Flexible Tools for Online Collaborative Learning: Integration of Adaptation Patterns Functionality in the WebCollage Tool

Magnisalis Ioannis
Aristotle University
Thessaloniki
Greece
imagnisa@csd.auth.gr

Demetriadis Stavros
Aristotle University
Thessaloniki
Greece
sdmetri@csd.auth.gr

Dimitriadis Yannis
University of Valladolid
Valladolid
Spain
yannis@tel.uva.es

ABSTRACT

This paper presents the architecture and functionality of an extended and more flexible version of the WebCollage tool. WebCollage was initially developed as a teacher-supportive web-based editor to facilitate the design of structured collaborative activities, following the tenets of scripting theory. However, earlier versions of WebCollage did not cater for design flexibility, as the collaborative activity could not be adapted during runtime and provide personalized support to students on an as-needed basis. In this work we explain how adaptation functionality is integrated in the WebCollage tool, based on our previously presented and elaborated theoretical framework of adaptation patterns. An "adaptation pattern" is defined as a higher level description of a corrective intervention that an experienced teacher would implement during a session of collaborative learning, in order to respond to identified learners' needs and improve the learning conditions. This work presents how the adaptation pattern (AP) theoretical framework is operationalized and integrated the AP functionality in AP-WebCollage architecture and user interface. We discuss the technical enhancements and modifications related to this endeavor, commenting also on issues relevant to IMS-LD compatibility and the efficiency of the AP-WebCollage version as a flexible tool for the design of adaptive collaboration scripts.

Categories and Subject Descriptors

K.3.1 [**Collaborative learning**]: IMS-LD, WebCollage Tool, adaptive CSCL scripts.

General Terms

Design, Standardization, Languages.

Keywords

IMS-LD, WebCollage Tool, adaptive CSCL scripts.

1. INTRODUCTION

The design and development of adaptive systems for collaborative learning (ASCL) emerges currently as a significant issue at the crossroad of adaptive educational hypermedia and Computer-Supported Collaborative Learning (CSCL) research traditions (see, for example, [1], [2]). From our point of view, we have emphasized the need for a generalized conceptual framework of adaptive scripting, relevant to all types of collaboration scripts, as a basis for formalizing the design of flexible adaptive interventions to support group learning [3]. Research has consistently emphasized that collaborating students might fail to engage in productive learning interactions when left without teachers' support (e.g. [4]). Consequently, collaboration scripts have been proposed as a means to structure the collaborative activity and engage all students in fruitful learning interactions (e.g. [5], [6]).

Interest about scripting has launched an interest on developing tools [38] for scripting (or learning design (LD)). For example, many editors appear in CSCL community to facilitate LD design-authoring but recently some web-based script editors appeared. One such script editor is WebCollage [32].

Nevertheless, adjusting the script level of granularity and flexibility emerges as an important issue that affects the outcome of scripted collaboration ([7]). We have argued elsewhere ([3], [8], [9]) that a solution to the script flexibility issue could be the integration of adaptive characteristics to systems for scripted collaboration by means of integrating "Adaptation Patterns" (APs) to the design. An AP captures some core idea of pedagogical value on how to adapt the collaborative learning activity when specific conditions occur. Therefore, an adaptation pattern is essentially an abstraction based on teachers' key ideas regarding adaptivity and flexibility during collaborative learning. "Adaptation Patterns" follow the general approach of design or pedagogical patterns (see e.g. [39]).

In this work specific design case studies are presented (as a proof of concept), exemplifying how the key issues of the adaptation pattern approach can be: a) incorporated in the WebCollage LD-editor and b) expressed using the IMS-LD modeling language. Please note that WebCollage is based on the use of patterns and therefore the inclusion of Adaptation patterns is consistent with its design philosophy. Also, the Learning Design (LD) modeling language is discussed, identifying hitherto advantages and limitations. In section 3, four APs are implemented in WebCollage and it is demonstrated how one specific AP is expressed in terms of IRMO, and IMS-LD. In fact these elements are embedded in the core design of WebCollage as integral part of a collaboration script, thus providing flexibility during runtime. In section 4 our experience from implementing APs in WebCollage is summarized and future directions of the research work are discussed.

PCI 2013, September 19 - 21 2013, Thessaloniki, Greece.

http://dx.doi.org/10.1145/2491845.2491858

2. BACKGROUND

2.1 Adaptation Patterns & CLFPs

In the context of technology-enhanced learning, system designers have tried to systematically exploit the modeling potential of computers and develop systems that support learners through adaptive or intelligent operation. Adaptive systems are model-based systems. An adaptive educational system (AES) is a system that aims to adapt some of its key functional characteristics (for example, content presentation and/or navigation support) to the learner needs and preferences [10]). Thus an adaptive system operates differently for different learners, taking into account information accumulated in the individual or group learner models. Introducing adaptive characteristics gave birth to the strand of Adaptive Hypermedia Systems (AHS), a significant subset of which is Adaptive Educational Systems (AES) with systems like AHA [11], InterBook and WebCOBALT [2]. Respectively the strand of Intelligent Tutoring Systems (ITSs) appeared with systems like ELM-ART, KBS-Hyperbook and SQL-Tutor [10]). According to Brusilovsky and Peylo [10], ITS traditionally focused on Curriculum Sequencing, Intelligent Solution Analysis & Problem Solving Support, while AES focused strongly on Adaptive Presentation and Navigation Support.

