educational Templates/Scenarios Finder* widget, for the search of educational templates, scenarios and workspaces, (c) the *Metadata Management* widget allows description of templates / scenarios with App. Profiles of LOM metadata, supporting this way the needs of different communities, (d) the *Educational Template/Scenarios Authoring* widget, allowing the authoring of educational templates and scenarios, (e) the *Workspaces Management* widget, for the administration of users' workspaces, educational scenarios and templates, (f) the *Multilinguality Support Widget* and the management of information related to the *system configuration*.

B. Server side

The server side consists of three layers: the service layer, the business logic layer and the data layer. The *Service Layer* controls the communication between the client and the business logic layer of the server by exposing a set of services to the client-side widgets. The services provided are: (a) *OAuth services* to authorize a foreign web application's requests for access to a user's data [18], (b) *User account CRUD services* (Create, Retrieve, Update, Delete) providing the appropriate services for the management of user accounts, (c) *LD (Educational Templates/Scenarios) CRUD services* for the management of educational templates and scenarios and their metadata, (d) *Workspaces (WS) CRUD services* for the management of workspaces and their content, (e) *OAI-PMH services* allowing the dissemination of educational templates/scenarios metadata to external repositories/federations implementing all methods required by OAI-PMH V2.0 [17], (f) *Search services* allowing search for users, workspaces, educational templates or scenarios based on the submitted information through the corresponding client-server module and matching with the corresponding metadata. *OpenSearch services* are also provided allowing external tools and environments to search for templates/scenarios, and (g) *System Properties services* for system properties' management based on information submitted through the corresponding client side component along with the related business logic.

This *Business Logic Layer* enforces the business rules on the data. It implements the necessary functionality in order to support the services in the service layer, making possible the use of the data in the data layer and consists of the following components: (a) The *User Account Management* component implementing the rules for the creation, retrieval, update and deletion of users' accounts, as well as their authentication and authorization. (b) The *Educational Templates/Scenarios Authoring and Persistency Management* component, responsible for the implementation of the corresponding CRUD services. It is also responsible for the implementation of the rules supporting the OAI-PMH services to disseminate the templates and scenarios metadata. (c) The *Search Management* component serving the search services utilizing the related information in the data layer. (d) The *System Properties Management* component for the management of system properties and the multilinguality support properties.

The *Data Layer* is responsible for the persistent data storage related with user accounts, workspaces, templates/scenarios, metadata, system configurations and physical files. The system repository adopts the OAI Reference Model [12] and accommodates modules for the ingestion, archival, indexing, and accessing of templates/scenarios, metadata, workspaces, users, content etc. It is able to ingest Submission Information Packages (SIPs) transferred by the Octopus authoring environment, preserve them as Archival Information Packages (AIPs) and disseminate them as Dissemination Information Packages (DIPs) both to the Octopus authoring environment by exposing a set of services to it, and to external repositories/environments by providing an OAI-PMH and an OpenSearch interface, as well as IMS LD/SCORM export mechanisms.

