COLearn: Supporting Collaborative Learning on top of Existing Learning Infrastructures

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

***Abstract*—COLearn is a web-based collaborative learning environment that supports the specification of collaborative learning workflows, their deployment and enactment. It addresses major challenges in modern learning infrastructures to enable contextualization, social constructivism and knowledge-pull. It offers a shell, easily integrated on top of existing open learning infrastructures (such as LMSs and OER repositories) to enrich their capabilities by offering functionality to design rich collaborative learning activities. COLearn run-time environment is used to enact these activities and workflows. This way, COLearn leverages the power of the underlying infrastructures, provides structure to the groups of learners that participate in learning workflows, dynamically adapts the workflows during their enactment, monitors their evolution to facilitate assessment and provides feedback to the learners. COLearn employs an intuitive graphical representation exploiting the Business Process Modeling Notation standard. As an internal representation and interoperability model it uses IMS Learning Design thus offering effective sharing and remixing of learning designs.**

***Index Terms*—Collaborative work, Distance learning, Social Computing, Workflow management software**

I. Introduction

Learning Management Systems (LMS) have been proven successful in the administration, documentation, tracking, reporting and delivery of learning contents of different granularities (learning resources, learning objects and courses). Despite the unquestionable impact of LMSs on the evolution of e-learning and education in general, there is room for improvement with respect to several aspects. COLearn addresses several issues related to the desired improvements on:

Contextualization: A traditional LMS follows a static approach to learning by offering uniform courses to learners. Our approach, in contrast, takes into account the design decisions of educators allowing them to organize learning materials, tools and learners within meaningful contexts (learning activities) that are structured within wider designed learning spaces (learning scenarios).COLearn provides a graphical authoring tool with which educators (acting as educational designers) can describe these learning spaces (learning scenarios) in the form of collaboration scripts[13].Each learning space can be reused or remixed, thus offering to educators the possibility to share their pedagogical strategies with other educators.

Social Constructivism: LMSs restrict opportunities for collaboration in learning and for the promotion of social constructivism. In our approach learners could be engaged in a distributed environment consisting of a network of people, services and resources. To provide such an environment COLearn platform at the run-time level incorporates specific tools for the support of group management and social interaction among participants, tools such as chat, micro-blogging, calendar etc. These tools are employing the Extensible Messaging and Presence Protocol (XMPP)[14]implementing the publish - subscribe communication pattern.

Knowledge-pull: Traditional LMSs adopt a knowledge-push framework. In our approach, however, we permit the educator to adopt a knowledge pull model. Educators could create their customized learning environments -at the authoring level- in the form of organized learning activities and -at the run-time level- those learning activities will guide the participants to pull knowledge that meets their particular needs from a wide range of knowledge sources. If the educator wishes to enable learners become creators of content, it is possible to incorporate content creation tools as external learning tools. In order to cope with the integration of external learning tools in a transparent way, we employ the IMS Learning Tool Interoperability (IMS LTI) [9]specification.

COLearnis designed to be easily integrated with LMSs in order to address the aforementioned challenges and offers a technological solution that enriches the underlying infrastructure and exploit its power. New services are offered (a) to educators to embrace a culture of open educational practices; and (b) to learners to engage in meaningful learning within the context of rich learning activities supported by digital resources and tools.

The rest of this paper is organized as follows: Section 2 presents the related work. Section 3presents the COLearnarchitecture and implementation. Section 4 presents, as a case study, how COLearn has been aligned with the Open Discovery Space portal to enable collaborative workflows on top of a teachers' social network and a repository of Open Educational Resources. Section 5 concludes and presents future plans.

II. Comparison with Related work

The most known platform that combines authoring and run-time aspects for learning scenarios is LAMS [10].LAMS is a complete platform for designing, managing and delivering online collaborative learning activities. LAMS is not intended to complement an existing learning environment as it is the case for COLearn. Activities in LAMS can include a range of individual tasks and group work. LAMS functionality is extended through the integration of external tools. However, to integrate an external tool one should cope with a complex LAMS Tool contract[1]. Despite the fact that activities can be assigned to groups of participants, LAMS lacks synchronous group support. LAMS provides some synchronous communication tools during the learning flow but these tools are initiated as activities as long as the author explicitly defines their use in the scenario. Also, these tools are not constantly active during the scenario execution. A teamwork environment should account the absence of physical interaction by providing synchronous collaboration among participants[11].

