Introduction

A unit of learning is an abstract term used to refer to any delimited piece of education or training, such as a course, a module, a lesson, etc [6]. When these units of learning are computationally represented following the IMS LD specification they are called a full ready-to-run Unit of Learning (UoL) that can be authored, interpreted and run in different tools. The IMS LD is an educational modeling language used to develop applications in educational contexts [6, 8]. In the TENCompetence project, this specification has been adopted as the facto standard.

One of the commitments of the project is the adoption of the IMS LD specification in diverse and real educational contexts. Nowadays, different tools conform with IMS LD for supporting the edition [2, 4], the instantiation [3] and runtime [1] have been developed. Some of these tools have been successfully tested in different learning contexts [5]. However, in a lifelong learning context these tools show some limitations. A pilot study carried out in the Association of Participants Agora [9], there were identified a set of problems regarding the management of UoLs. To execute and perform a UoL it is necessary to define the list of students in the course, the roles that each student is going to play and the course that they participate. For that, it is necessary to create in the production time what is called a run. A run is an instance of a course that serves as a bridge towards the students and a UoL [12]. Only those students registered in a run are allowed to execute (or run) the UoL. Normally, the creation of a run, and the assignation of the students to it is done by an administrator or a teacher that decides who is participating in which UoL. But in a lifelong learning context each learner organizes his/her own training and decides in which course he/she wants to participate. The figure of a teacher disappears and the learners should be able to create their own runs and manage his/her courses. The experience of the pilot showed that the current tooling is not enough for supporting the self-management of UoLs in Lifelong learning contexts and new solutions are needed. Different solutions addressing similar problems have been developed in other areas, e.g. in identity roll-on and roll-off processes in a company or automatic subscriptions to online course with Moodle Platform [10]. Although these solutions are enough for covering the necessities in their particular areas, they are not enough for solving the requirements in a scenario involving auto-management of UoLs a Lifelong Learning context. Moreover, it does not exist tooling compliant with IMS LD supporting the user in the management of their courses.

The aim of this study is to understand the needs of the UoL’s management in Lifelong Learning contexts. More concretely, we analyze the use of LD Runtime System in the context of Agora and propose a tool called LD admin or Link Tool as a solution to support the auto-management of UoLs in such contexts.

This paper is structured as follows. Section II describes and analyzes the solutions adopted in the 1st Agora pilot for solving the management of UoLs. From the results of the analysis, the main requirements regarding the auto-management of UoLs in this context and the feasible solutions proposed to solve them are presented. Section III presents the details of the Link Tool as the first approach for covering these necessities and presents the results of its usage in a 2nd pilot in the Agora Association. Finally, section IV states the main outcomes from the paper and the future work.
II. REQUIREMENTS FOR SUPPORTING THE SELF-MANAGEMENT OF UoLS IN LIFELONG LEARNING CONTEXTS

To understand the requirements arising from the management of UoLS in a Lifelong learning context, we analyze a realistic experience of a pilot carried out in the Association of Participants Ágora. The main aim of the pilot was to implement, test and investigate the benefits of the TENCompetence infrastructure and its support for the participants’ competence development [9]. The participants in the pilot used the Personal Development Plan (PDP), the PDP tool developed in the TENCompetence project, as the central tool for the creation of their own personal development plans and the performance of the activities. Some of the activities were UoLS codified in IMS LD that run in the TENCompetence LD Runtime System [1]. The LD runtime system is compliant with the IMS Learning Design specification [7] and facilitates the provision of structured activities (similar to courses) that learners can follow as part of their competence development. It provides an administrator view for uploading UoLS, registering users and creating runs and a player view that allows the learner to perform the activity. In this section we describe the use of the LD Runtime’s functionalities for managing UoLS in this context and report and analyze the solutions adopted to deal with its limitations. The results from the analysis lead to a set of requirements needed for the development of a new tool able to support the whole process in such type of contexts.

