Marjanovic, O., Skaf-Molli, H., Molli, P., & Godart, C. (2007). Deriving Process-driven Collaborative Editing Pattern from Collaborative Learning Flow Patterns. *Educational Technology & Society*, 10 (1), 124-135.

Deriving Process-driven Collaborative Editing Pattern from Collaborative Learning Flow Patterns

Olivera Marjanovic
University of Sydney, Sydney, NSW 2006, Australia
o.marjanovic@econ.usyd.edu.au

Hala Skaf-Molli, Pascal Molli and Claude Godart
LORIA-INRIA, 615, Rue du Jardin Botanique, Villers-lès-Nancy, 54600, France
skaf@loria.fr
pascal.molli@loria.fr
godart@loria.fr

ABSTRACT

Collaborative Learning Flow Patterns (CLFPs) have recently emerged as a new method to formulate best practices in structuring the flow of activities within various collaborative learning scenarios. The term "learning flow" is used to describe coordination and sequencing of learning tasks.

This paper adopts the existing concept of CLFP and argues that many of these patterns are already using, or could use, collaborative editing activities that need to be process-driven. Consequently, the paper proposes a Process-Driven Collaborative Editing Pattern (PDCEP). The paper also describes how this new pattern relates to the existing examples of CLFPs and IMS-LD best-practices. The proposed pattern incorporates temporal and deontic constraints, used to specify the process of collaborative editing. The approach is demonstrated by an example of the electronic debate learning activity and its corresponding CLFP.

Keywords

Learning Designs, Collaborative Learning Flow Patterns, Coordination, Collaborative editing

Introduction

These days, numerous applications of the existing educational technologies are based on the same content-driven pedagogy. The same approach is also supported by the current educational standards. For example, SCORM (Sharable Content Object Reference Model) by (ADL, 2004) enables sequencing and dynamic presentation of the learning content to a learner, based on their progress through the prescribed material. Furthermore, although the concept of Learning Objects, as proposed by the Learning Object Metadata Standard (LOM, 2002), has evolved to include much more than educational content, it is still widely used to enable sharing of content-based educational resources among different repositories and educational platforms.

However, this content-oriented approach to learning and teaching does not really reflect the reality of creative teaching/learning processes. As (Koper and Olivier, 2004) pointed out, the content-driven pedagogy is based on the following, quite limited, set of guiding principles "Learning is the process of consumption of content… In order to learn, a single user needs to go through a sequence of learning objects… Teaching is the art of: (i) selecting and offering content in a structured, sequenced way and (2) tracking the learner's process and assessing the acquired knowledge" (Koper and Olivier, 2004, pg. 97).

In reality, teaching/learning processes are highly creative and involve collaborative interactions among many roles, guided by various pedagogical models. These processes need to be captured, shared and adequately supported by educational technology. This particular problem has been the main focus of the recently proposed theory of Learning Designs (Koper, 2005). This theory strongly promotes the top-down, rather than bottom-up design of learning experiences. More precisely, it starts from the proven pedagogical models rather than the available technology. This is why this emergent theory appears to be very significant for the future developments in the area of educational technologies (Britain, 2004).

Conceptually, learning designs describe "under which conditions, which activities have to be performed by learners and their teachers to enable learners to attain desired learning objectives" (Koper and Olivier, 2004). The associated

IMS-Learning Design standard (IMS-LD, 2003) promotes formal specifications of learning designs, executable by computers. In this paper, they are referred to as IMS-LD. Ultimately, IMS-LDs can be reused and shared by software platforms as well as teachers. The current version of IMS-LD formal model consists of three components: (i) an information model, (ii) a best-practice and implementation guide and (iii) an XML binding. For more details see (LD-IMS, 2003).

One of the ongoing research and practical challenges in this area is the problem of possible reuse of learning design methods (or their parts). A *learning design method* describes the actual *teaching-learning process* i.e. activities undertaken by all participants (teachers and learners) in the given learning environment (Koper, 2005). Learning design methods are typically derived from the existing educational theories, the examples of best practices used by experienced educators or from common patterns that can be observed in the best educational practices (Koper and Tattersall, 2005). In general, patterns are used to capture common solutions to recurrent problems in a particular domain (Alexander et. al. 1977). In this context, these are common solutions or pedagogical practices used in education. From the knowledge management perspective, it is possible to observe that educational patterns are used to express the experiential knowledge of teachers, related to common teaching/learning practices.