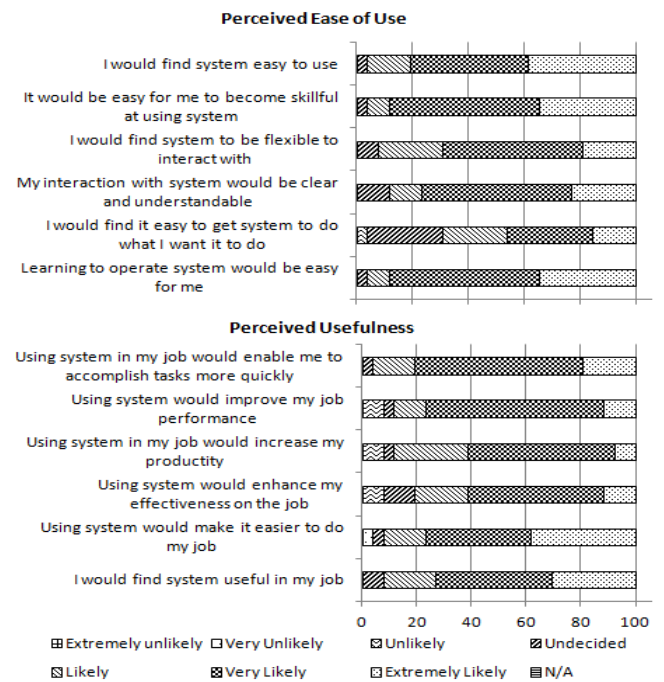


Fig. 8. Perceived Usefulness and Perceived Ease of Use results

V. DEPLOYMENT, USE AND EVALUATION

Octopus has been used in ODS [22] and Natural Europe [21] EU projects. Although only six months have passed since its initial deployment, more than 1200 educational scenarios have been already developed by pedagogy experts and educators in the context of these projects, while many more are planned to be developed in the next years.

Four pluralistic usability walkthrough sessions with 15 participants each were organized, comprised of pedagogy experts, educators, system designers, and usability experts. The users were asked (a) to create an educational template and (b) use it to develop an educational scenario. These two high level goals were analyzed in a number of sub-goals associated with particular screens. After the analysis of these sessions, 13 user interface issues were detected and fixed. Moreover, some interesting ideas and recommendations came up from the users and were implemented in the system. Furthermore, a heuristic evaluation was performed in the context of the graduate HCI course of the Electronic and Computer Engineering Dept. of the Technical University of Crete. In this course, the students had to perform usability evaluation on several products including Octopus. The evaluation was based on Jakob Nielsen's heuristics [15]; 72 errors (10 major) were detected and fixed. Finally, a think-aloud evaluation was performed with 8 educators/teachers. 8 problems were detected and fixed.

Apart from the above activities directly focusing on identifying usability issues, several sessions were organized with actual users to measure the perceived usefulness and ease of use of Octopus. Three workshops/pilot schools were organized in the context of ODS, TRANSIT [29] projects, while others are planned for the next years. In the ODS Opening Day Conference organized in Greece/Athens, 60

secondary education teachers used Octopus to develop learning scenarios from specific educational templates and share them through the ODS portal. 25 persons participated in the TRANSIT Pilot School organized in Greece/Panormo and 22 scenarios were developed with Octopus and published in ODS portal. Moreover, Octopus has been used in the context of the “Pre-school education – teaching practices” course of the Department of Preschool Education of the University of Crete. The teacher of the course used Octopus to develop a template that was given to the students (50 in total) to develop their teaching scenarios. The scenarios were enriched with resources including power point presentations, sound recordings and photos taken during the implementation of the scenarios in kindergarten. After their experience with the tool, the participants completed a Perceived Usefulness and Perceived Ease of Use Questionnaire [5].

Fig. 8 presents the results (%) of the Perceived Usefulness and Perceived Ease of Use Questionnaire that were promising: Most of the pedagogy experts/educators found Octopus easy to use and useful for their jobs.

VI. CONCLUSION AND FUTURE WORK

Exploiting usability engineering approaches, as in the case of Octopus, can address the current shortcomings of instructional design tools and enable effective collaboration and support for instructional designers and teachers while, at the same time, exploit the dominant standard for instructional design, namely IMS LD. Our experience supports the findings of Derntl et al. [6] that “conceptual complexity does not impede effective IMS LD authoring, so the barriers to adoption appear to lie elsewhere”. Effective user interface metaphors can hide the most difficult parts of the IMS LD model and raise the perceived ease of use and usefulness of tools. Consequently, the main contribution of our work is that usability engineering principles and methodologies are important to ensure alignment of instructional design tools with users’ needs and expectations without sacrificing compliance with standards.

Following the initial usability studies and hands-on workshops for educators, the next steps to be taken include using Octopus: 1) to enable sharing of adult learning scenarios developed within the context of the ALICE project [2]; 2) to support development and monitoring of project-based scenarios in secondary education in Greece within the context of the “Research Project” obligatory course; and (3) to enable the establishment and sustenance of ODS [22] teachers’ communities of practice for resource-based eLearning within which best practices are documented and shared as educational templates in Octopus to facilitate development of contextualized educational scenarios.

ACKNOWLEDGMENT

This work is partially funded in the scope of the Natural Europe project (FP7-ICT-PSP: 250579), the Open Discovery Space project (FP7-ICT-PSP: 297229), and the ALICE project (518106-LLP-1-2011-1-IT-GRUNDTVIG-GMP).