The above approaches aim principally on helping the individual learner. Recently research efforts have focused on introducing adaptivity and intelligence in the context of CSCL bringing together AESs and ITSs on the one hand and CSCL systems on the other. Additionally, there is initial evidence that adaptation advances the learning effects of CSCL (e.g.[3]). Computationally supported adaptive and intelligent operations are increasingly integrated in the design of CSCL systems in an effort to maximize the user-tailored support provided to group learners, focusing both on improved domain learning and development of collaboration skills. In general, creating adaptive/intelligent systems for CSCL is considered to be more demanding than creating respective systems for individual study, since apart from the pedagogical aspects one must also take into account aspects related to social relations and group dynamics [12].

Working in the area of adaptive CSCL systems (see [13]) we have coined the acronym “AICLS” (Adaptive and Intelligent Collaborative Learning Support) as a general term to denote the broader research area of adaptive and/or intelligent systems that aim specifically to support the collaborative learning activity. The intervention in such a system can target either of two layers: Layer 1: Preparation of the activity (pre-task adaptation, such as group formation). This layer deals with pre-task issues. Layer 2: Support of the activity itself (in-task adaptation) providing domain knowledge-type support or peer interaction-type support [13].

From our point of view, we have emphasized so far the need for a generalized conceptual framework of adaptive scripting, relevant to all types of collaboration scripts, as a basis for formalizing the design of flexible adaptive interventions to support group learning (Demetriadis & Karakostas in [8]). Our proposed framework is based on the notion of adaptation pattern (AP). An adaptation pattern is described as a teacher’s targeted intervention during collaboration, to adjust and improve the conditions of learning in order to maximize student engagement, satisfaction and, consequently, the learning outcomes (Karakostas & Demetriadis, in [3]). An AP can focus on practicalities of collaborative learning (such as reformatting groups in case a student is missing) or on the pedagogical efficiency of the script (e.g. adjusting the details of student scripting (roles) during a collaboration phase to enact peer interactions which are more appropriate for the students). For an adaptation pattern at least three issues should be defined: (a) conditions of initiation, (b) aspects of script to be adapted, and (c) processes to be executed. An adaptation pattern essentially is the reification of key ideas regarding adaptivity and flexibility, strongly connected to anticipated situations where an appropriate strategy would be the enactment of adaptive system behavior. For example a system in a group of learners can adapt the difficulty level of a task for the advanced learner -for example, providing more demanding material and/or assigning a more demanding role to the advanced learner-, thus, making the activity more interesting for him/her. The adaptation pattern may adjust also the guidelines offered to and the role assigned to the novice partner(s) making the activity more beneficial for all learners of the group.

Rationale of Adaptation Patterns (APs) is presented extensively by Karakostas & Demetriadis in [14]. They are supporting together social and construction elements of the learning process [15] and are in general independent and abstracted from the details of a specific learning process. On the other hand applying an adaptation pattern implies using it along a specific learning process or ideally in conjunction with other patterns, such as specific “CLFP” (collaborative learning flow pattern). CLFPs are best practice learning designs, i.e. learning designs that when applied under certain circumstances may lead to a successful CSCL process [17] and examples of them are Jigsaw, Pyramid, TAPPS etc. [18], [19].

In general, a situation is envisioned where teachers would be able to select and implement the type of adaptivity they deem necessary in any demanding situation during collaboration. Of course, this generalization leads to the question of how to define what a demanding situation is and how to develop accordingly the needed adaptation patterns. We have proposed and exemplified elsewhere (Karakostas & Demetriadis, in [3]) a design methodology (DeACS) for identifying adaptation patterns to be embedded in adaptive scripting systems.

An adaptation pattern is a core idea of how to adapt the collaborative learning activity when specific conditions occur. By contrast to a design pattern (or CLFP) (which prescribes a course of action as a solution to a commonly occurring problem) an AP suggests a valuable alternative (to the whole or part of the solution) depending on conditions. We argue that introducing adaptation patterns can help reusable knowledge on common and pedagogically valuable adaptations to become part of the design process. Moreover, APs could be integrated in authoring tools, much like some script editors (e.g. WebCollage [32]) encourage editing of a whole learning design (script) composed by several design patterns (or CLFPs) in different combinations (for example, collections of adaptation techniques could become available in the form of an ‘adaptation toolbox’). Additionally, teachers and designers may become familiar with and reflect on the use of valuable adaptations during collaboration and transfer research-based conclusions on adaptation to everyday educational practice.

However, although an AP is eventually experienced as a specific adaptation of the collaborative activity, it is essentially more than that. An AP needs to be somehow modelled in order to become a reusable software component. Thus, what differentiates the adaptation pattern approach from simply introducing hardcoded possibilities for adaptation to a CSCL system is the need for modelling the patterns at a more abstract level.

Naturally, the important technical challenge is how to link the core non-adaptive pedagogical design of the script (currently supported by various non-standardized script editors) with adaptive design functionalities. Our position on this is that adaptation patterns can be built either as software add-ons or web services that are invoked by a script editor when available (i.e. the software extends its functionalities depending on the available add-ons library or list of web services). The teacher then could integrate the selected adaptation pattern at the appropriate point of the computerized script representation and parameterize the properties and methods of the pattern as desired. In this way the "adaptive logic" can reside at a separate software component (outside IMS-LD manifest) and pedagogically effective units of learning UoLs are decoupled from the flexibility it is desired to have under certain circumstances. The adaptive strategy can thus be modified without touching the original pedagogy pattern expressed with LD.