This paper focuses on a special category of patterns called Collaborative Learning Flow Patterns (CLFPs) that have been previously introduced by (Hernandez-Leo et al., 2005). These patterns focus on *learning flows* i.e. coordination and sequencing of learning tasks in a process. More precisely, "CLFPs formulate the best practices in structuring the flow of type of learning activities (and to some extent types of tools) involved in collaborative learning scenarios" (Hernandez-Leo et. al, 2005, pp.76). These patterns are identified by the process called deductive pattern mining, designed to capture the essence of generic models for solutions to recurrent problems identified by experienced learning designers (Baggetun, et. al. 2004). However, the main emphasis of CLFP is on the *learning flows* rather than other aspects of collaborative learning activities (such as, for example, instructional design). There are many examples of CLFPs that can be found in the field of collaborative learning such as Think-Pair-Share, Jigsaw, Pyramid and Brainstorm, as described by (Hernandez-Leo et. al, 2006).

Compared to the existing theory and practice of CLFP, this paper goes one step further. It recognises the fact that during collaborative learning students are often required to collaboratively edit/produce an artifact (i.e. a shared text document/ a piece of programming code /a presentation). In this paper, we use a generic term "document" to refer to all text-based shared artifacts. Collaborative editing is not a new concept. However, in many cases, collaborative editing needs to be *process-driven* i.e. all participants are required to perform certain editing actions at different times and these actions need to be coordinated. Consequently, components of a shared document need to be made available to the right person at the right point of time and these people should be given the appropriate editing rights. Furthermore, certain actions (e.g. editing events) may also generate further obligations for the other participants to perform some future actions within the same process. For example, when a teacher posts a debate issue, two teams of students will be obliged to post their arguments by the given deadline. In essence, many CLFP require (or may benefit from) collaborative editing activities driven by a process.

To capture and describe this process, this paper proposes Process-Driven Collaborative Editing Pattern (PDCEP). More precisely, the paper uses a high-level formal modeling language to express this pattern and capture its temporal and deontic constraints. Here, temporal constraints are used to express temporal relationships among various editing events while deontic constraints are used to capture dynamic obligations, permissions and prohibitions resulting from these events. Formal modeling of these constraints enables formal verification and reasoning about various editing events at the conceptual level.

The paper uses an example of Electronic debate CLFP to motivate the need for the proposed PDCEP. After formally describing this pattern, the paper will also discuss how selection of different types of documents and editing events can lead to different types of CLFP. The paper will also discuss how the proposed pattern relates to the examples of IMS-LD best practices, as described by (IMD-LD, 2003).

Motivating example

Suppose that a teacher is interested to implement an online debate to supplement her/his weekly lecture on computer ethics. From the existing education literature (Habeshaw et. al., 1987; Ramsden, 1992; Angelo and Cross, 1993;

Snider and Schnurer, 2002) s/he knows that the debate learning activity teaches persuasion, argument construction and improves students' critical thinking and communication skills. This well-known activity has been used in many different disciplines to help students gain a better understanding of a controversial issue and explore different perspectives of the same problem.

For the purposes of this exercise, the teacher decides to divide all students into several groups. Ideally each group should have an even number of members, although it is possible to use some other group formation strategy. Each group of students is given an issue to debate. Then each group (e.g. A) is divided into two teams, one that will argue the affirmative side (+) (here called TeamA+) and the other that will argue the negative side of the debate (-) (here called TeamA-). The teacher decides to post a debate issue online, as a follow up to his/her face-to-face lecture. For example, suppose that Group A is given the following issue to debate: "Keeping the electronic versions of private medical records is good for the society". TeamA+ and TeamA- are then given 4 days to prepare their respective arguments and post them online. After both teams have posted their arguments, they are then given 2 days to prepare and post their rebuttals in order to respond to the opposite team's argument. After both teams complete their rebuttals, the teacher will evaluate the quality of their arguments, provide online feedback and select the winning team. Obviously the same scenario is replicated for the other groups (B, C, D etc.)

This activity could be done as a part of student's formative assessment to make sure students understand a critical issue covered by the previous lecture and use the relevant resources to support their argument. Alternatively, it could be used as a part of their summative assessment.

CLFP of Electronic Debate

First of all, it is important to note that the electronic debate learning design method could be analysed from many different perspectives including, for example, instructional design, group dynamics etc. While acknowledging that all these perspectives are important for the given collaborative activity to succeed, this paper focuses on a single perspective, captured by the concept of learning flows. This particular perspective deals with the problem of coordination of individual tasks in the given learning/teaching process.

