Innovative Learning Designs Enabled by Process-Driven Collaborative Editing

Olivera Marjanovic, Hala Skaf-Molli, Pascal Molli, Claude Godart

Innovative Learning Designs enabled by process-driven collaborative editing

Olivera Marjanovic, Hala Skaf-Molli* , Pascal Molli* and Claude Godart*

University of New South Wales
Sydney, NSW 2052, Australia
o.marjanovic@unsw.edu.au

*LORIA-INRIA
615, Rue du Jardin Botanique
Villers-lès-Nancy, 54600, FRANCE
{skaf, pascal.molli, godart}@loria.fr

Abstract:

Learning Designs (LDs) are used to describe, at the conceptual level, a learning experience in form of a teaching/learning process that is based on a particular pedagogical model. More precisely, LD defines which activities need to be performed by teachers and students, under which conditions and with what resources to enable students to achieve the intended learning objectives. This is a very new area of research, related to both educational theory and technology and consequently there are still many open research problems. One of them is identification of possible LD patterns that can be reused to create more examples of innovative teaching/learning activities.

The main objective of this paper is to address this research problem for a particular type of learning design. More precisely, the paper will describe the process of identification and formal modeling of a LD pattern called *process-driven collaborative editing*. It will also illustrate how this pattern can be reused to generate more examples of innovative learning/teaching activities. The proposed approach is demonstrated by an example of the electronic-debate activity that can be used both in e-learning as well as the blended mode of teaching/learning.

1. Introduction

These days most universities are using some form of educational technology. In addition to domain and content specific applications (e.g. a multimedia chemistry lessons), there are many more generic, web-based applications used for different aspects of student and learning management. The existence of a large number of different systems (even within the same organisational environment) has created the need for standardisation of content, tools and approaches.

Most of the existing standards concentrate on educational content (learning and teaching resources). For example, the LOM: Learning Object Metadata Standard (LOM, 2002) enables sharing of learning objects (LOs) among content repositories. Another important standard is certainly SCORM (Sharable Content Object Reference Model) by (ADL, 2004). This standard enables sequencing and dynamic presentation of learning content to a learner, based on their educational needs and progress through the prescribed content.

However, it is necessary to point out that this "content-oriented" approach to learning and teaching does not really reflect the reality of creative teaching/learning processes. As (Koper and Tattersall, 2005) 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, 2005, pg. 97).

In reality, teaching/learning processes are highly creative and involve collaborative interactions among many roles, guided by various pedagogical models. This was exactly the main motivation for the development of the emerging theory of Learning Designs (LD) (Koper, 2005) and its associated standard called the IMS Learning Design Specification (LD-IMS, 2003). *Learning Design (LD)* is used to describe, at the conceptual level, a learning experience in form of a teaching/learning processes. More precisely, LD defines "under which conditions, which activities have to be performed by learners and teachers to enable learners to attain the desired learning objectives" (Koper and Olivier, 2004). The LD theory is very significant for future developments in the area of educational technologies, because, for the first time, it promotes the top-down (rather than bottom-up) design approach. More precisely, it starts from the proven pedagogical models rather than the available technology. Furthermore, LDs can be used to describe any teaching/learning scenarios (that can include both technology-supported as well as face-to-face, or manual tasks). Thus, they can be used to describe learning/teaching scenarios in both e-learning as well as the mixed (blended) mode of learning.

Due to the fact that this is a relatively new area of educational research, there are currently many open research challenges generated by modeling and current implementations of learning designs as well as the current version of the IMS-LD standard. One of the current research challenges includes identification of learning design patterns (Koper, 2005). This is a very important problem, because such patterns, when *identified* can be *reused* to enable creation of new teaching/learning activities and scenarios.

The main objective of this paper is to address this research problem for a particular type of learning design. More precisely, the paper will describe the process of identification and formal modeling of a LD pattern here called *process-driven collaborative editing*. More importantly, it will then illustrate how the same pattern can be reused to generate more examples of innovative learning/teaching activities. The proposed approach is demonstrated by an example of the electronic-debate activity that can be used both in e-learning as well as the blended mode of teaching/learning.

