

REUSING IMS-LD FORMALIZED BEST PRACTICES IN COLLABORATIVE LEARNING STRUCTURING

Davinia Hernández-Leo¹, Juan I. Asensio-Pérez¹, Yannis Dimitriadis¹,
Miguel L. Bote-Lorenzo¹, Iván M. Jorrín-Abellán², Eloy D. Villasclaras-Fernández¹

¹School of Telecommunications Engineering, University of Valladolid

²Faculty of Education, University of Valladolid

Campus Miguel Delibes s/n, 47011 Valladolid, Spain

{davher@tel, juaase@tel, yannis@tel, migbot@tel, ivanjo@doe, evilfer@ulises.tel}.uva.es

Abstract

Designs of CSCL (Computer Supported Collaborative Learning) activities should be flexible, effective and customizable to particular learning situations. On the other hand, structured designs aim to create favourable conditions for learning. Thus, this paper proposes the collection of representative and broadly accepted (best practices) structuring techniques in collaborative learning. With the aim of establishing a conceptual common ground among collaborative learning practitioners and software developers, and reusing the expertise that best practices represent, the paper also proposes the formulation of these techniques as patterns: the so-called CLFPs (Collaborative Learning Flow Patterns). To formalize these patterns, we have chosen the educational modelling language IMS Learning Design (IMS-LD). IMS-LD has the capability to specify many of the collaborative characteristics of the CLFPs. Nevertheless, the language bears limited capability for describing the services that mediate interactions within a learning activity and the specification of temporal or rotated roles. This analysis is discussed in the paper, as well as our approaches towards the development of a system capable of integrating tools using IMS-LD scripts and a CLFP-based Learning Design authoring tool.

Key Words

Best practice, CSCL, pattern, script, IMS-LD

1. Introduction

In the last decade, the progress of the research, development and education practice in the Computer Supported Collaborative Learning (CSCL) domain has been remarkable [1]. CSCL systems reflect the importance of social interactions as an essential element of learning [2], as well as the role of participatory design of the potential actors that are involved in collaborative learning situations [3].

Nevertheless, there is little advance yet in the development of CSCL systems capable of supporting effective, reusable, flexible and customizable

collaborative learning activities designs. In this point, the problem that arises is actually twofold: designing collaborative learning activities and developing CSCL systems that implement them. However, solutions to these problems are not trivial.

First of all, since CSCL applications are developed by technologists, it is needed a collaborative learning domain knowledge from a technological viewpoint [4]. Besides, the requirements posed by educators, participants and the particularities of their educational context are highly dynamic. Thus, designs of collaborative learning activities should be flexible and adaptable to particular learning situations. Moreover, users of CSCL systems are not usually technical experts, so they should be provided with suitable authoring tools that avoid the need of technical knowledge. On the other hand, potential users of CSCL are not necessarily experienced in collaborative learning; hence reusing successful activity designs would be interesting.

P. Dillenbourg in [5] affirms that unfocused activities and free collaboration does not systematically lead to learning outcomes. Therefore, structured designs aim to create favorable conditions for learning. An approach of structuring consists of CSCL systems that support the learners by encouraging them to interact according to a collaboration script [6]. A script describes a sequence of activities (collaborative or not), how students should form groups and how they are expected to collaborate to solve a problem.

In order to enhance collaborative learning designs in CSCL, this paper proposes the formalization of representative and broadly accepted structuring techniques in collaborative learning in such a way that CSCL systems could reuse, particularize and customize these best practices according to the requirements of a concrete learning situation. The approach that has been adopted by the authors [4] consists in formulating these best practices as patterns: CLFPs or Collaborative Learning Flow Patterns define effective sequences of collaborative learning activities that can be easily reused and communicated to others. However, the original description of CLFPs is based on natural language due to

the fact that they are proposed by non-technical people. That implies that software tools cannot automatically process CLFP definitions. The standard formalization of the CLFPs using IMS Learning Design (IMS-LD) [7] specification provides a promising way of introducing effective design techniques in CSCL systems and eases integration and reuse. Through the formalization of the collaborative learning expertise captured in CLFPs [8] and some particular learning situations structured according to several CLFPs, the paper illustrates and evaluates the support of IMS-LD for key CSCL requirements.