A. Administrating UoLS in the 1st Ágora Pilot

All the UoLS in the pilot were created with the LD Editor Recourse [2] by two experts. All were designed including collaborative widgets (chats in most of the cases) and with a unique learner role (UoLS can be designed including different roles with different privileges when performing the course). The administrator of the system (an IMS LD expert) used the LD Runtime administration section for uploading the UoLS to a server and registering the users participating in the experience. The administrator generated one instance for each course by creating a unique run for each of the UoLS uploaded and enrolled all the participants to these runs for associating the participants registered to the courses available. All these steps were done before starting the course.

When the pilot started, the learners accessed to the UoLS from an activity in the PDP in which there was a link to the LD Runtime and the title of the Unit of Learning corresponding to the activity. When clicking on the link, they were redirected to the LD Runtime (via an Internet browser), log in to the system with the credentials facilitated by the administrator and visualized the list of runs associated to the UoLs available in which they were previously registered. By clicking on the run, they could access to the LD Runtime players and run the UoL to perform the activity.

From the experience of the pilot, there were detected some problems and limitations of the LD Runtime system not only regarding the administration processes before the course but also in supporting unexpected situations once started. In the next lines, we describe the main problems and the solutions adopted. Figure 1 depicts the full workflow of the whole process.

- Registering the users. The LD Runtime does not include any registration module for allowing the users register the system on their own. For the pilot, the registration process was carried out by the administrator. Before the pilot started, all the users enrolled in the course were previously registered to the system and informed about their credentials via email. However, during the pilot some students that were not registered to the course joint it once it had started. As a solution, the administrator maintained a list of users with their respective users and passwords that was updated when a new user joint the course.
- Repeating a course. One of the main problems that appeared during the pilot occurred when some students wanted to repeat one of the courses that they had already performed. Since the instance of a course is related to a unique run, a user can perform the UoL only one time and all the actions are stored. Once the activity is finished, if the learner access again to the same run, they find it finished. For the pilot, this need was solved via email. When the users wanted to repeat a UoL they sent a message to the administrator and he/she created a new run of the same UoL only for that user. The administrator maintained a list with the runs created for each of the users and inform them when the run was ready.
- Identifying the correct UoL. The user linked the LD Runtime player from a particular activity in the PDP. Once in the player, they were showed a list of runs. But, from the list, the learner should choose only that run indicated in the activity of the PDP. The users had some difficulties in distinguishing the run they had to select. For facilitating this process, all the runs were named with the title of the UoL and the link in the PDP activities were edited exactly with the same name. In that way, the users could better identify the correct run. For avoiding conflicts with the name of the runs associated to the same UoL (see the case of repeating a course in the bullet above) all the runs were described with the number of the run that the user was going to play (e.g. “Practice Vocabulary-1” for the first run and “Practice Vocabulary-2” for the second run of the same UoL titled “Practice Vocabulary”).
During the course:

1. Upload the UoLs to the system
2. Create an account for each of the students that belong to the TIC course
3. Create a run for every UoL and associate it to the User.
4. Update the table of users and runs

Are all users registered?

Before the course starts:

Create the user’s accounts

Sends an email to the user informing that he/she can repeat the UoL using the X run

Figure 1 Flow diagram of the whole process of following a course, from the PDP to the LD Runtime player, and the roles involved in the process.
B. Analysis of the requirements

If we examine the whole process followed in the Agora Pilot, we observe that all the solutions adopted for managing the UoLs lead on the Administrator. However, in a Lifelong learning context the learners should be able to register to the system and create their own runs of those UoLs they are interested in. Therefore, it is necessary to provide the learner with administration tools. From a detailed analysis of the problems detected in the Agora pilot, we expose in Table I the limitations of the LD Runtime tool and propose a set of requirements as a basis for developing a tool for the auto-management of UoLs.