2. An overview of the LD theory

This section gives a brief overview of the current state-of-the art in the LD theory as described by (Koper and Tattersall, 2005), (Koper, 2005), (Koper and Olivier, 2004) and (Britain, 2004). The purpose of this overview is to introduce the basic terminology as well as to describe the underlying principles of this theory and the related standard.

As already pointed out, LDs are used to describe a learning/teaching process that can be based on any pedagogical model. "It is an application of a pedagogical model for a specific learning objective, target group and a specific context or knowledge domain" (Koper, 2005). These pedagogical models describe a set of experiential rules or guidelines that describe how to achieved the set of learning objectives in the given educational domain in the best possible way. They can come from the experiences of individual teachers, best practice examples, educational theories etc.

Furthermore, when implemented every LD refers to a set of physical resources (learning objects and learning services) that are needed during the actual implementation of the process. LD and its associated resources are packaged into "units of learning" (UOL) that can be further distributed and reused a number of times.

The central ideas behind LD theory and practice are not new but are more relevant than ever, especially having in mind the widespread use of content-oriented educational systems. These ideas can be summarised as follows (Britain, 2004):

- people learn better when they are actively involved in the learning process
- learning activities may be structured (e.g. sequenced) in a *learning workflow*

- it would be useful to be able to record learning designs (innovative practices) for future sharing and reuse.

Hence, a LD is a high level description of a learning/teaching scenario that can be implemented by using a number of different tools. These are multi-role processes. Thus, every person (teacher or student) plays a role in the teaching/learning process. In this role, a person performs more or less structured learning and/or support activities that are designed to help them to achieve the intended learning outcomes. All activities are performed within an environment that incorporates the necessary learning objects as well as services that need to be activated and used during teaching/learning process. However, rather than restricted by technology, all activities are guided by pedagogical methods. These models determine which activities need to be done, by which role and how they all need to be coordinated to achieve the given objectives.

To illustrate the basic concept of roles and their synchronised activities, Koper and Tattersall (2005) use the metaphor of a theatrical play (as shown by Figure 1). A play consists of one of more sequential act(s) and an act is related to one or more concurrent role-parts (defining which role plays which activity). The acts in a play can follow a sequence or a more complex structure that may include concurrent acts. Activities can be assembled into activity-structures. An activity is played in an environment that consists of learning objects (resources) and learning services (various applications or tools) necessary for its execution. All roles and their corresponding activities are guided by the shared theatrical script. The same script is a generic description that can be reused many times, in many different environments by different people playing the same roles.

However, it is also important to note, that a LD "script" is meant to be much less prescriptive that the script used in a theatrical play. Thus, if necessary, new activities can be implemented during the actual teaching/learning process. This is exactly what good teachers do. They carefully observe the progress of their students and are ready to attempt the alternative ways to achieve the same learning objectives. Only in this way we can ensure that learning/teaching activities are truly flexible processes and driven by teachers rather than technology.