REFERENCES

- [1] C. Alexander, “The timeless way of building”, New York: Oxford University Press, 1979.
- [2] ALICE project, <http://www.alice-llp.eu/>
- [3] J. Bourdeau, and R. Mizoguchi, “Collaborative Ontological Engineering of Instructional Design”. Proc. of the 6th Intl. Conference on Intelligent Tutoring Systems, 2002.
- [4] J. Dalziel, “Implementing learning design: The learning activity management system (LAMS)”, 2003.
- [5] F. D. Davis, “Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology”. MIS Quarterly, 13:3, 1989.
- [6] M. Derntl, S. Neumann, D. Griffiths, and P. Oberhuemer, “The Conceptual Structure of IMS Learning Design Does Not Impede Its Use for Authoring”. IEEE TLT, no.99, pp.1-14, 2011.
- [7] J. Eckstein, M.L. Manns, H. Sharp, and M. Sipos, “Teaching from different perspectives”. Proc. of the 8th European Conference on Pattern Languages of Programs, Irsee, 2003.
- [8] Edutech Wiki. s.l. <http://goo.gl/Hw30ee>.
- [9] I. Falconer, G. Conole, A. Jeffery & P. Douglas., “Learning Activity Reference Model – Pedagogy”, 2006.
- [10] D.Griffiths, P. Beauvoira, O. Libera & M. Barrett-Baxendaleb, “From Reload to ReCourse: learning from IMS Learning Design Implementations”, 2009.
- [11] D. Griffiths, J. Blat, R. Garcia, H. Vogten, and K.L. Kwong, “Learning Design Tools”. s.l. : In R. Koper & C. Tattersall (Eds.), Learning Design, A Handbook on Modelling, 2005.
- [12] ISO 14721:2012 Open Archival Information System (OAIS) Reference Model. Accessed 23/8/2013 from <http://goo.gl/PMh53x>
- [13] IMS Learning Design Specification Version 1.0. [Online] Available at: <http://goo.gl/4vzF1R>, last visited 10 April 2005.
- [14] M. D. Merrill, “Components of instruction toward a theoretical tool for instructional design”. Instructional Science, 29(4-5), 2001.
- [15] J. Nielsen, “Heuristic evaluation”. In Usability Inspection Methods, John Wiley & Sons, New York, NY, 1994
- [16] S. Neumann S. and P. Oberhuemer, “User Evaluation of a Graphical Modeling Tool for IMS Learning Design”, 2009.
- [17] Open Archives Initiative (OAI-PMH v2.0). Open Archives Initiative. [Online] 2008. <http://www.openarchives.org/>.
- [18] OAuth, <http://oauth.net/>
- [19] B. Olivier, “Learning Design Update”, 2005.
- [20] M. Mylonakis, P. Arapi, N. Pappas, Moumoutzis, S. Christodoulakis, “Metadata Management and Sharing in Multimedia Open Learning Environment (MOLE)”, Proc. of the MTSR2011 Conf., October 2011.
- [21] Natural Europe project, <http://www.natural-europe.eu/>
- [22] ODS project, <http://www.opendiscovery-space.eu/>
- [23] G. Paquette, O. Marino, K. Lundgren-Cayrol, M. Léonard, “Principled construction and reuse of learning”, 2008.
- [24] D. Ragbir and P. Mohan, “Design of an IMS LD Editor using non-IMS LD concepts”. Proc. of the ICALT 2009 Conference, 2009.
- [25] A. Ram A. & D. Leake, “Goal-Driven Learning”, MIT Press, 1995.
- [26] P. L. Rogers (Ed.), “Designing Instruction for Technology-enhanced Learning”, Hershey, PA: Idea Group Publishing, 2002.
- [27] D. Sampson, P. Karampiperis, and P. Zervas, “Developing Web-Based Learning Scenarios Using the IMS Learning Design”. Proc. of the WISE’05 Conference, 2005.
- [28] P.L. Smith, and Ragan T.J., “Instructional Design”, Second Edition, Wiley, John Wiley & Sons, Inc.
- [29] TRANSIT project, <http://www.transit-project.eu/>
- [30] W. Van der Vegt, “CopperAuthor”, vol. 2005. heerlen: Open University of the Netherlands, 2005.