SLeD[12] and Reload Player[4]are run-time environments that enable the enactment of IMS LD scenarios. They both lack group coordination and collaboration. In addition, there are no communication tools available during scenario enactment. In contrast, the COLearnrun-time environment supports group-based learning. The learning scenarios are performed by group of users in a collaborative and synchronized manner. To enable synchronization and personalization of the learning process, the CopperCore engine is integrated to the COLearn platform. Furthermore, the adoption of the XMPP real-time protocol offers to the COLearn platform a messaging system to notify group members about the changes in the progress of the scenario in real-time. XMPP-based tools are provided by the COLearn platform to support synchronous and asynchronous collaboration/ communication. Note that XMPP can provide asynchronous communication as well. Messages addressed to absent users are delivered to them when they enter the platform. External learning tools are integrated to the COLearn platform in a transparent way by implementing the IMS LTI specification.

III. COLEARN ARCHITECTURE

COLearn consists of two main components as depicted in figure 1: a) the authoring tool that provides the means to specify the learning workflows and b) the run-time environment where theses workflows are enacted.

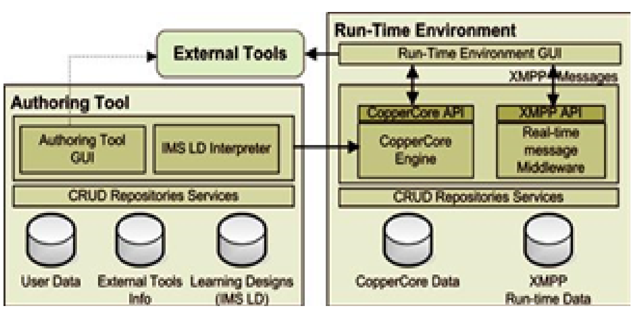

Fig. 1. *COLearn Architecture*

A. *Authoring Tool*

In order to develop and offer collaborative learning facilities on top of an open learning infrastructure one has to cope with learning content, communication/ collaboration facilities, and run-time execution (i.e. learning process). Collaboration scripts can specify the components as well as the orchestration of the learning process. COLearn offers an authoring tool to describe collaboration scripts as learning workflows, based on the Business Process Modeling Notation(BPMN)[3]. BPMN is the model for the graphical representation of the learning workflows. The COLearn authoring tool, as seen from the user, is presented in Fig. 2.

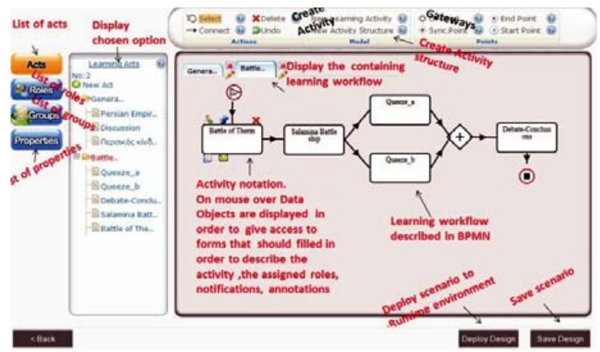

Fig. 2. COLearn Authoring Tool

Learning workflows are parsed and transformed to IMS Learning Design Level C[8]. Details regarding the internals of the Authoring Tool have been given in Stylianakis et al.[6] along with the rationale for differentiating[5] the user interface representation model from the internal representation model and their mapping.

B. *Deployment of collaboration scripts*

The deployment of a collaboration script refers to the identification of the participants and their roles for enacting it. This is a two-step process: Initially, the IMS LD document is published to the CopperCore engine. Then, the appropriate communication channels among the members of each group are established. The latter, is based on the XMPP protocol and made feasible by integrating the Openfire Server into COLearn run-time environment. The corresponding services and strategy to deploy collaboration scripts which describe learning scenarios, is presented in Stylianakis et al[6].

C. *Run-time Environment*

Enactment is a term that refers to the actual implementation of a collaboration script through the COLearnrun-time environment after its deployment, i.e. after the assignment of scenario-specific roles to particular users that will participate in the scenario implementation. Consequently, to enact a script it is necessary to deploy it first. Note that each scenario could be deployed many times. Each distinct deployment corresponds to unique implementation of the scenario with specific participants. The COLearnrun-time Environment consists of two main components: a) The CopperCore engine[7] and b) the Real-Time Message Middleware. These components are described in the next paragraphs. Figure 5 presents the COLearnrun-time environment and displays its functionality.