As a starting point we adopt the existing concept of the Collaborative Learning Flow Patterns, described by (Hernandez-Leo, et. al., 2005). In essence, CLFP represent patterns derived from the common learning design methods used in collaborative learning, that place a special emphasis on the learning flows among all participants in the same process. Therefore, other aspects of instructional design of collaborative learning, although very important, are out of the scope of these patterns. Table 1 describes the CLFP that corresponds to the previous motivating example. To express this CLFP, we use the formal notation also introduced by (Hernandez-Leo, et. al., 2005).

Table 1: Electronic debate described as CLFP, based on the model proposed by (Hernandez-Leo, et. al., 2005)

Facet	Explanation	Collaborative Learning Flow Pattern
Name	Name of the CLFP	Electronic Debate
Problem	Learning problem to be solved by CLFP	A complex, controversial issue that includes two opposing sides of the same argument. Participants are required to defend a position (i.e. argue for or against the debate issue).
Example	A real-world learning activity capable of being structured according to CLFP	Students debating a controversial computer ethics issue such as: "Keeping the electronic versions of private medical records is good for the society".
Context	Environment type in which the CLFP could be applied	Participants exploring two opposing sides of the same controversial issue.
Solution	Description of the proposal by the CLFP for solving the problem	Two debate teams are required to defend their given (chosen) position on the same debate issue. Each team studies the debate issue and collaboratively proposes their argument. When both teams complete their arguments, they are expected to respond to the opposing team's argument. After both teams complete their rebuttals, the teacher evaluates contributions of both teams and selects the winner.

Actors	Actors involved in the collaborative activity described by the CLFP	➢ Teacher ➢ Learners	
Types of Tasks	Types of tasks, together with their sequence, performed by actors involved in the activity.	Teacher: 1. Definition of a debate issue 2. Provision of additional information 3. Group/team selection and dimensioning (optional) 4. Decision about time for each step (i.e. definition of temporal constraints) 5. Progress Monitoring 6. Result evaluation and selection of a winning team	Learners: 1. Access to the debate issue 2. [REPEAT for each group] 3. Selection of two teams (i.e. *for* or *against* debate issue) 4. Team discussion 5. Proposal of team's argument 6. Review of the argument of the opposite team 7. Post team rebuttal 8. Review teacher's summary and the outcome (Until the wining team is selected for each group)
Types and Structure of Information	Description of the types of information identified in the collaborative activity and how they are related.	➢ Input information about the debate issue ➢ Collaborative arguments of both teams ➢ Rebuttals of both teams ➢ Result information including the evaluation of both sides of argument and the selected winning team	
Types and Structure of Groups	Description of the types of groups of learners identified and how they are related.	A number of groups (at least one), each divided into two teams.	

In order to explain how the proposed PDCEP has been identified, it is necessary to start from several important observations related to the above CLFP.

First of all, all roles are engaged in structured collaborative writing. Coordination of their individual and collaborative activities is guided by the corresponding learning flow model, as expressed by CLFP. The coordination aspect is critical for the implementation of this CLFP, because unless the individual activities are done in the right order and at the right time, the resulting process will not make any sense. For example, unless the teacher posts the initial debate issue, students cannot start preparing their arguments. Similarly, in order to prepare their rebuttal, each team should be able to see the argument posted by the other team.

Furthermore, there are also various temporal constraints that need to be implemented and monitored to make sure the activities are synchronised in the right way. These constraints are defined by the teacher to make sure that the overall activity can be completed as required, but at the same time, students are given enough time to complete each task. For example, students are given 4 days to complete the initial argument and 2 days for the rebuttal. Furthermore, when this particular activity is instantiated, students need to have access to various learning resources that are appropriate for this activity, at the right point of time (e.g. current version of lecture notes, relevant internet resources etc.).

It is also important to observe that all identified activities use the same document. Different teams will work on different components at different times. In essence this is an example of collaborative editing of a shared document that is guided by a process described by the corresponding CLFP. Therefore, different roles should have rights to edit different components as required, and these editing activities need to be coordinated. Consequently, all access rights need to be dynamically assigned and revoked. For example, once the initial arguments are completed, members of both A+ and A- teams cannot go back and edit their initial postings. It is also important to observe that access rights are given to roles for different components of the same document (rather than the whole document) for a particular period of time.