2.2 Modeling Adaptation Patterns

In our work so far, it is argued ([20], [21], [22]) that the structure of an AP can be modeled through four major components, namely: Input, Rule(s), Model(s) and Output [21] ('IRMO' specification, figure 1). Input refers to one or more parameter(s) which are monitored by the AP during runtime and trigger the enactment of the adaptation (these could be, for example, a student assessment outcome, a group deliverable, the synthesis of a group, etc.). Rule(s) implies input processing: one (or more) rules (of the form: "IF Input satisfies condition THEN the Output is ADAPTED") are applied to input. The Model part defines which (one or more) entities of the collaborative activity are necessary to be modeled in order for the AP to function properly (these entities could be learner or group characteristics, collaboration script aspects, activity phases, material, etc.). Finally, Output refers to the result produced when applying Rule to Model according to some Input. The Output could be, for instance, a change in the synthesis of the group, the material provided to individual learners, the sequence of the activity phases, the roles of the learners, etc. In general, the Output results to an updated representation (internal and/or external) of the activity.

Put briefly, the IRMO specification suggests that for constructing an AP one has to: a) define monitored parameters (e.g. from interaction analysis tools) to be used as Input, b) construct Rules, c) decide which Model characteristics, in various databases and even the manifest of an IMS-LD script, are to be affected by Rules, and d) define Output (form, content, etc.).

IRMO resembles AHA [11] architecture as one can locate a one-to-one correspondence of basic elements: a) Input of IRMO is linked to User requests of AHA, b) Model of IRMO corresponds to the user and domain model of AHA, c) Rule of IRMO is the Adaptation Model-rules of AHA and d) Output of IRMO corresponds to served output of AHA. AHA is a simple Web-based adaptive engine designed and used in individual learning environments. So far, AHA architecture has not encountered problems in CSCL learning scenarios (e.g. user-group modeling has not been necessary in AHA).

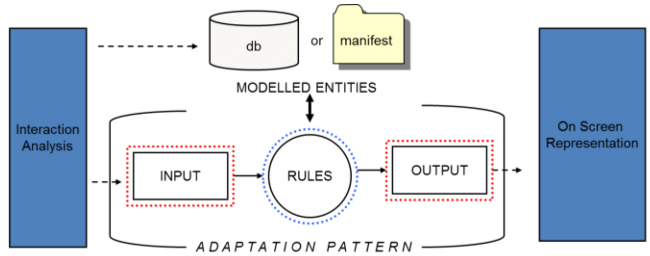

Figure 1. The IRMO specification for defining the structure of an adaptation pattern

However, if an AP is to be reusable it has to be expressed using a common modeling language 'understandable' by CSCL systems. The next step, therefore, is to explore how the IRMO modeled structure of an AP can be expressed using the IMS Learning Design specification, which is reckoned as one of the most promising efforts to aid CSCL activity design and play by a machine. Learning Design is a scientific field, while IMS-LD is an approach-specification for modeling, etc. Then, IMS-LD or other Educational Modelling Languages are based on a meta-model. An authoring tool can produce learning designs (units of learning in IMS-LD terminology) that are interpretable or not by computers. Among these, there are authoring tools that produce IMS-LD compatible representations.

IMS-LD manifest and pedagogically effective UoLs are decoupled from the flexibility it is desired to have under certain circumstances. This way the adaptive strategy is more flexible and can be adjusted without touching the original pedagogy pattern expressed with LD.

2.3 LD basics and Tools: Adaptation Capabilities, Limitations and WebCollage tool

LD is primarily a modeling tool, which uses the metaphor of a theatrical play for describing a teaching-learning process [30]. Its main components are: metadata, roles, acts, environment, role-part (i.e. activities of actor, who does what, when and how), sequence of activities within a role-part, conditions and notifications (interactivity and control over a live learning design as a form of event driven messaging system within an LD player). Through LD tool a unit of learning (UoL) is formally expressed, that is, a complete, self-contained unit of education or training, such as a course, a module, a lesson etc.

To be usable by computers, Learning Design has to be given a concrete syntax and semantics. Thus, Learning Design specification and IMS-LD standard notation ([24], [25]) are worth mentioning. IMS-LD specification consists of three levels of implementation and compliance and each level is mapped to separate XML Schemas:

(a) Learning Design Level A: contains all the core vocabulary needed to support pedagogical diversity. (b) Learning Design Level B: adds Properties and Conditions to level A, which enable personalization and more elaborate sequencing and interactions based on learner portfolios. (c) Learning Design Level C: adds Notification to level B, much like an event-driven messaging system, which provides more interactivity and control during CSCL script runtime.

The approach followed in IMS-LD specification is not to define a single large schema with a core of mandatory elements and numerous optional elements, but rather to define a complete core that is yet as simple as possible, and then to define two levels of extension that capture more sophisticated features and behaviors.

Analyzing the IMS-LD structure Burgos et al. [26] identify three levels of support that the specification can offer to various types of adaptation: (a) well supported (for learning flow, content, evaluation and interactive problem solving support), (b) partially supported (for user grouping, interface adaptation, adaptive evaluation and full modification of a course on-the-fly), and, finally, (c) no support (for dynamic modification of learning structure and method in run-time, and adaptive information filtering and retrieval).