In order to apply and evaluate the above concepts in real computer-based educational settings, CSCL systems capable of interpreting IMS-LD language and Learning-Design authoring tools should be developed. Currently there are several proposals of IMS-LD authoring tools and IMS-LD players, such as [9,10,11,12] and [13,14] respectively. We propose a system that includes a CLFP-based Learning Design authoring tool and a system that integrates tools using a CLFP-based Learning Design script [15].

Hence, this paper is structured as follows: section 2 is devoted to the problem of designing collaborative activities in CSCL systems. Section 3 illustrates some partial examples of CLFP-based Learning Designs, pointing out specific capabilities and limitations of IMS-LD for CSCL. Our proposals for IMS-LD authoring and interpreting systems are exposed in section 4. Finally, section 5 concludes this document and indicates our future work.

2. Structuring Collaboration in CSCL

Structuring the collaborative learning process in order to favour productive interactions enhances the effectiveness of collaborative learning activities. According to [6], there are three different types of systems that may be used to structure collaboration. The first type of systems takes advantage of natural communication tool affordances. However, the free use of communication tools does not necessarily produce learning. Another type of structuring systems includes tools deliberately designed to structure collaboration (e.g. structured dialogue interfaces). A third type refers to the use of collaboration scripts by a Learning Management System (LMS). A collaboration script is a set of instructions regarding to how the group members should interact, how they should collaborate and how they should solve the problem [5].

The structuring approaches on which the paper is focused are related to the idea of scripts, i.e., the description of general flows of collaborative (or not) learning activities. *Learning flow* or *learnflow* are terms used in the learning domain in a similar way to (from a collaboration point of view) the CSCW (Computer

Supported Cooperative Work) workflow or activity-level coordination [16].

2.1 Collaborative Learning Flow Patterns

Collaborative Learning Flow Patterns (CLFPs) represent broadly accepted techniques that are repetitively used by collaborative learning practitioners (e.g. teachers) when structuring the flow of types of learning activities involved in collaborative learning scenarios. Thus, CLFPs can be understood as a way of collecting “best practices” in collaborative learning. These best practices refer to ways of arranging participants in collaborative learning sessions, sequencing types of collaborative learning activities, etc. CLFPs are *patterns* because they provide a recurrent solution to a recurrent problem: what flow and type of activities are the most suitable for promoting the achievement of a set of desired educational objectives. CLFPs are represented according to a structure proposed in [4], which is based on the use of natural language.

The identification, collection, and formulation of collaborative learning best practices as CLFPs imply several potential advantages [4]:

- They provide a **way of communicating collaborative learning expertise** to other (eventually novice) practitioners: instead of trying to create their own collaborative scripts from scratch, practitioners can use CLFPs as a help or a guide for structuring their own collaborative scenarios.
- They provide a **conceptual common ground** among collaborative learning practitioners and software developers: the structuring and sequencing of learning activities provide useful information to software developers when identifying requirements for CSCL tools capable of supporting collaborative learning scenarios based on a particular CLFP.
- They promote **software reuse**: software developers can identify what type of CSCL tools could be needed in order to support collaborative learning scenarios compliant with the same CLFP. Besides, a subset of those tools could potentially be reused in the support of several of those scenarios.

As a consequence of all the above ideas, CLFPs motivate the envisioning of a new kind of CSCL tools: **CLFPs-based collaborative Learning Design authoring tools**. This tool would guide educators to obtain effective collaborative learning scripts for their specific learning situations. The selection of a CLFP as the basis for the desired design would guarantee (to a great extent) the achievement of a set of objectives (as dictated by previous experiences from which the CLFP emerged).