The proposed Electronic Debate CLFP is not the only example where process-driven collaborative editing could be, or is, currently used. For example, Brainstorming CLFP (Hernandez-Leo, et. al., 2005) could be also enhanced by process-driven collaborative editing. Thus, to make sure that *all* students contribute, a process could be implemented that require students to post their ideas in the round robin fashion. Again, all editing events need to be coordinated and all editing rights should be dynamically managed to prevent re-editing of the previous postings.

Furthermore, Pyramid CLPF (as described by Hernandez-Leo, et. al., 2005) could also use process-driven collaborative editing because the editing events need to be coordinated and contributions of different groups combined in a certain order. Similarly, the examples of IMS-LD best practices, such as Literature Circles (IMS-LD, 2003) could also incorporate process-driven collaborative editing in a similar fashion. This idea will be further explained in the discussion section of this paper.

In essence, all these examples illustrate that it is possible to observe a common pattern of collaborative editing that is process-driven. This important observation has led us to propose the *Process-Driven Collaborative Editing Pattern (PDCEP)* as described in the next section.

A Formal model of Process-Driven Collaborative Editing Pattern

Let $\mathcal{D}$ be a shared document that consists of a set of components:

$$\mathcal{D} = \{C1, C2, \ldots, Cn\}$$

where all components *C1, C2, ..Cn* are mutually exclusive (i.e. they do not overlap).

A set of roles participating in the collaborative editing process is represented by:

$$\mathcal{R} = \{R1, R2, \ldots, Rn\}$$

Note that, the same role can be shared by more than one person.

A set of editing activities that roles can perform on a given component is represented by

$$\mathcal{A} = \{A1, A2, \ldots, An\}.$$

Process-driven collaborative editing can be represented as a coordinated sequence of editing events *E1, E2, … En* where each event *Ei* is represented by a tuple:

$$\mathbf{Ei}\ (Rj, Ci, Ak, te)$$

where $Ri \in \mathcal{R}$; $Cj \in \mathcal{D}$; $Ak \in \mathcal{A}$ and *te* denotes the actual (absolute) time of *Ei* completion. This time can be used to determine the order of individual events. At the same time, these events correspond to different tasks as described by CLFP (i.e. *te* corresponds to their start or completion time).

Obviously, some actions (e.g. the teacher posts a debate issue) will result in the subsequent obligations for students to complete their respective arguments by the given deadline. On the other hand, other actions are more informative by nature and do not result in any subsequent obligations (e.g. a teacher posts an additional explanation for the given debate issue). To distinguish between these two types of actions, we adopt the classification proposed by (Searle, 1969) in the context of Speech act theory. This particular theory was developed for the purposes of formal modeling of various speech acts (actions) uttered by people participating in formal conversations. This theory argues that some speech acts (e.g. declarations) have the power to create further obligations for the participants in a particular type of formal conversation (e.g. business negotiation). To distinguish between these and other types of actions, Searle (1969) proposed the concept of *perfomative* and *informative* acts, where the former results in creation of obligations (e.g. an auction bid).

In the context of collaborative learning, we also observe that certain actions (editing events) will result in further obligations, while others are more informative. Therefore, we adopt the same theory and distinguish between *performative* and *informative* actions of different participants and their corresponding editing events.

In the context of collaborative editing this distinction is very important for two reasons. First of all, it is necessary to capture the resulting obligations of different participants so they can be monitored. Furthermore, these obligations create the need for management of dynamic access rights to different components of a shared document that will change according to the temporal constraints defined for the particular CLFP.

To express obligatory actions and dynamic access rights (including permissions and prohibitions), we use the concept of *deontic* constraints. They originate from the so-called deonic logic, introduced by von Wright (1968). This formal logic enables representation of permissions, prohibitions and obligations as well as their relationships (i.e. if an action is obligatory, it has to be permitted). Since its introduction, this formal logic has been used for modeling of organisational knowledge in different domains (e.g. electronic contracting and electronic negotiation systems).

In this context, we use deontic logic to express different obligations, permissions and prohibitions generated by different performative actions (editing events).

Thus, obligations are formally represented by a tuple

Obligation *(Ri, Cj, Ak, tb, te)*

where Ri $\in \mathcal{R}$; Cj $\in \mathcal{D}$; Ak $\in \mathcal{A}$ and (tb,te) is a time interval that denotes period of validity of this obligation. Both times tb and te will be dynamically instantiated during the actual learning activity.

This obligation is used to represent the fact that Ri is obliged to perform action Ak on document component Cj during the given time interval (tb,te).