There is a debate in CSCL community regarding IMS-LD adaptive capabilities and limitations. According to [34] IMS-LD lacks expressiveness, while according to [26] this is not fully true. On one hand, there are studies that propose totally new languages for specifying a CSCL script; for instance [37] with LDL language. On the contrary, studies are located claiming that standards can be extended or combined to achieve the desired result (e.g. [36], [35]) and that some aspects of adaptivity can be modeled and supported by today's technology (for a review see [27]).

Technically, it is clear that a number of issues should be considered when different types of adaptation need to be supported with some formalization method, such as IMS-LD. Interest about scripting has launched an interest in developing tools for scripting (or learning design (LD)). For instance, many editors exist in CSCL community to facilitate LD design-authoring -like ReCourse [30]- but recently some web-based script editors appeared. One such script editor is WebCollage [32]. Besides being flexible in usage (i.e. web-based), WebCollage also facilitates CLFP (or design pattern-DP) based script design, along with assessment activities and the ability to combine these features (e.g. combining Jigsaw and Pyramid scripts e.g. in [40]).

In the following section we discuss how the WebCollage IMS-LD editor can be enriched to a) design APs, b) to formally express the adaptive design of collaborative learning activities in IMS-LD syntax, c) run adaptive LDs in an IMS-LD compatible player.

3. INTEGRATING ADAPTATION PATTERNS TO WEBCOLLAGE

3.1 Tools choice & technicalities

Although many LD editor tools have been investigated, a triplet of them has been selected to be used for our endeavor to facilitate flexibility-adaptivity in IMS-LD scripts. Firstly, an LD player was necessary and Coppercore IMS-LD player [28] was used (along with Astro, SLeD [29] and Coppercore enhancements), because it is the one that is closest to IMS-LD standard. Moreover, ReCourse [30] was used as a proofing tool for our IMS-LD designs [25]. Moreover, Recourse supports design of basic IMS-LD elements (i.e. conditions to implement Rules part of IRMO) that provide flexibility to script design, and incorporates other standard technologies like QTI [31].

After a thorough review of IMS-LD design tools, WebCollage [32] was selected as the editor that provides capability of interconnecting CLFPs and APs. The aim is to provide evidence for APs' realization upon standardized and computerized CSCL scripts (i.e. IMS-LD based). WebCollage is web-based (in opposition with Recourse). It is based on open-source technologies like MySQL for persistence layer of data and Dojo Toolkit [33] for interface design. Moreover, the architecture of WebCollage tool is easily understood by a developer, who wishes to enhance the tool. Additionally and more critically WebCollage is based on a pattern-based design approach that is consistent with the one followed by the APs [39]. The tool has been specifically enhanced in three directions: a) Hellenization, b) IMS-LD player integration, c) AP capability assimilation in WebCollage's user interface. Many technical details are left out of the scope of the current discussion though dealt with during implementation (i.e. fixing some identified bugs of WebCollage in exporting IMS-LD packages and running them in IMS-LD players).

3.2 Hellenization and IMS-LD player integration in WebCollage

One of the goals of this work was the Hellenization of WebCollage. The existence of the Greek edition of WebCollage adds usability in WebCollage tool thus enabling the use of the tool by Greek teachers and learners. Now, WebCollage is available to be used in Greek context. Moreover, the technical prerequisites to make the tool international have been tackled with. Also, multilingual features in WebCollage help users deal with the collaborative planning scenarios within WebCollage in multilingual contexts (i.e. with students speaking different languages). Previously, it is has been noted how important are CLFPs for CSCL script design. For this reason, CLFPs have been translated into Greek language texts and additional information (summary, chart, instructions and examples) are provided for each CLFP. This helps users to choose the right CLFP and thus create more effective CSCL scenarios.

Internationalization (globalization) is the process of creating an application flexible regarding the implementation of different languages. The Dojo Toolkit –which is the basis of the user interface of WebCollage-, supports through specific components the process of internationalization.

Moreover, another valuable enhancement was incorporated in WebCollage. That is, Sled (which is an enhanced Coppercore IMS-LD player) was integrated inside WebCollage's environment (i.e. in a new tab). Thus, the designer besides creating a script and packaging it in a UoL, can now publish and run the designed script.

3.3 APs value added to WebCollage

Four APs, among others (see Karakostas & Demetriadis, [24]), were chosen for implementation with IMS-LD. These are:

AP1. "Advance the Advanced" AP. This pattern aims at offering a more challenging version of the task for the advanced learner and, also, an adapted version of the task for the novice learner exploiting the partnership with the advanced learner.

AP2. "Lack of confidence" AP. This AP expresses the simple idea that a novice learner in a specific known group needs support taking into consideration the context of the group (i.e. other learners' domain knowledge) in which learner belongs in a specific CSCL setting

AP3. "Group of Novices" AP. This pattern aims at providing extensive support to groups with domain novice partners, when the group members are domain knowledge novices (or at least most of the group members).

AP4. “Assign Moderator” AP. This AP expresses the simple idea that to assign a competent moderator to groups, one has to model and pre-require: (a) group size greater than three, (b) at least one participant in the group must have the appropriate experience to be assigned as moderator.

Due to space limitations, along with WebCollage enhancements to facilitate flexible scripting, only AP2 is discussed in more detail (the choice is made because AP2 incorporated both personal and group modeling aspects in a CSCL script). Therefore, we demonstrate upon a specific AP the process of: a) designing an AP according to IRMO design methodology, b) implementing an IRMO-modeled AP in IMS-LD terms, c) incorporating APs in WebCollage environment, and d) running the adaptive script.