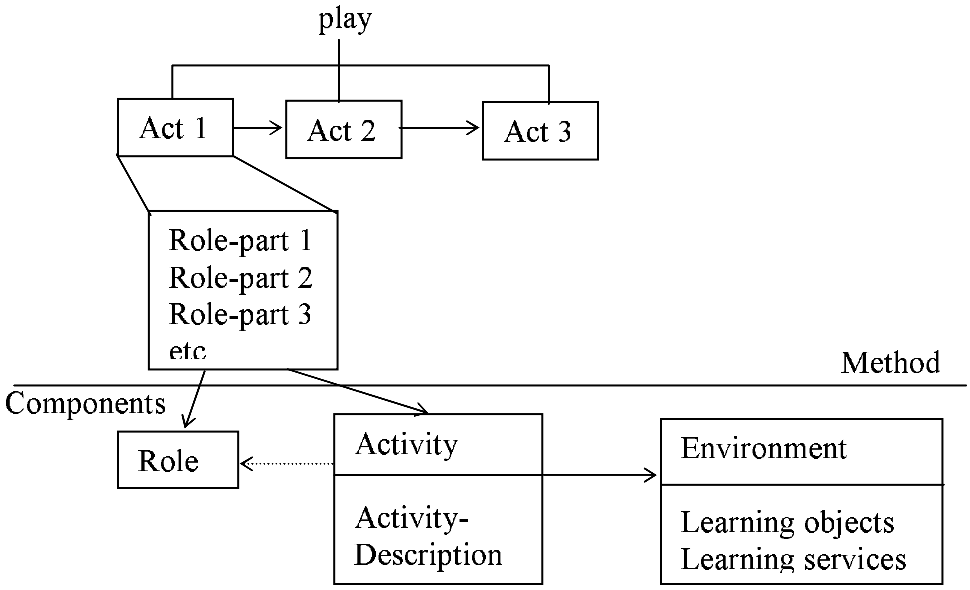

Figure 1: LD represented by the metaphor of a theatrical play (Koper and Tattersall, 2005)

The theoretical model of LD has also resulted in the emerging standard proposed to cover different aspects of LD modeling and implementation. The main objective of the IMS Learning Design Specification Standard (IMS-LD, 2003) is to make digital representation of learning designs consistent and consequently reusable by different software packages. The current version enable specification of learning objectives, creation of method using play, act and role-parts as well as specification of roles, resources environment and services (Britain, 2004).

Even though the theory and practice of learning designs is relatively new, there are already notable developments of various systems that either directly implement learning designs or follow the underlying principles. Examples include Coppercore, EduBox, Eduplone, LAMS, Lobster, Reload Software. An excellent comparison of these systems is presented in (Britain, 2004)) while (Koper and Tattersall, 2005) describe them in more details.

3. Motivating example

This section gives introduces a motivating example of a learning design that will be later used to illustrate the process of derivation of the process-driven collaborative editing pattern. Although this example can be used in e-learning, in this paper we consider its use in the blended (mixed) mode of learning.

Suppose that a teacher is interested to implement an online debate to supplement her weekly lecture on computer ethics. From the existing education literature, she knows that a debate is a very powerful teaching/learning activity that teaches persuasion, argument construction and helps to improve critical thinking and communication skills. This particular learning activity is not new. It has been used across all disciplines to help students to gain a better understanding of a controversial issue or a different point of view.

For the purposes of this exercise, the teacher decides to divide all students in groups, ideally with even number of members (e.g. in groups of four) although she could use some other strategy. Then each group is divided into two teams, one that will argue the affirmative side (+) and the other that will argue the negative side of the debate (-). Each group of students is then given one issue to debate. The teacher decides to post debate issue(s) online, as a follow up to her lecture. For example, suppose that group A is given the following issue to debate: “Keeping electronic versions of private medical records is good for the society”. Team A+ (the affirmative) and A- (the negative) are then given 4 days to prepare their respective arguments and post them online. When both groups have posted their arguments, they are then given 2 days to prepare and post their rebuttals to respond to the opposite team’s argument. After both groups complete their rebuttals, the teacher will then formally evaluate the quality of their arguments, provide online feedback and select the winning team.

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, so the students will be given a formal mark for their work.

4. Electronic debate LD and its corresponding pattern

Before we can discuss possible support of electronic debate activity by educational technology, it is necessary to further analyse its properties by using the LD theory. The main objective of this section is to illustrate this process as well as the process of derivation and reuse of its corresponding LD pattern. The implementation issues are discussed later in the paper.

4.1. Electronic Debate LD

The motivating example is an example of LD. All roles are engaged in collaborative writing. Coordination of their individual and collaborative activities is guided by a debate process model. The coordination aspect is critical for the implementation of this LD, because unless the activities are done in the right order and at the right point of 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 of the opposite team.

There are also various temporal constraints that need to be implemented and monitored to make sure the activities are synchronized in the right way. In this particular example, students are given 4 days to complete the initial argument and 2 days for the rebuttal. This is set up by the teacher to make sure that the overall activity can be completed on time e.g. before the next lecture. Furthermore, when this particular activity is instantiated, students need to have access to various learning resources that are appropriate for this environment and this particular group of learners (e.g. current version of lecture notes, relevant internet resources etc.).