To fulfill this obligation, role Ri has to have the appropriate access right. Thus, this obligation will automatically generate the following access rights:

Permission *(Ri, Cj, access-type, tb,te)*

where *access-type = {read, write}*

For example:

Permission *(teacher, C1, write, tb, te)*

indicates that the teacher has a permission to edit component C1 during the time period indicated by tb and te. Note that the time interval (tb,te) can be open on the right side to indicate that a particular role has access rights starting from tb until further notice.

Dynamic access rights to different document components will make these components visible to the appropriate roles at the right point of time.

Electronic debate CLFP and the process-driven collaborative editing pattern

Let us now use this formal model to express PDCEP used by Electronic debate CLFP. Due to the limited space, we will represent only a part of this model, that is sufficient enough to illustrate the proposed concept.

First of all, it is possible to identify a set of different document components that are used by different participants at different points of time. Figure 3 depicts the identified components along with their corresponding semantic meaning. Note that their relative position is arbitrary and used only to illustrate the concept.

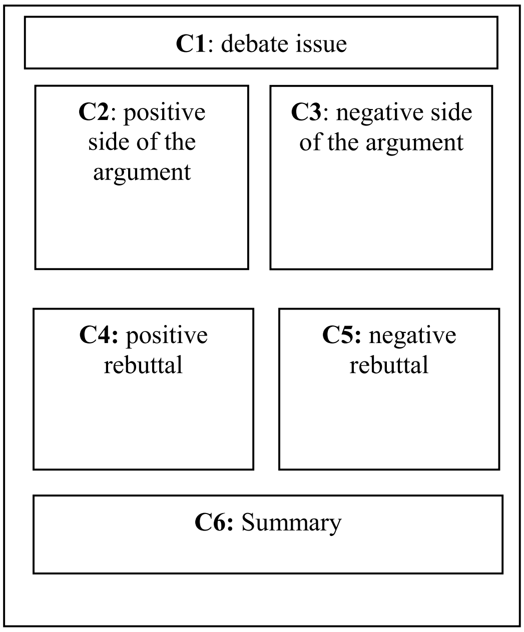

Figure 3: Components of a shared document

Furthermore, there are three different types of roles:

ℛ= {teacher, team+, team-}

note again that the same role may be played by more than one person.

Collaborative editing activities are represented by the following set:

𝒜 = {"post-debate-issue", "post-argument", "post-rebuttal", "post-summary"}

The very first collaborative event occurs when the teacher posts a debate issue:

E1*(teacher, C1, "post-debate-issue", t1)*

obviously the pre-requisite for this activity is that teacher has editing rights i.e.

Permission *(teacher, C1, write, t3, t4) where t4=t1*

This temporal constraint "*t4=t1*" is used to express the fact that once the teacher posts a debate issue, this posting cannot be changed. This is because the corresponding permission will be revoked at time *t4*.

Obviously this could be specified in a different way to allow teachers to change their original posting, but any change has to be implemented before students start working on their arguments.

The previous action "*post-debate-issue*" is a perfomative action that will generate the following obligations and permissions:

Obligation *(team+, C2, post-argument, tb, te)*

Obligation *(team-, C3, post-argument, tb, te)*

where in both cases *tb=t1* and *te = t1 + 4 days*

Recall that this temporal constraint "*te = t1 + 4 days*" was originally set up by the teacher, as described by the motivating example. It is used to set the deadline for the submission of the initial arguments by each team.

These obligations will generate automatic permissions for the members of *team+* and *team-* to post their respective arguments (components of the shared document).

However, the following explicit prohibitions have to be added to make sure the opposite teams do not see each other's postings.

***Prohibition** (team+, read, C3, tb, -)*

***Prohibition** (team-, read, C2, tb, -)*

where *tb = t1* and "-" is used to indicate an open time interval.

Suppose that events *E2* and *E3* correspond to *team+* and *team-* posting their arguments, at times *t5* and *t6* (respectively). Suppose that *t6* occurs after *t5* (i.e. *t5 < t6*).

Obviously, both *t5* and *t6* need to occur within the time interval determined by *t1* and *t4* i.e. both arguments need to be posted on time.

Before each team can prepare and post their rebuttals, they should be able to see the argument posted by the opposite team.

***Permission** (team+, C3, read, tb, -)*

***Permission** (team-, C2, read, tb, -)*

where *tb= t1 + 4* days.