3.4 An AP -WebCollage case implementation

3.4.1 AP to IRMO design

The pattern ‘lack of confidence’ expresses the simple idea that a novice learner in a specific known group needs support in the “collaborate to learn” process of a CSCL setting (especially when this learner is the sole novice learner in a group), so that novice learners maintain their interest, interactivity and participation and the activity becomes beneficial for them [39]. It can be more formally described as following:

Name: Lack of Confidence

Key-idea: Support and encourage novice learners in larger groups in order to be more confident to participate.

Activation Conditions: When one (or more) novice learner participates in a large group (more than three participants and novices are minority).

What to model: (a) learners’ domain knowledge, (b) group size and synthesis, (c) supportive material, (d) script alternative organizational aspects (i.e. student roles).

What to adapt: the support and encouragement offered to novice students. This may include: (a) providing specific to the task material to improve their contributions (e.g. helpful guidelines to better accomplish a task), (b) assigning specific roles to novices in order to make their contribution more clear and straightforward, (c) providing metacognitive support to novices to help them reflect and self-assess their own and others’ contribution.

According to the IRMO specification this AP is described as follows:

1. Input: the outcome of a prior knowledge questionnaire which is used as a measure of learners’ expertise in both domain knowledge and communication skills,

2. Rule: IF Learner is Novice (meaning that the questionnaire outcome is below a certain level) THEN provide New supportive material AND/OR assign specific roles to novices AND/OR provide metacognitive support (e.g. messages),

3. Model: Learner’s prior knowledge (Advanced/Novice) & Learning Material (Supportive) & Group size and synthesis & Roles

4. Output: provide New (supportive) material to Novice Learners AND/OR assign specific roles to novices AND/OR provide metacognitive support (e.g. messages).

3.4.2 IRMO-modeled AP to IMS-LD

First, the pyramid script was integrated in the “Lack of Confidence” AP code. The key idea in this pattern is to provide support for a novice learner in a group. This is done by modeling a) the learners’ domain prior knowledge (or in the same manner collaborative skills could be modeled) and b) average domain prior knowledge within group. Thus, Novice is an individual whose personal knowledge level of the domain (i.e. *personal_knowledge_level* property) is below the average personal knowledge level of the domain in a group of learners (i.e. *average_group_knowledge_level* property). Then a rule is defined to enact the adapted behaviour of the system (e.g. provide support against lack of confidence) for the novice learner.

Table 1. Rule of Lack of confidence AP (notice comparison between properties personal_knowledge_level and average_group_knowledge_level)

```
<imsld:if><imsld:greater-than>
<imsld:property-ref ref=" personal_knowledge_level " />
<imsld:property-ref ref=" average_group_knowledge_level" />
</imsld:greater-than>
</imsld:if><imsld:then>
<imsld:hide><imsld:support-activity-ref ref="Support_Novice-Lack_of_Confidence" /></imsld:hide>
</imsld:then><imsld:else>
<imsld:show><imsld:support-activity-ref ref="Support_Novice-Lack_of_Confidence" /></imsld:show>
</imsld:else></imsld:else>
```

Finally, the IMS-LD compliant XML code is presented, which describes the support activity triggered depending on whether the learner is “Novice or Not” (see Table 2).

Table 2. Modifying the Output of the AP

```
- <imsld:support-activity identifier=" Support_Novice-Lack_of_Confidence " isvisible="false">
<imsld:title> Support_Novice-Lack_of_Confidence </imsld:title>
<imsld:environment-ref ref="env-12" />
- <imsld:activity-description>
- <imsld:item identifier="item-798" isvisible="true" identifierref=" Support_Novice-Lack_of_Confidence ">
<imsld:title>Resource</imsld:title>
</imsld:item></imsld:activity-description>
</imsld:support-activity>
```

The way we elicit and set property personal_knowledge_level is not shown (i.e. through a questionnaire). There are two necessary properties in order to model groups in the Pyramid script DP. These properties have to be global in IMS-LD terms. Number of learners and number of groups are necessary global properties for the script and in this example have initial values of 5 and 2 respectively. Moreover, a local personal property in IMS-LD terms (i.e. *group_memebership*) is necessary in order to denote which group exactly an individual is member of. In “Lack of Confidence” AP this is assumed as already set (according to table individual belongs to group 1). The properties that classify the learner as “Novice or Not” are *personal_knowledge_level* and *average_ group_knowledge_level* which is of type Integer and Real respectively and have initial values of 10 and 4 respectively in this example case. The core of the AP is presented in Table 1 where the pattern main rule (conditions in IMS-LD terms) is implemented. Notice the comparison between the properties *personal_knowledge_level* and *average_group_knowledge_level* in order to identify the individual as Novice. Also, notice the rule implements in simple words the idea that: IF learner is Novice then show *Support_Novice-Lack_of_Confidence* support activity else hide *Support_Novice-Lack_of_Confidence* support activity.

3.4.3 AP in WebCollage

To incorporate an AP into WebCollage the architecture of the WebCollage system has been leveraged to: a) be able to integrate APs in the backend database and the user interface, b) to guide designer in utilizing a specific AP. More specifically, the designer, can now choose from a list of APs in a phase, and insert an AP after an activity (see figure 2).

Moreover, when an AP is chosen, the relevant AP dialog box (see figure 2) is depicted to designer. Dialog box has the form of IRMO, informing designer about the AP he/she is going to apply in IRMO terms (i.e. input, model, rule, and output). Moreover, it provides an application example resembling the application of a CLFP (which can be characterized in these terms as a design pattern). Each AP has its own dialog box, and can have various formats of information in IRMO parts. For instance (see figure 3) the 'Advanced the Advanced' AP has as Output part of IRMO the capability to let designer choose which activities to hide and/or to show to an Advanced student as characterized by system IMS-LD Level B properties.