Fig. 1 and Fig. 2 show respectively the UML activity diagrams of TAPPS (Thinking Aloud Pair Problem Solving) and Pyramid CLFPs. The diagrams have been

elaborated by technicians from two examples of best practices in collaborative learning structuring provided in natural language by practitioners [17]. In TAPPS CLFP students are paired and given a series of problems. The two students are assigned specific roles that switch with each problem: *problem solver* and *listener*. The problem solver explains her/his solution to the problem. The listener follows the explanation and catches any errors that occur.

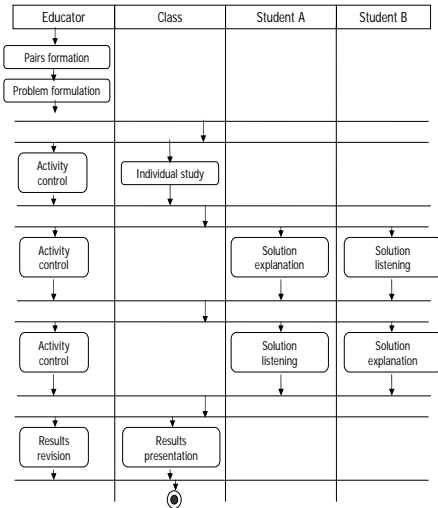


Figure 1. Activity Diagrams of TAPPS CLFP

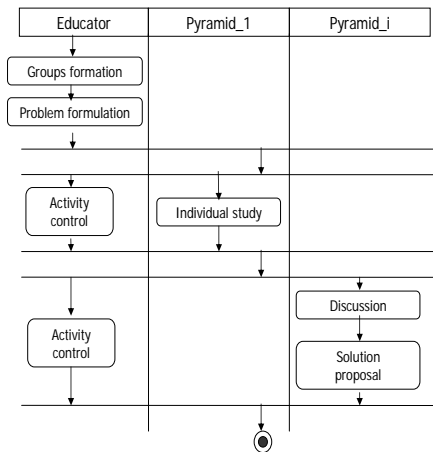


Figure 2. Activity Diagrams of Pyramid CLFP

On the other hand, in the Pyramid CLFP (see Fig. 2) each individual participant (or initial group, participants in the first level of the pyramid “Pyramid_1”) studies the problem and proposes a solution. Then, groups of participants compare and discuss their proposals and, finally, propose a new shared solution (second level of the pyramid, $i=2$ “Pyramid_2”). Those groups join

successively in larger groups (“Pyramid_i”) in order to generate new agreed proposals. At the end ($i=N$), all the participants must propose a final agreed solution.

Other examples are the well-known Brainstorming or the Jigsaw CLFPs [8]. The Jigsaw proposes to divide a problem into sub-problems, one for each group member. The students with the same sub-problem, after an individual study of their section, join together and form a new group (expert group) in order to discuss the concepts in their section. Then, they join their original group (jigsaw group) with the aim of finding a solution to the global problem.

The educational benefits of the CLFPs are significant. For example, Jigsaw CLFP promotes positive interdependence (team members need each other to succeed) and ensures individual accountability (students must contribute their fair share). TAPPS CLFP encourages problem-solving skills, permits students to rehearse the concepts, relate them to existing frameworks, and produce a deeper understanding of the material [17].

It is not actually realistic to consider that collaborative learning scenarios are always structured as said by a unique CLFP. Often the idea of *CLFP hierarchies* is applied: a learning scenario might be designed according to several CLFPs in different levels. A CLFP-based scenario could have one or several of its activities based on another CLFP. It is possible to design, for instance, the problem discussion of the jigsaw group according to the TAPPS CLFP, where each expert acts successively as a problem solver, or the sub-problem discussion as indicated by the Brainstorming CLFP. Or, simply, a collaborative learning scenario may adopt separate sequenced CLFPs, i.e., the first half of a learning situation might be organized according to one Collaborative Learning Flow Pattern, and the second one according to other CLFP.