<table>
<thead>
<tr>
<th>Limitations of the LD Runtime System</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>REGISTRATION PROCESS</strong></td>
<td>R1. Provide the system with an automatic User Registration/Authentication Module</td>
</tr>
<tr>
<td>The LD Runtime does not provide any registration module for allowing the user to create an account for accessing the system. An administrator is needed for creating the user accounts.</td>
<td>R2. Provide the system with functionality for the learner to upload his/her own UoLs.</td>
</tr>
<tr>
<td>R3. Add the possibility of deleting a UoL. For avoiding problems with user privileges this functionality should be restricted to the owner of the UoL.</td>
<td></td>
</tr>
<tr>
<td><strong>UPLOAD UOLS</strong></td>
<td>R4. The learner should be able to create their own runs for a particular UoL and to assign a starting date for accessing the course.</td>
</tr>
<tr>
<td>The LD Runtime only allows the administrator to upload UoLs.</td>
<td>R5. The learner should have the possibility of accessing to an already created run. This run can be previously created by a teacher/administrator/expert or another user of the system.</td>
</tr>
<tr>
<td>R6. The learner should be able to delete his/her own runs.</td>
<td></td>
</tr>
<tr>
<td><strong>MANAGING RUNS</strong></td>
<td></td>
</tr>
<tr>
<td>The runs for the UoLs are only managed by the administrator in the LD Runtime.</td>
<td></td>
</tr>
</tbody>
</table>

The problems in Table I arise directly from the necessities detected during the Agora Pilot and are very context-related. But, the requirements extracted are enough generic for covering the needs of other Lifelong Learning scenarios. In section IV we shed light over other necessities that could be also considered.

III. THE LINK TOOL FOR SUPPORTING THE AUTO-MANAGEMENT OF UoLs IN LIFELONG LEARNING CONTEXTS

This section presents a web tool called Link Tool as a first effort for supporting UoLs management in Lifelong Learning scenarios according to the requirements detected in the first Agora Pilot. The firsts outcomes of using the Link Tool framed in a second pilot in the same school are also presented here.

A. The Link Tool

The Link Tool is a web-based application that enables the auto-management of UoLs. According to requirement R1 in table I, the tool includes a registration module in the main page that a users access when entering for the first time. It also includes a section for login in case that the user is already registered (Figure 1 on the top left position). Once logged in, it is shown to the user a unique view that mixes some of the functionalities reserved for the administrator role in the LD Runtime environment with those reserved for the user. According to requirement R2, the user can upload his/her own UoLs (Figure 1 on the top right position). It is validated by the system and included to the list of UoLs available if it does not have errors (Figure 1 on the bottom). The user can also delete his/her own UoL if necessary (requirement R3).
The Link tool allows the user to manage his/her runs. It offers the possibility of creating a run of an existing UoL (requirement R4) or joint an existing run (requirement R5) created by other user. This last functionality is especially useful in case of UoLs including some collaborative elements such as a chat or a forum. For example, if a user creates a UoL with a forum activity and a run associated to it, other users should be able to register to this run in order to visualize the messages in the forum. Otherwise, each user will belong to his/her run and will not visualize the messages of other learners. For that cases, it has been also included the possibility of picking a date for accessing a run. This run can be only accessed in that date and, therefore, force the users to perform the course in that period. The owner of a run is able to delete it.

Figure 1 Screen shots of the link tool. Top left: Main page with the registration and login menus. Top right: Page for uploading UoLs. Bottom: Validation page once the UoL is uploaded.

The Link tool allows the user to manage his/her runs. It offers the possibility of creating a run of an existing UoL (requirement R4) or joint an existing run (requirement R5) created by other user. This last functionality is especially useful in case of UoLs including some collaborative elements such as a chat or a forum. For example, if a user creates a UoL with a forum activity and a run associated to it, other users should be able to register to this run in order to visualize the messages in the forum. Otherwise, each user will belong to his/her run and will not visualize the messages of other learners. For that cases, it has been also included the possibility of picking a date for accessing a run. This run can be only accessed in that date and, therefore, force the users to perform the course in that period. The owner of a run is able to delete it.