Notice, that all necessary IMS-LD code (showcased in previous section) is created by extended WebCollage thus facilitating the adaptive-flexible evolution of the extracted script.

3.4.4 Running APs

SLED [42] was used to 'run' the code of the adaptation pattern produced by WebCollage. In figure 4 the Output of the adaptation pattern (i.e. the adapted user interface) is presented. This output is adapted according to: a) each learner's answers (property personal_knowledge_level is set) and b) the answers of each learner's collaborators. Thus, each learner is classified as Novice or not. The learner is eventually offered support (Support Activity in IMS-LD terms) and is prompted to perform a specific course of actions (see figure 4).

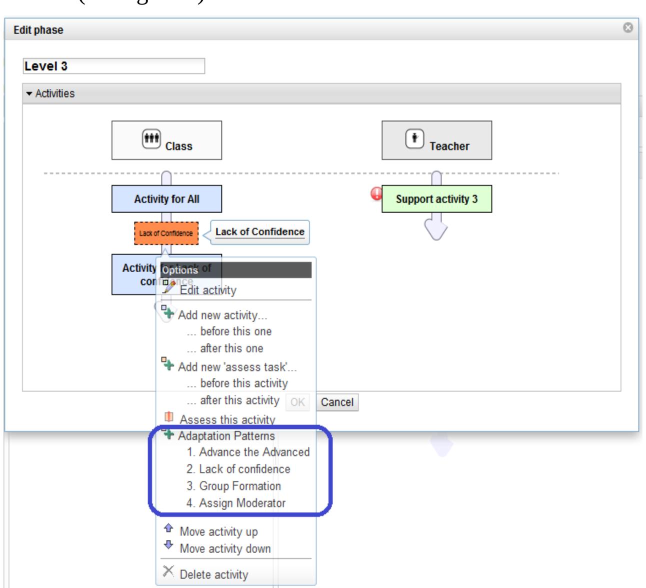

Figure 2. Submenu Adaptation Patterns in WebCollage

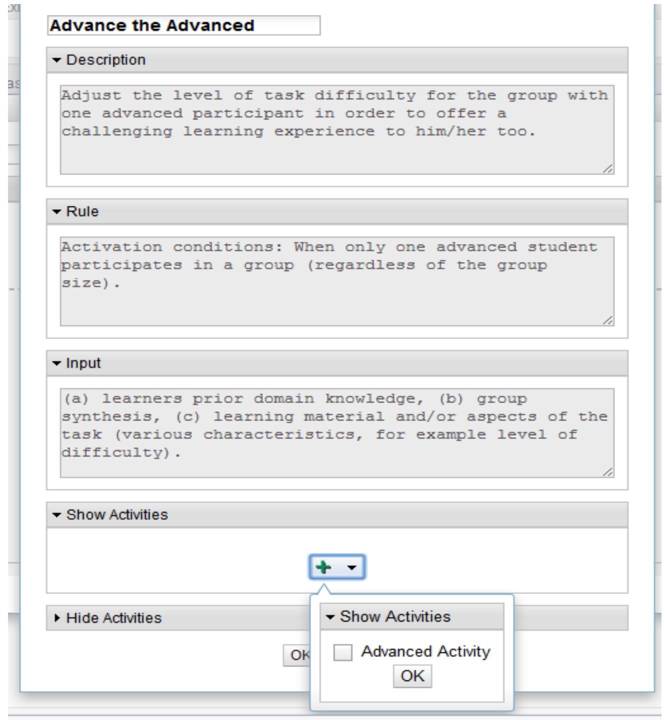

Figure 3. Dialog Boxes of two Adaptation Patterns in WebCollage

Figure 4. SLED screenshot of the adapted user interface for the novice learner according to the implemented adaptation pattern

4. DISCUSSION & FUTURE DIRECTIONS

A problem that had to be tackled was applying the theoretical framework (i.e. APs) to practice. The code editor of the WebCollage tool was modified: new capabilities are integrated and the whole experience of designing CSCL learning scripts is enriched. The following was successfully completed:

a) The Hellenization of WebCollage

b) The extension of WebCollage into a full IMS-LD compliant editor. The tool was not fully compatible with the specification at first and, the necessary corrections were made towards this objective.

c) Design and development of four Adaptation Patterns, namely the: Advance the Advanced, Lack of Confidence, Group of novices and Assign Moderator, thereby adding flexibility to WebCollage. Thus, WebCollage was leveraged to a tool, which allows to author-design adaptive collaboration scenarios.

It is also important to note, that the implementation of the Adaptation Patterns added to the tool the potential for incorporating IMS-LD Level B constructs (like properties conditions), a fact that extends the compatibility of WebCollage at level B of the standard IMS-LD. Moreover, the development of APs was based upon component-based highly decoupled implementation strategies, which applies for further enhancements (i.e. more APs, more changes in user interface etc.). Every change is documented and nothing from the original WebCollage tool functionality was jeopardized.