2.2 IMS Learning Design

With the related aim of describing the sequencing of learning activities, IMS-LD specification provides a standard language for formally expressing scenarios based on different pedagogical theories [7]. A *Learning Design* is a description of a *method* enabling learners to attain particular *objectives* by performing *learning activities* in a certain order in the context of a *learning environment*. The environment consists of the appropriate *learning objects* and *services* to be used during the performance of the activities. The Learning Design is preferably integrated into an IMS Content Package [18] to create a so-called *Unit of Learning*.

Describing collaborative learning scenarios using IMS-LD is feasible [8]. It enables the design of processes that include several *roles*, each of which can be played by several people. A collaborative learning experience can

be described by associating multiple people and/or multiple roles to the same learning activity. Furthermore, IMS-LD enables their activities to be specified in coordinated learning flows.

Nevertheless, IMS-LD has also some drawbacks in reflecting collaborative learning experiences [8,19,20]. For instance, IMS-LD provides no means to specify how the members of a group collaborate within each learning activity. It only states that if multiple individuals are to collaborate or work together at the same time, this has to be done through a *service* in their assigned environment which supports this collaborative capability [7]. Thus, we propose in [8] an extension of the IMS-LD service definition consisting of the definition of a special type of service, called *groupservice*.

2.3 IMS-LD for CLFPs

It has been already discussed that CLFPs and IMS-LD have similar purposes in the sense that both aim to describe learning processes. However, CLFPs represent natural-language best practices and IMS-LD is merely a formal language. Formalizing the CLFPs using IMS-LD specification (and the proposed extension [8]) provides the following advantages:

- Software tools could **automatically process** CLFP definitions.
- It facilitates the **introduction of design techniques in CSCL systems**.
- CLFPs can be **particularized** and **customized** to particular learning situations of different disciplines following the three-stage process proposed in [8]:
 - 1) IMS-LD description of the CLFP
 - 2) CLFP-based Learning Design (particularization depending on the learning scenario)
 - 3) CLFP-based Unit of Learning (packaging with particular resources)
- Learning **scenarios structures are dissociated from the learning resources** (content, tools). Therefore, resources can be reused within different scenarios structures (Learning Designs). In the same way, Learning Designs or CLFPs can be reused when particularized with different resources that depend on the concrete learning scenario (Units of Learning).
- Since the specification provides a standard language, the **reuse** and **integration** of CLFPs or CLFP-based Learning Designs in different systems are facilitated.

3. Describing CLFP-based Learning Designs

Summarizing the ideas exposed in the previous session regarding the formalization of CLFPs and its reuse as a result of their particularization to create complete Learning Designs, the following definition is proposed: A

CLFP-based Learning Design is the result of particularizing and customizing the IMS-LD description of a best practice in collaborative learning activities structuring according to the requirements and conditions of a particular learning scenario.

This section illustrates, through partial IMS-LD description of CLFPs and CLFP-based Learning Designs, specific capabilities and detected limitations of IMS-LD for describing some key CSCL aspects, such as coordination of activities, definition of roles and groups and specification of learning flows and CSCL tools.

3.1 Pyramid CLFP-based LD: Groups Definition

The first question when describing collaborative learning experiences using IMS-LD is probably how to specify groups. Fig. 3 shows the subtle way in which the groups defined by the Pyramid CLFP can be specified with IMS-LD. Left part of the figure (a) refers to the IMS-LD formalization of the Pyramid CLFP roles. “Pyramid_N” is the role that will be played by the participants in the last (N) level of the pyramid. “Role Pyramid_i” refers to the first and intermediate levels of the Pyramid CLFP (i ranges from 0 to (N-1)). Right part illustrates a possible result of the dynamic role assignment while interpreting (but before running) a Pyramid CLFP-based Unit of Learning.