It is important to point out that, form the educational perspective design of learning activities is guided by the activity-centered instructional design approach (rather than "technology" or "resource" centered design). This approach was introduced by (Oliver, Harper, Hedberg and Willis, 2002) and (Agostinho, Oliver, Harper, Hedberg and Wills, 2002) and further extended by (Marjanovic, 2005) by placing the special emphasis on its process aspect.

Figure 2 depicts a simple, graphical model of the "electronic debate" LD and, in particular, its coordination aspect. It is possible to use a variety of existing languages (e.g. graphical such as UML or XML based such as XML-LD) to model this example in more details. However, for the purposes of this paper, we choose a very simple language to express the coordination requirements of this LD as they will be needed in the subsequent sections of this paper.

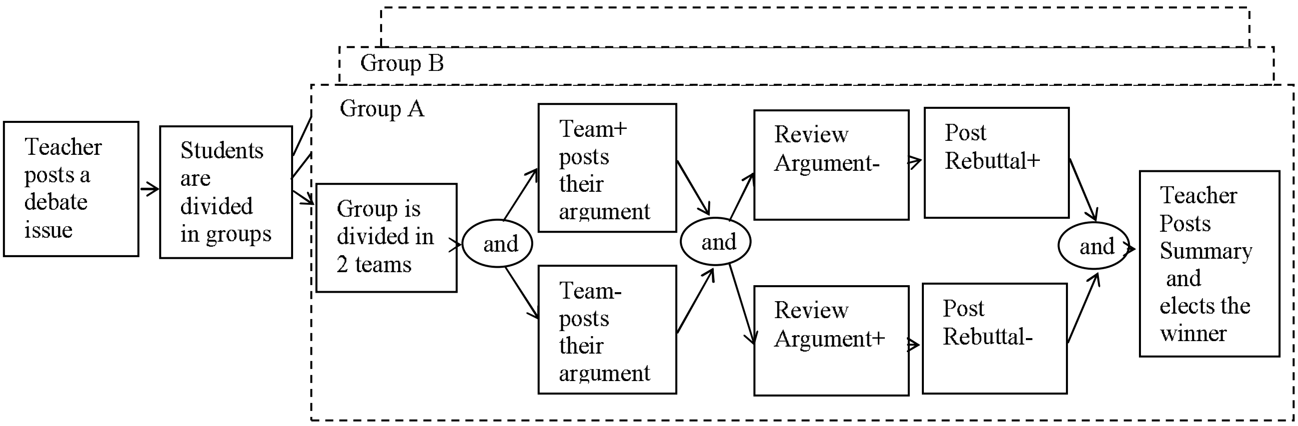

Figure 2: The coordination aspect of "electronic debate" LD

4.2. Identification of process-driven collaborative editing pattern

Identification and derivation of a possible design pattern form LD is both a creative as well as an exact (engineering) process. Its creative part makes it hard to be precisely described and generalised. From the knowledge management perspective, this is a process of externalisation of tacit knowledge with the purpose of its later reuse.

In this particular case, we start the process by observing several properties of this learning design. Firstly, all identified activities use the same document. Furthermore, different roles should have rights to edit different components of the same document as required by the corresponding LD. As already pointed out, all activities have to be properly coordinated to achieve the intended learning objectives. Consequently, all access rights are dynamically created and revoked as required by the process. 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 time period again as guided by the intended learning activity.

Hence, the process of collaborative editing is driven by a process model, that fully corresponds to the given learning design. However, the process-driven collaborative editing pattern is a process represented at the much lower level of abstraction. Consequently, it is independent from the application domain. More importantly, when identified, it becomes independent from the original learning design. This paper argues that this is a crucial property of a design pattern that will enable its reusability (as illustrated in the next section).

So in this case, we can observe that the process-driven collaborative editing pattern consists of a set of coordinating editing events. Formally, we can represent it as follows:

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

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

where all components C1, C2, ..Cn are mutually exclusive, thus they do not overlap.