Future work includes carrying out experiments in a authentic or controlled environments with teachers and students in order to gather data on the usability of Adaptation Patterns, implemented in the design of adaptive collaborative scenarios. Moreover, data may arise, regarding the localization of the tool. An experiment like the one described here, can lead to the creation of new features or other improvements in the features already implemented. Our aim is to add new CLFP's in WebCollage environment, giving the designer the opportunity of employing more complex design flows. Furthermore, the target is to enrich the tool with new adaptation patterns, thus providing users with more customization options. Notice, that there are APs and even adaptations that even if they are modeled by IRMO need more complex implementations and cannot be totally integrated in an IMS-LD based environment (see [27]).

Additionally, a future direction of enriching WebCollage tool is to create an interface / tab which can give values to properties, at least for testing purposes. Moreover, we have opted for creating a component for communication with external tools such as the Moodle Forum. External tool information can be used as an input to an AP and can be held in WebCollage (in fact in IMS-LD properties). This information can be helpful for deciding, for instance, whether to apply an activation condition in another adaptation pattern. From our efforts working with WebCollage two more enhancements have been identified: a) To implement the capability inside WebCollage to create a copy of a script (duplicate). Due to the autosave functionality of WebCollage any change done during design is saved immediately. This poses a hindrance when someone wishes to save part(s) of a design. b) To add a text editor so that users can edit the document type resources within the tool WebCollage. However, as a first short-term goal, experiments in workshops are planned; during these experiments WebCollage is to be provided to teachers, letting them design (and run) scripts with adaptive features (i.e. incorporating APs in their design).

5. CONCLUSION

Research on collaborative learning has emphasized the need for providing flexible yet supportive tools to teachers in order to design collaborative learning tasks. In our work a next step in our pattern-based approach is presented demonstrating how educators' ideas can provide the basis for adaptation patterns which, in turn, can be expressed in IMS-LD modeling language. In this paper representative and selective design case studies are described exemplifying the implementation of the core specification of an Adaptation Pattern (Input, Rules, Model and Output) on the basis of using tools compliant to IMS-LD. It is analyzed what is necessary for implementing an adaptation pattern and discuss the benefits of the pattern-based approach. Finally, it is highlighted what issues would be important toward integrating the adaptation pattern capabilities in LD compliant tools for collaborative learning design.

6. ACKNOWLEDGMENTS

Our thanks to Yannis DImitriadis and the researchers and developers of the GSIC/EMIC research group at the University of Valladolid, which supported the current work. Special thanks to Dr. Eloy Villasclaras-Fernández, main author of the WebCollage tool and Vasilis Peitos, co-developer of parts of "Adaptation Patterns" functionality in WebCollage tool.

7. REFERENCES

[1] Gweon, G., Rosé, C.P., Carey, R., Zaiss, Z.S. 2006. Providing Support for Adaptive Scripting in an On-Line Collaborative Learning Environment. *In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Montreal, Quebec, Canada, 251–260.

[2] Harrer, A., Malzahn, N., Wichmann, A. 2008. The remote control approach - An architecture for adaptive scripting across collaborative learning environments. *jUCS* 14(1), 148–173.

[3] Karakostas, A., Demetriadis, S. 2008. Systems for Adaptive Collaboration Scripting: Architecture and Design. *In: Adaptive Collaboration Support Workshop in 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 7–12.

[4] Hewitt, J. 2005. Toward an understanding of how threads die in asynchronous computer conferences. *JLS* 7(4), 567–589.

[5] Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., Häkkinen, P., Fischer, F. 2007. Specifying computer-supported collaboration scripts. *ijCSCL* 2(2), 211–224.

[6] Bote-Lorenzo, M.L., Gomez-Sanchez, E., Vega-Gorgojo, G., Dimitriadis, Y.A., Asensio-Perez, J.I., Jorrin-Abellan, I.M. 2008. Gridcole: A tailorable grid service based system that supports scripted collaborative learning. *Computers and Education*, 51(1), 155–172.

[7] Dillenbourg, P., Tchounikine, P. 2007. Flexibility in macro-scripts for computer-supported collaborative learning. *JCAL* 23(1), 1–13.

[8] Demetriadis, S., Karakostas, A. 2008. Adaptive collaboration scripting: A conceptual framework and a design case study. *In: Xhafa, F., Barolli, L. (eds.) Proceedings of the CISIS 2008: 2nd International Conference on Complex, Intelligent and Software Intensive Systems*, 487–492. IEEE Computer Society.

[9] Demetriadis, S., Magnisalis, I., Karakostas, A. 2009. Adaptation Patterns in Systems for Collaborative Learning and the Role of the Learning Design Specification. *In: Scripted vs. Free CS Collaboration: Alternatives and Paths for Adaptable and Flexible CS Scripted Collaboration Workshop in CSCL 2009*, Rhodes, 43–47.

[10] Brusilovsky, P., Peylo, C. 2003. Adaptive and intelligent Web-based educational systems. IJAIED, Special Issue on Adaptive and Intelligent Web-based Educational Systems 13(2-4), 159–172.

[11] De Bra, P., Calvi, L.: AHA 1998. a Generic Adaptive Hypermedia System, *Proc. of the 2nd Workshop on Adaptive Hypertext and Hypermedia*, 5-12.

[12] Chen, W. 2006. Supporting teachers ’ intervention in collaborative knowledge building, *Journal of Network and Computer Applications*, 29, 200-215.

[13] Magnisalis, I., Demetriadis, S., Karakostas, A. 2011. Adaptive and Intelligent Systems for Collaborative Learning Support: A Review of the Field, *IEEE TLT*, 4(1), 5-20 .