Fig.3. COLearn run-time environment

a) Copper Core Engine

CopperCore engine provides coordination support for learning processes. Checking, synchronization and personalized view by each participant is supported by the CopperCore engine. The engine fully supports IMS LD levels A, B, C and provides the persistence storage and the APIs to manage the administration and delivery of IMS LD. In addition COLearnrun-time environment incorporates the appropriate graphical components to exploit the CopperCore functionality and deliver the IMS LD learning scenarios.

b) Real-time Message Middleware / Social Interactions

COLearn incorporates Openfire server, which is a real-time collaboration server that implements the XMPP protocol. XMPP is an open technology for implementing real-time request response services. XML messages can be transferred from one entity to another in real-time. Additionally it provides the establishment of peer-to-peer media sessions supporting interactions such as voice/video chat or file transfer.

COLearn adopts the following strategy for supporting collaboration: When an author deploys a collaboration script to the run-time environment, a contact list is created for each member of each specific working group. The contact list is essentially, an instant messaging "friend list" providing the option for one-to-one messaging among members of the same group. Additionally, for each group a communication channel implemented as publish-subscribe node is established. Each member of the group is subscribed to this node. The XMPP publish-subscribe extension provides a framework to implement notification, multi-party messaging and file transfer in the COLearn platform. In addition, it offers an extensible approach to integrate tools compliant to instant messaging mechanisms for the organization/ management of working groups as well as the social interaction among group members. Finally, the message middleware interacts with the CopperCore engine, providing information and notifications among the users and triggering actions in the learning workflow process.

c) Using IMS-LTI to integrate external learning tools

COlearn integrates external learning tools by implementing the IMS LTI specification. In COLearn run-time environment, each external tool is launched as a service employing IMS-LTI. In the COLearn authoring tool, an author can choose to use an external tool from a list of available ones. This list is maintained/ updated by the administrator of the system. The administrator provides the URL of the external tool, a secret and a key that are needed in order to establish the connection. The secret and the key parameters are needed because the connection of the COLearn platform with the external tools (at the run-time level) is using the Open Authentication protocol. When the author selects the desired external tool, she additionally submits the IMS-LTI specific attributes described in the specification. At the run-time environment, when an activity which defines external tool usage is activated, then the tool is automatically launched.

IV. Alignment with Open Discovery Space Portal

Following the presentation of the overall architecture for enabling learning scenarios on top of existing open learning infrastructures, this section presents a case study that exemplifies the details of the approach. This is the case of the Open Discovery Space (ODS) community portal targeting teachers in European countries. The portal provides a common access point to many national and international repositories of open educational resources through its federated repository. The main functionalities offered through the alignment of the ODS portal with COLearn are the following:

Automating the navigation between the ODS portal and COLearn to support ODS communities' members in creating, sharing, finding and enacting learning scenarios by the usage of the COLearn authoring tool and run-time environment.

Exploiting ODS harvested resources to enable their inclusion in learning activities managed by COLearn. This is done through a special pane implemented in COLearn that provides search functionality on top of the SOLR search engine.

Assign ODS community/group members with COLearn scenario-specific roles to enable deployment of collaborative learning designs (scenarios or lesson plans in the ODS terminology) using information from ODS portal about its users.

To make the description of these functionalities more concrete, let us present them within the context of five specific use cases that reflect the actual user needs addressed by the aligned ODS-COLearn environment:

1. Creating a new scenario or lesson plan

The creation of a learning scenario is triggered from the ODS portal interface where an ODS user is able to see a list of existing scenarios and create new ones. This list of scenarios is provided through a web service offered by COLearn. When the user selects to create a scenario, the COLearn authoring tool is triggered and the user automatically logs in. This underlying authentication mechanisms is based on OAuth[2].

2. Editing/Cloning an existing scenario or lesson plan

Learning scenarios can be further edited and copied across

ODS communities. To do so, the ODS user can select a specific learning scenario from the list of available scenarios within a specific community and trigger the COLearn authoring tool. Then, within the authoring tool, the user is able to edit the scenario. It is also possible to clone it to make a copy of it to another ODS community that the user participates. Special services are employed for this process that provide information about the ODS communities where the user participates.

3. Searching the ODS repository

During the editing of a learning scenario in the COLearning authoring tool, the user is able to trigger a special pane that exploits the SOLR search engine service of the ODS federated repository to present a list of qualifying resources based on a faceted-search approach. Finally, the user can select the resources that are appropriate for inclusion in a certain learning activity. Resource links are then retrieved from the ODS repository and stored as references within the learning activity elements' metadata.

4. Deploying a scenario or lesson plan

Deploying a scenario or lesson plan means to identify the participants that will take part in an enactment of the scenario or lesson plan and assign roles to each of them so that the enactment can take place in the CoLearnrun-time environment. The deployment process in the aligned CoLearn environment exploits information from the ODS portal regarding the members of the community/group within which the scenario to be deployed is available.