A set of roles that participate in collaborative editing 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

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

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

$$Ei\ (Rj, Ci, Ak, te)$$

where Ri $\in\mathcal{R}$; Cj $\in \mathcal{D}$; Ak $\in\mathcal{A}$ and te is an absolute time value that denotes the time of completion of Ei

Editing events could be ordered in a sequence, could be parallel or even conditional (based on the previous activities). Consequently, their coordination structure could be easily represented by a high-level coordination graph that corresponds to the previously identified learning design (as depicted by Figure 2).

However, a more challenging issue is modeling of dynamic obligations and access rights, crucial for this particular design pattern. For this purpose, we distinguish between two different types of actions: *performative* and *informative*. Performative actions generate further obligations for other roles and result in dynamic change of access rights (creation of new permissions and prohibitions). For example, when the teacher posts the debate issue, completion of this event will create further obligations for students to post their arguments. Other actions are informative i.e. they do not result in any further obligations or change of access rights. For example, students can read a given component of the document. The concepts of informative and performative actions, used in this paper, originates from the *Speech-act theory*, formally introduced by Searle (1969).

Furthermore, to express obligatory actions and as well as dynamic access rights (permissions and prohibitions) we use the concept of deontic constraints. They originate from the so-called deonic logic. This is the formal logic of permissions, prohibitions and obligations, originally introduced by von Wright (1968). Since its introduction, this logic has been widely used to model various aspects of organisational knowledge.

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 constraint.

The previous example indicates that role Ri is obliged to perform action Ai on document component Ci 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 appropriate access rights:

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

where access-type = {read, write}

For example:

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

Indicates that teacher has a permission to edit component c1 during period indicated by tb and te. Note that the time interval (tb,te) can be open on the right hand side to indicate that a role has a certain access right starting from tb until further notice.

Dynamic access rights to different document components will make these components visible to the roles given the rights to read or edit them.

The following section illustrates how this model can be used to instantiate "process-driven collaborative editing pattern" to represent the previous motivating example (as described in section 3).

4.3. Electronic debate expressed by a process-driven collaborative editing pattern

Let us now use this formal model to express the process-driven collaborative editing pattern used for the "electronic debate" learning design. Due to the limited space, we will represent only a part of this design to illustrate the expressability of the introduced pattern.

First of all, we identify a set of document components that will be used in this particular learning design. Figure 3 depicts the identified components along with their corresponding semantic meaning (relevant for this example of LD).

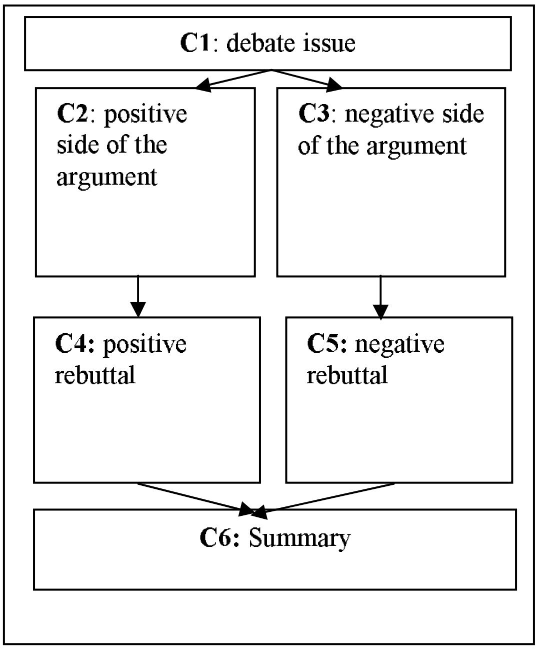

Figure 3: Components of the collaborative document

Furthermore, there are three different types of roles:

$$\mathcal{R} = \{\text{teacher, team+, team-}\}$$

Note that the same role may be played by more than one person (i.e. more than one member of team+ or team-)

Collaborative editing activities are represented by the following set:

$$\mathcal{A} = \{\text{“post-debate-issue”, “post-argument”, “post-rebuttal”, “post-summary”}\}$$

The very fist collaborative event represents teacher posting an issue:

E1(teacher, C1, “post-debate-issue”, t1)

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

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

this is used to indicate the fact that once teacher post a debate issue this posting cannot be changed). 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

These obligations will generate the automatic permissions for the members of team+ and team- to edit the respective components. However, the following prohibitions have to be added to make sure the opposite teams do not see each other's postings.