[14] Karakostas, A., Demetriadis, S. 2010. Adaptation Patterns as a Conceptual Tool for Designing the Adaptive Operation of CSCL Systems. *Educational Technology Research & Development* 23(1), 1042–1629.

[15] Vygotsky, L.S. 1978. Mind in society: The development of higher psychological processes. Harvard University Press, Cambridge (1978).

[16] Nardi, B.A. 1996. Context and Consciousness: activity theory and human-computer interaction. The MIT Press, Cambridge.

[17] Hernández-Leo, D., Asensio-Perez, J.I., Dimitriadis, Y. 2005. Computational Representation of Collaborative Learning Flow Patterns using IMS Learning Design. *JETS* 8(4), 75–89.

[18] Perez-Sanagustin, M., Burgos, J., Hernandez-Leo, D., Blat, J. 2009. Considering the Intrinsic Constraints for Groups Management of TAPPS and Jigsaw CLFPs. *INCOS 2009*, 317–322.

[19] Hinze, U., Bischoff, M., Blakowski, G. 2002. Jigsaw Method in the Context of CSCL. *In: Proc. of World Conference in Educational Multimedia, Hypermedia and Telecommunications*, 789–794.

[20] Magnisalis, I., Demetriadis, S. 2011. Modeling adaptation patterns in the context of collaborative learning: case studies of IMS-LD based implementation, *In: Daradoumis, T., Caballé, S., Juan, A., Xhafa, F. (eds), Technology-Enhanced Systems and Tools for Collaborative Learning Scaffolding*, Springer, 279-310.

[21] Magnisalis, I., Demetriadis, S. 2009. Modeling Adaptation Patterns with IMS-LD Specification: A Case Study as a Proof of Concept Implementation, *Proceedings of INCOS '09*, 295-300.

[22] Magnisalis, I., Demetriadis, S., Pomportsis, A. 2010. Implementing Adaptive Techniques in Systems for Collaborative Learning by Extending IMS-LD Capabilities, *INCOS*, 70-77.

[23] Halm, J., Olivier, B., Farooq, U., Hoadley, C. 2005. Collaboration in Learning Design Using Peer-to-peer technologies. *In: Koper, R., Tattersall, C. (eds.) Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, 203–213. Springer.

[24] Koper, E.J.R., Olivier, B. 2004. Representing the Learning Design of Units of Learning. *Educational Technology and Society*, 7(3), 97–111.

[25] IMS LD, 2003. IMS Global Learning Consortium: Learning Design Specification, http://www.imsglobal.org/ specifications.html (accessed Jan. 5, 2011)

[26] Burgos, D., Tattersall, C., Koper, E.J.R. 2006. Representing adaptive and adaptable Units of Learning. How to model personalized eLearning in IMS Learning Design. *In: Fernández Manjon, B., Sanchez Perez, J.M., Gómez Pulido, J.A., Vega Rodriguez, M.A., Bravo, J. (eds.) Computers and Education: E-learning - from Theory to Practice*.

[27] Magnisalis, I., & Demetriadis, S. (2011). Modeling adaptation patterns in the context of collaborative learning: case studies of IMS-LD based implementation. *Technology-Enhanced Systems and Tools for Collaborative Learning Scaffolding*, 279-310.

[28] Coppercore. The IMS LD Engine (2008), http://coppercore.sourceforge.net (accessed January 5, 2013)

[29] SLeD. Service Based Learning Design Player (2005), http://sled.open.ac.uk/ (accessed January 5, 2013)

[30] Recourse Learning Design Editor (2009), http:// www.tencompetence.org/ldauthor/ (accessed Jan. 5, 2013)

[31] IMS Question & Test Interoperability Specification (2012), http://www.imsglobal.org/question/ (accessed Mar. 4 2013)

[32] WebCollage Tool (2010), http://www.ld-grid.org/resources/tools/webcollage, (accessed Mar. 4 2013)

[33] Dojo Toolkit 1.8 (2013), http://dojotoolkit.org/ (accessed March 4 2013)

[34] Paramythis, A. 2008. Adaptive Support for Collaborative Learning with IMS Learning Design: Are We There Yet?, *Proceedings of the Adaptive Collaboration Support Workshop*, held in conjunction with the Adaptive Hypermedia conference, 17-29.

[35] Towle, B., Halm, M. 2005. Designing Adaptive Learning Environments with Learning Design. *In: Koper R & Tattersall C (eds) Learning Design: A Handbook on Modelling and Delivering Networked Education and Training*, Berlin-Heidelberg: Springer, 216-226.

[36] de-la-Fuente-Valentin Luis, A. Pardo, , C.K. Delgado 2011. Generic service integration in adaptive learning experiences using IMS learning design, *Computers & Education*, 57, 1160-1170.

[37] Ferraris, C., Martel, C., Vignollet, L. 2007. Helping teachers in designing CSCL scenarios: a methodology based on the LDL language, *In Proceedings of the 8th Intl. CSCL*, 193-195.

[38] LD tools, http://www.ld-grid.org/resources/tools, (accessed Mar. 4 2013).

[39] Hernández-Leo, D., Villasclaras-Fernández, E. D., Dimitriadis, Y., & Asensio-Pérez, J. I.2010. Pattern languages for generating CSCL scripts: From a conceptual model to the design of a real situation. *E-learning, design patterns and pattern languages*, 49-64.

[40] Villasclaras, D. Hernández, J.I. Asensio, Y. Dimitriadis,, 2013. Web Collage: an implementation of support for assessment design in CSCL macro-scripts, Computers and Education, 59(), in press.