The open time interval *(tb,-)* indicates that both teams will be able to see all postings after the deadline, in this case *tb.*

Alternatively, *tb = t6* means that after both teams post their arguments, they will be able to see the posting of the other team, even before the deadline. It is also possible to set up *tb* so one team can see the argument of the other team as soon as it is posted. This will be all determined by the teacher during the instructional design of this learning activity.

Finally, suppose that *E4* and *E*5 indicate that both teams posted their rebuttals at times *t7* and *t8*

***E4** (team+, C4, "post-rebuttal, t7)*

***E5** (team-, C5, "post-rebuttal", t8)*

again, *t7* and *t8* need to occur before the given deadline of 2 days.

Once they have completed their rebuttals, the access rights of both *team+* and *team-* with respect to components *C2, C3, C4* and *C5,* are revoked back to *"read"* and an obligation is created for the teacher to post a summary.

This activity could be also extended to enable knowledge sharing among different groups. For example, different groups could be given different issues to debate. Then after the teacher posts the summary for each group, all groups could be given the reciprocal rights to read documents posted by other groups.

Discussion

The main objective of this section is to position the proposed PDCEP within the context of the related work. This includes Collaborative Learning Flow Patterns (CLFPs), formal specifications of learning designs proposed by (IMS, 2003) and learning design best practices, as described by (IMS-LD, 2003). This section will also illustrate how the proposed pattern can be reused to create new examples of CLFPs.

First of all, it is important to point out that not every instance of collaborative editing activity is *process-driven*. A group of students can collaboratively write a document and their teacher could be only interested in the final outcome (e.g. the final document). The term *process-driven* means that editing events need to be coordinated but also constrained by the corresponding temporal and deontic constraints. In other words, events are constrained to appear in a certain order and each event may generate new obligations, prohibitions and permissions for other participants in the collaborative process. These constraints are determined by the corresponding CLFP.

Therefore, compared to the existing CLFP, the proposed pattern is conceptually located at a different level of abstraction. At the same time, different CLFP are likely to use, or could use, this pattern. For example, as already pointed out, Pyramid and Brainstorming CLFP as described by (Hernandez-Leo, et. al., 2005), could also implement a process to guide collaborative editing activities.

Furthermore, (Hernandez-Leo, et. al., 2005) describe the relationship between CLFPs and IMS-LD. They argue that collaborative services, currently supported by IMS-LD, are quite limited, as they include only email and discussion forum services. Consequently, they propose an extension to the IMS-LD specification of learning services to include a special type of service, called *groupservices*. In that respect, PDCEP is a further extension of the proposed specification of *groupservices,* at the conceptual level. More precisely, PDCEP describes the process of collaborative editing that is *independent* from any technical implementation. This new pattern includes temporal and deontic constraints (obligations, permissions and prohibitions), that are not captured by the current IMS-LD model of services. Consequently, it is possible to say that the proposed process-driven collaborative editing pattern is conceptually located “between” CLFP and learning services.

The need to formally express the collaborative editing process at the conceptual level, including different events and the associated temporal and deontic constraints, has prompted us to propose a formal model. Because this formal model has its foundation in formal logic, it is possible to use it to reason about different constraints (e.g. to answer questions such as “Is it possible to complete the whole process by the given deadline?”). It is also possible to verify their mutual consistency (e.g. obligations to do certain actions require corresponding permissions). This can be all done at the conceptual level, before the actual learning activity takes place. Reasoning and verification features can be also used to further enhance monitoring of process execution.

At the same time, it is important to point out that teachers are not expected to directly use the proposed formal model. The main reason for going to that level of details was to help us identify and formally express the main components of this pattern (i.e. performative and informative actions, obligations, permissions, prohibitions and temporal constraints). This, in turn, can be used to elicit the requirements for the implementation of an executable, configurable component that will support the proposed pattern. The main idea here is to provide teachers with a template of this pattern (process) so they can configure the individual elements (i.e. change the parameters of the process and the underlying document) in order to implement different CLFP without any programming involved. Ultimately, this component could be offered as a configurable learning service that could be reused by different CLFPs.

The proposed pattern could be also used to help teachers to invent new types of CLFP. This could be done by replacing a given document by other types of documents and changing semantic meaning of various editing events. For example, teachers can implement collaborative editing of their lecture notes. Thus, after a lecture, teacher can post lecture notes and give students several days to add any questions/comments to a particular slide. In the next step, the teacher will go over each slide, identify problematic issues and post a brief answer, give references to further reading or set up some additional activities for students to complete before the next class. Furthermore, the same pattern could be used to implement collaborative reflective journal writing by a group of students, problem-solving exercises, peer-assessment, various forms of electronic brainstorming (such as pooling, nominal group technique), collaborative writing of software applications etc.