Prohibition (team+, read, C3, tb, te)

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

Similarly, events E2 and E3 will correspond to team+ and team- posting their arguments are parallel. When they are completed, they are followed by a set of parallel activities E4 and E5 where teams post their rebuttals.

E4(team+, C4, "post-rebuttal, t4, -)

E5(team-, C5, "post-rebuttal", t5, -)

"-" is used to indicate an open time interval. In this case it could be interpreted as "until further notice".

These two editing activities require the *corresponding components* to be visible. In other words, team+ should be able to see the previous posting of team- and vice-versa.

Permission(team+, C3, read, t3, -)

D-constraint(P, team-,C2, read, t3, -)

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, the same activity could be used by more then one group, but groups could be given different issue 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.

4.3. Reusing the process-driven collaborative editing pattern

It is important to observe that the same pattern can be easily replicated and reused with a number of different groups of students, irrespectively of the teaching discipline. By creating different editing sequences and using deontic constraints to specify obligations as well as editing rights (permissions and prohibitions) it is possible to generate a variety of scenarios.

More importantly, this process-driven collaborative editing pattern can be used to help teachers to invent new types of learning designs (innovative learning/teaching activities). This could be done by replacing the given document by other types of documents and changing the semantic meaning of various editing events. For example, teachers can implement collaborative editing of lecture notes. Thus, after a particular lecture, teacher can post lecture notes and ask students to add questions to every slide if there is an issue or concept they do not understand on that particular slide. For example, students may be several days to complete this activity. In the next step, the teacher will go over each slide, identify problematic issue and then post a brief answer, references to further reading or decide to address the problem at the beginning of the next lecture.

The same design pattern can be used to implement collaborative reflective journal writing by a group of students, problem-solving exercises, peer-assessment, various forms of electronic brainstorming (including pooling, nominal group technique etc.), collaborative writing of software applications etc. In essence, all these learning activities reuse the same pattern where collaborative editing of the same document is guided by the process that correspond to the intended learning design.

Finally, it is also very important to note that we do not expect teachers to use the formal model of design pattern as defined in the previous two sections. The main reason for going to that level of details was to help us understand the components of this pattern (i.e. performative and informative actions, obligations, permissions, prohibitions and temporal constraints).

However, this formal modeling can serve one more purpose. It can be used to elicit the requirements for the implementation of an executable component that will support a process-driven collaborative editing 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) implement different LD without any programming involved. This component could be offered as a part of knowledge sharing environment designed to help teachers to share their innovative learning designs. An example of such environment is the web-based handbook of LD, introduced by (Marjanovic, 2005). However, to make this idea possible it is necessary to solve some technical issues involved in implementation of process-driven collaborative editing pattern as described in the next section.

5. Implementation issues

It is possible to support the electronic debate activity in many different ways. Because we are interested to extend the learning experience beyond lectures, this could be an ideal online activity. So how to support this activity? 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. Our teacher would need to manually coordinate student's activities and in some cases even manually match the original arguments with their corresponding rebuttals for each team and each group.

Another possible option is to use the existing collaborative editing systems. These systems are used to support a group of people editing a document collaboratively synchronously or asynchronously over the computer network. People may work simultaneously on the same document, simultaneously on different copies of the document, or at different times on the original or copied document. The document types include text, diagrams, more complicated graphic objects, images, CAD drawings, multimedia, etc.

There are several notable examples of this category of systems. For example, Wikiwikiwebs or wikis is a family of very popular collaborative editors[]. It is a web application that allows users to freely create and edit Web page content using any Web browser. In wikis application, there is a central wiki server. A wiki page is duplicated during editing session without locking mechanism. In case of concurrent editing of the same document, the last saving the document will win. There is no propagation of modifications. In wikis each saving command generates a new version of the document.