5. Enactment of a scenario of lesson plan

Enactment is a term that refers to the actual implementation of a scenario or lesson plan through the CoLearn runtime environment after its deployment, i.e. after the assignment of specific scenario-specific roles to a set of users that will participate in the scenario implementation. Consequently, to enact a scenario or lesson plan it is necessary to deploy it first. Note that each scenario could be deployed many times corresponding to different implementations with different users participating each time.

V. CONCLUSIONS ANDFUTURE WORK

We presented COLearn, a platform acting on top of an existing open learning infrastructure to enrich its capabilities. An infrastructure (e.g. an LMS) that uses COLearn, could extend its functionality by offering to teachers the ability to design rich learning activities and learning workflows, enact them, and provide structure to the groups of learners that participate in these workflows, dynamically adapt the workflows during their enactment and monitor their evolution to facilitate assessment. The COLearn authoring tool produces collaboration scripts employing the BPMN model for their graphical representation. These scripts are internally represented using the IMS LD specification. The run-time environment consists of the CopperCore engine, which supervises the execution of the design. The group management, social interaction and collaboration among the participants are made feasible by integrating the Openfire server which implements the XMPP real-time protocol. For the integration of external learning tools, COLearn implements the IMS LTI specification. The COLearn platform is an open platform that can collaborate with different types of internal/external learning applications. As a case study we presented the integration of COLearn with the Open Discovery Space (ODS) community portal. The alignment is targeting to support ODS communities' members in creating, sharing, finding and enacting learning scenarios, to exploit ODS harvested resources

Our future plans in enhancing COLearn services focus on supporting mobile learning scenarios including support for augmented reality games. To enable such scenarios it is necessary to model interactions with tools running on mobile devices and incorporating parameters for the specification of physical contexts on which learning activities take place. The learning domains that will be facilitated with these extensions include field trips for science projects and physical inquiries related to history, geography and language learning.

ACKNOWLEDGMENT

This work is partially funded in the scope of the Open Discovery Space project - http://portal.opendiscoveryspace.eu (FP7-ICT-PSP: 297229).

REFERENCES

[1] C. Alario-Hoyos, M.L. Bote-Lorenzo, E. Gómez-Sánchez, J. I. Asensio-Pérez, G. Vega-Gorgojo, and A. Ruiz-Calleja, "GLUE!: An Architecture for the Integration of External Tools in Virtual Learning Environments." Computers & Education (2012).

[2] Barry Leiba, "OAuth Web Authorization Protocol", IEEE Internet Computing, vol.16, no. 1, pp. 74-77, January/February 2012, doi:10.1109/MIC.2012.11

[3] BPMI.org, OMG: Business Process Modeling Notation Specification. Final Adopted Specification. Object Management Group, 2006, available at: http://www.bpmn.org

[4] D. Griffiths, P. Beauvoir, O. Liber, and M. Barrett Baxendale, "From Reload to ReCourse: learning from IMS Learning Design Implementations." Distance Education, 30, no. 2 (2009): 201-222.

[5] D. Sampson, P. Karampiperis and P. Zervas, "ASK-LDT: A Web-Based Learning Scenarios Authoring Environment based on IMS Learning Design", International Journal on Advanced Technology for Learning (ATL) (ISSN 1710-2251), vol. 2(4), pp. 207-215, ACTA Press, 2005

[6] G. Stylianakis, P. Arapi, N. Moumoutzis,S. Christodoulakis, CoLearn: Real time collaborative learning environment. International Conference on e-Learning and e-Technologies in Education (ICEEE) (2013)

[7] H. Vogten, H. Martens, P. Rosmalen, and E. J. R. Koper, CopperCore. Retrieved January14, 2005, from SourceForge: http://coppercore.org

[8] IMS Learning Design Specification Version 1.0. [Online] Available at:. http://www.imsglobal.org/learningdesign/index.cfm IMS

[9] IMS Learning Tools Interoperability Version 1[Online] Available at http://www.imsglobal.org/lti/blti/bltiv1p0/ltiBLTIimgv1p0.html

[10] J. Dalziel, "Implementing learning design: The learning activity management system (LAMS)." LAMS Teacher's Guide, 2006, V2.0.

[11] M. N. K. Boulos, A. D. Taylor, and A. Breton, "A synchronous communication experiment within an online distance learning program: a case study." Telemedicine Journal & e-Health 11 no. 5(2005)

[12] OUUK. Sled player (2005) (cited April 8th, 2007)

[13] P. Dillenbourg, "Over-Scripting CSCL: The risks of blending collaborative learning with instructional design." Three Worlds of CSCL. Can We Support CSCL? (2002): 61-91.

[14] XMPP Standards Foundation at http://xmpp.org/

#