Furthermore, PDCEP can be used to further enhance the existing examples of IMS-LD best practices, as described by (IMS-LD, 2003). For example, Literature Circle LD requires students to collaborate and their collaboration need to be process-driven. So when the "Discussion Director" posts a list of questions, this will create obligations for the other roles within the same Circle to contribute within the given timeframe. However, it is also very important to make sure that the overall process is not too restricted by the given constraints. Thus, as students progress through each Literature Circle, their teacher should have the ability to gradually remove the imposed constraints (if required).

In summary, we argue that any collaborative editing activity, where it is necessary to track individual editing events as well as monitor and manage the corresponding temporal and deontic constraints could benefit from the proposed pattern. Finally, it is important to point out that the same concept of temporal and deontic constraints could be further gerneralised and applied to other tasks in CLFP. However, in this paper we concentrate only on the collaborative editing activities. The following section discusses the main challenges related to possible implementation of PDCEP.

Implementation issues

It is possible to support the above described electronic debate activity, or any other collaborative editing activity, in many different ways. Obviously, e-mail or simple sharing of documents (for example in a word format) would make this activity very complex from the administrative point of view. The teacher would need to manually coordinate students' activities, and in some cases, even manually match the original arguments with their corresponding rebuttals for each team and each group. This could be a very complex and time consuming task, as all groups and teams are likely to use different documents.

Another possible option is to use the existing collaborative editing systems. These systems are used to support a group of people to collaboratively edit a document in the synchronous or asynchronous mode. Possible document types include text, diagrams, images, CAD drawings, multimedia, etc.

There are several notable examples of this category of systems. The first category includes Wikiwikiwebs or wikis. This is a family of very popular collaborative editors (http://www.wiki.org). In essence, these are web applications that allow users to freely create and edit a Web page content using any Web browser. In wikis application, there is a central wiki server and a wiki page is duplicated during an editing session without a locking mechanism. In the case of concurrent editing of the same document, the last saved document will be preserved. Modifications are not propagated. In wikis each "save" action, generates a new version of the document.

Real-time editors allow same time/ any place editing of the same document by multiple users. However, one of the major problems, that needs to be solved when building these systems, is the problem of concurrency control. Relevant examples of the previous work in this area are given by (Sun and Chen, 2002).

Currently, the area of collaborative editing mainly focuses on technical issues such as: concurrent editing, data replication and modifications propagation. However, none of these issues addresses the problem of structured collaborative editing as required by PDCEP, proposed in this paper.

In summary, a simple collaborative editing tool is not sufficient. It has to incorporate process-driven support that will guide the editing activities as well as support management of dynamic access rights, as required by the corresponding CLFP. Possible support for PDCEP could lead to more sophisticated learning services. These services could be then used to enable a new type of collaborative learning experience, well-beyond currently available forums and shared documents.

To address this design problem, we are currently considering possible extension of an existing collaborative platform called LibreSource (http://www.libresource.org). This platform was originally implemented to support collaborative development of open source software. Currently, this platform provides all services necessary for collaborative work. From the technical perspective, LibreSource is based on JAVA/J2EE technology. It includes an innovative data sharing management mechanism based on an optimistic data replication and synchronization tool, described by (Molli et. al, 2003). Compared to the existing tools in the same category (such as G-Forge http://gforge.org/ and Savannah http://savannah.gnu.org/), LibreSource offers a high level of integration. In addition to coordination, communication and forum tools, it also provides the awareness and collaborative editing tools. Our aim is to implement the process-driven collaborative editing pattern, described in this paper, by using a flexible coordination mechanism for the collaborative editing process. Implementation of a prototype of this technology is currently in progress by the LORIA ECOO team.

Conclusion

The emerging theory of Learning Designs sees learning/teaching activities as creative, collaborative activities where students and teachers play different roles and their activities are carefully coordinated to achieve the intended learning objectives. One of current research challenges in this area is certainly the problem of sharing and reuse of learning design methods.

This paper adopts the concept of Collaborative Learning Flow Patterns, proposed by (Hernandez-Leo, et. al., 2005). In essence, CLFPs focus on the learning flow perspective of the learning design methods. This paper argues that many CLFPs already include, or could include, collaborative editing of a shared artifact that needs to be process-driven. This means that individual editing events need to be coordinated. Furthermore, it is also necessary to dynamically manage access rights to different components of a shared artifact, to ensure that corresponding obligations can be fulfilled as required.