Figure 2 Screen shot of the page for managing runs. A run can be associated to a date using a calendar.
B. Using the Link Tool in a LifeLong Learning context

To understand how the Link Tool helps in supporting the auto-management of UoLs in a Lifelong learning context, the tool was used in a second pilot carried out in the same Association of Participants Agora [11]. The aim of this second pilot was also to investigate the benefits of the TENCompetence infrastructure in the school. The main difference between the tooling used in the 1st and the 2nd pilots was that, in the 2nd one, the tools were improved: switched to a Web version and integrated in a LifeRay portal. The users accessed to this portal directly from the Web page of the association for registering to the system. As in the 1st pilot, the PDP was the central tool for generating the personal development plans and performing the activities. Some of these activities were UoLs. In this case, the UoLs were managed using the Link Tool instead of the administrator interface of LD runtime system.

Some of the UoLs used in this pilot were recovered from the previous one and new ones were created by two experts. However, in this case, each user could freely create his account using the registration module when entering the system for the first time.

The users accessed to the UoLs using a link (a URL pointing to Link tool) included in the description of the PDP activities. For that, the two experts created a default run for each of the UoLs uploaded by the coordinator starting the same day than the pilot started and link directly to that run from the PDP. The link directly redirects the users to this generic run and started following the course. In that way, the users did not have to search the correct UoL from the list of courses available and login to the system every time. It was also a good solution for those UoLs including collaborative modules such as a forum, in which different users should be included in the same run. When a user wanted to repeat a course, he/she could access the Link tool directly with login credentials, select the UoL he/she wanted to repeat and create his/her own run. This functionality avoided the necessity of maintaining a list of runs per user and facilitated very much the administration tasks of the course coordinator.

The “linking” URL used in the PDP tool could also contain information such the specific run to be used from among the number of runs available for a UoL as well as the user role for completing the run. This ensures better control in administrating UoL runs in more complex settings.

The results from the first experience show the Link tool as a good solution for supporting the administration of UoLs in a LifeLong learning context. On one hand, it provides the learners with some administration functionalities that allow them to freely manage their own courses. During the 2nd pilot, participants did not have to wait when they wanted to repeat a UoL because they could create his/her own runs. On the other hand, it facilitates the tasks of the course coordinators that only have to take care of making the UoLs available.

IV. CONCLUSION AND FUTURE WORK

The existing IMS LD tooling fails in supporting the needs arising from the UoL’s management in Lifelong Learning contexts. This paper presents the results of analyzing these necessities by studying the LD Admin IMS LD tool in the context of the Association of Participants Agora. The results of this study are a set of requirements that have served as a basis for the development of the Link Tool. This new tool compatible with LD is an improvement of the LD Admin tool that provides the learners with some administration functionalities that allow them to manage their own courses. A first experience in the Agora school has shown the Link Tool as a good support for the auto-management of UoLs in Lifelong Learning contexts.

From the experience, there were collected some problems regarding the use of the Link tool. Most of them were related with the vocabulary used in the interface and usability aspects. Specific words such as run were not well understood by the users. In future versions of the tool, the vocabulary should be adapted to the context and the visual aspects of the interface improved.

Currently, the Link Tool is being adapted to be included as a portlet in a LifeRay portal so as to be integrated with other TENCompetence tools. A portlet is a module in a LifeRay portal that can be added or deleted according to the user needs. This new version of the Link Tool will maintain the main functionalities of the current version and improve some of the usability problems observed during the experience in Agora. For the integration, some of the modules such as the registration module are changing because all the users will register to the LifeRay portal and not to the tool itself. All this new functionalities will be tested in future experiences beyond the TENCompetence project for analyzing if the requirements covered by the Link Tool are still enough for the auto-management of UoLs in Lifelong Learning contexts.

AKNOWLEDGEMENT

This work was supported in part by the European Comission in the TENCompetence project IST-2004-02787

REFERENCES