Another family of collaborative editing tools are real-time editors. These tools allow multiple users to concurrently edit the same document from different sites. However, one of the major problems, that need to be solved when building such systems, is the problem of concurrency control. Examples of previous work in this area are given by [ ].

It is important to note that most issues faced in collaborative editing are usually technical. They focus on concurrent editing, data replication and modifications propagation. However, none of them address the problem of structured cooperative editing as required by the process-driven collaborative editing pattern, identified in this paper.

In conclusion, 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 particular learning designs.

To address this design problem, we are currently considering possible extension of an existing collaborative platform called LibreSource (Godart et all, 2005). Originally LibreSouce was developed to support cooperative open source software development. It is designed to provide all services necessary for cooperative work. From the technical perspective, LibreSource provides an innovative data sharing management mechanism, based on an optimistic data replication and synchronization tool (..). In addition to coordination, communication and forums tools, it includes awareness and collaborative editing tools. LibreSource is based on the JAVA/J2EE technology. Its decentralized approach made it possible to develop a set of robust applications while disregarding the underlying software architecture. Our aim is to implement the process-driven collaborative editing pattern as described in this paper by incorporating flexible coordination mechanism for the collaborative editing process. Implementation of a prototype of this technology is currently in progress by the LORIA ECOO team.

8. Conclusion

The content-oriented pedagogy, currently used by the existing educational technologies is very limited. It is based on a fundamental principle that learning is consumption of educational content and teaching is the art of sequencing and presenting the content to students.

The emerging theory of Learning Designs (LD) sees learning/teaching activities in a fundamentally different way. They are viewed as creative, collaborative activities where students and teachers play different roles and their activities are carefully coordinated to achieve the intended learning objectives. This is a new research area of educational technology and there are many open research problems yet to be solved.

This paper aims to address a particular research problem – identification and reuse of LD patterns for a particular type of LD. This is a challenging problem that requires both creativity as well as domain knowledge. We illustrate the process of identification and derivation of a specific LD pattern called *process-driven collaborative editing pattern* and describe how this pattern can be used to generate more examples of innovative LDs. The approach is illustrated by an example of LD called electronic debate that is commonly used in many different teaching domains.

Our current work includes technical implementation of a prototype of process-driven collaborative pattern based on the ideas presented in this paper as well as user friendly modeling of this particular type of collaborative process. Our aim is to offer it as an executable component that could be reused by teachers without any programming involved.

References

Agostinho, S, Oliver, R., Harper, B., Hedberg, H., & Wills, S. (2002). A tool to evaluate the potential for an ICT-based learning design to foster "high-quality learning". Proceedings of the 19th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education. (pp. 29-38). Auckland, New Zealand:

Oliver, R., Harper, B., Hedberg, J. and Willis, S. (2002), "Information and Communication Technologies and their role in Flexible learning", *AUTC project report*, December 2002 (retrieved August 2004 from www.learningdesigns.uow.edu.au).

von Wright, C.G (1968), "An Essay in Deontic Logic and the General Theory of Action", Acta Philosophica Fennica, 21, North Holland.

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

ADL (2004), SCORM- 2nd Edition Overview, Advanced Distributed Learning, [On-line]. Available http://www.adlnet.org

Britain, S. (2004), "A Review of Learning Design: Concept, Specifications and Tools", *A report for the JISC E-Learning Pedagogy Programme*, May 2004, [On-line] Available http://dspace.learningnetworks.org/handle/1820/267.

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, Inc. [On-line] Available http://www.imsglobal.org/content/learningdesign/

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

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

Koper, R. and Tattersall (eds.) (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

Marjanovic, O. (2005a), "Towards Web-Based Handbook of Process Oriented Learning Designs", *Educational Technologies and Society journal*, IEEE Learning Technology Task Force, 8 (2), pp. 66-82.

Hala to add references for collaborative editing systems and LibreSource and Moodle.