The main objective of this paper is to propose the so-called process-driven collaborative editing pattern (PDCEP). The formal model of this pattern includes temporal and deontic constraints, determined by the corresponding CLFP. The paper also positions the proposed pattern within the context of the related work including the existing CLFPs and relevant examples of IMS-LD best practices. The proposed approach is illustrated by an example of Electronic debate CLFP.

Our current work includes technical implementation of a prototype of the proposed pattern, based on the ideas presented in this paper. Our aim is to offer it as an executable component that could be reused by many different CLFPs, in the form of an user-friendly, configurable learning service. This service could be also integrated into the existing LD editing and management systems.

References

ADL (2004). SCORM 2nd Edition Overview, *Advanced Distributed Learning*, retrieved 5 September 2006 from http://www.adlnet.org.

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). *A Pattern Language*, New York: Oxford University Press.

Angelo, T. A., & Cross, K. P. (1993). *Classroom assessment techniques: A handbook for college teachers*, San Francisco: Jossey-Bass.

Baggetun, R., Rusman, E., & Poggi, C. (2004). Design patterns for collaborative learning: from practice to theory and back. In Cantoni, L. & McLoughlin, C. (Eds.), *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Norfolk, USA: AACE, 2493-2498.

Britain, S. (2004). A review of Learning Design: Concept, specifications and tools. *A report for the JISC E-Learning Pedagogy Programme*, May 2004, retrieved 5 September 2006 from http://dspace.learningnetworks.org/handle/1820/267.

Habeshaw, S., Habeshaw, T., & Gibbs, G. (1987). *53 Interesting Things to Do in Your Seminars and Tutorial*s, Technical and Educational Services.

Hernandez-Leo, D., Villasclaras-Fernandez, E. D., Asensio-Perez, J. I., Dimitriadis, Y., Jorrin-Abellan, I. M., Ruiz-Requies, I., & Rubia-Avi, B. (2006). COLLAGE: A collaborative Learning Design editor based on patterns, *Educational Technology & Society, 9* (1), 58-71.

Hernandez-Leo, D., Asenio-Perez, J. I., & Dimitriadis, Y. (2005). Computational representation of Collaborative Learning Flow Patterns using IMS Learning Design, *Educational Technology and Society, 8* (4)*,* 75-89.

IMS (2003). *IMS Learning Design v.1.0. Final Specification,* retrieved 5 September 2006, from http://www.imsglobal.org/learningdesign.

IMS-LD (2003). *IMS Learning Design. Information model, best practices and implementation guide*, XML Binding, schemas. Version 1.0 Final Specification, IMS Global Learning Consortium, retrieved 5 September 2006 from http://www.imsglobal.org/content/learningdesign/.

Koper, R. (2005). Learning Design: State of the Art and Future Perspectives. *Keynote Talk at the IEEE International Conference on Advanced Learning Technologies*, *ICALT'2005*, July 5-8, 2005, Kaohsiung, Taiwan.

Koper, R., & Olivier, B. (2004), Representing the Learning Design of units of learning. *Educational Technology and Society, 7* (3), 97-111.

Koper, R., & Tattersall (2005), *Learning Design: A Handbook on Modeling and Delivering Networked Education and Training*, Springer, Berlin.

LOM (2002). *Standard for Learning Object Metadata*, Learning Technologies Standards Committee of the IEEE 148.41.21.

Molli, P., Oster, G., Skaf-Molli, H., & Imine, A. (2003). Using the transformational approach to build a safe and generic data synchronizer. *Paper presented at the the ACM 2003 International Conference on Supporting Group Work (GROUP 2003)*, November 9-12, 2003, Sanibel Island, Florida, USA.

Ramsden, P. (1992). *Learning to teach in higher education*, London: Routledge.

Searle, J. R. (1969). *Speech Acts - An Essay in the Philosophy of Language*, Cambridge: The University Printing House.

Snider, A. C., & Schnurer, M. (2002). *Many Sides: Debate across the Curriculum*, Amsterdam: International Debate Education Association.

Sun, C., & Chen, D. (2002). Consistency maintenance in real-time collaborative graphics editing systems. *ACM Transaction Computer-Human Interaction, 9* (1), 1-41.

von Wright, C. G. (1968). *An Essay in Deontic Logic and the General Theory of Action*, Amsterdam: North Holland Publishing.