The “Pyramid_N” role should be played by several individuals (minimum of 2 persons, *min-person*="2"), which actually form the largest *group* of the Pyramid CLFP. Various pyramid groups can be dynamically created (*created-new*="allowed" determines that multiple instances of a particular role are allowed [7]). When a new instance of a role is created, the new instance is the parent of its defined sub-roles. Each individual in a “Pyramid_i” group can be bound exclusively to one “Pyramid_i-1” sub-role (*match-persons*="exclusively in roles"). A participant bound to the instance (1) of the “Pyramid_1” role cannot be bound to any other instance of the same role and is also bound to instance (1) of “Pyramid_2” role and to “Pyramid_3” role (see Fig. 3). Note that if a unique individual will be bound to each role in the first level of the Pyramid (Pyramid_1), these roles are not strictly necessary. In this case, “Pyramid_N” and “Pyramid_1” are equivalent. “Pyramid_N” represents a group or an individual depending on the environment associated to the activity that this role plays in a particular moment (*act*), i.e. whether the service included in the environment is collaborative or not (see next subsection 3.2).

Concluding, groups can be formed with IMS-LD by binding multiple individuals to the same (instance of a) role or by associating multiple roles to activities that provide a shared environment that mediates collaboration.

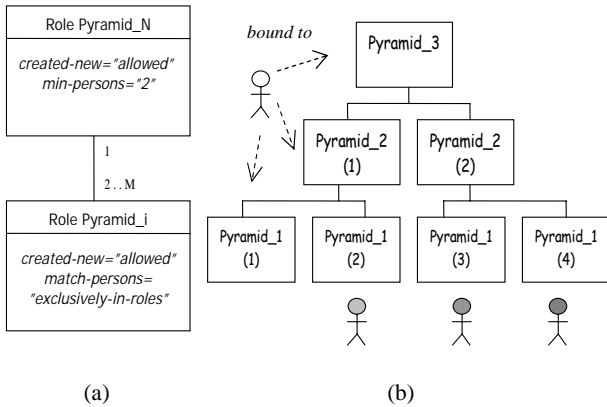


Figure 3. IMS-LD formalization of the Pyramid CLFP groups (a) and example of a result of a dynamic role assignment of a CLFP-based Unit of Learning (b)

3.2 Jigsaw CLFP-based LD: Learning Flow and CSCL Tools

The activity-level coordination of a collaborative scenario [16], i.e., the learning flow defined by the Jigsaw CLFP for example, can be expressed in the IMS-LD *method* (see Fig. 4). A method contains one or more *plays*, which are modelled according to a theatrical play with *acts* and *role-parts*. These plays run in parallel, independent from each other. Acts determine whether, when, and for what roles an activity and resources are to be used [7].

In addition, Fig. 5 partially illustrates the IMS-LD description of the Jigsaw CLFP: the different groups of the CLFP and the educator can be modelled with the IMS-LD element *role* (as explained in subsection 3.1), and the learning flow is defined using several *acts* as it has been already mentioned (note that synchronization points in the activity diagram determine the boundaries of the acts, see also Fig.4).

Method			
Play	Act	Role	Role-part assigned
Jigsaw-CLFP	1.1	Educator	Activity-structure: <i>jigsaw-organization</i>
	Complete act when jigsaw is organized		
	2.1	Expert	Activity-structure: <i>subproblem</i>
	2.2	Educator	Support-activity: <i>activity-control</i>
Complete act when a time limit has passed			
Jigsaw	3.1	Jigsaw	Activity-structure: <i>global-problem</i>
	3.2	Educator	Support-activity: <i>activity-control</i>
Complete act when a time limit has passed			
Complete play when last act has been completed			
Complete method when play Jigsaw-CLFP completed			

Figure 4. Example of the Jigsaw CLFP Learning Flow Definition in IMS-LD: the Method

Furthermore, Fig. 5 shows the kind of aspects that should be determined in order to particularize and customize a CLFP and achieve then a CLFP-based Learning Design. These aspects are, for instance, what are the problem and subproblems that are to be solved using a Jigsaw CLFP-based learning situation (e.g. collaborative design of a computing system, each subsystem is assigned to each expert); and which tools will be used during the discussions (e.g. discussion forum and collaborative conceptual-map tool). Here, the necessity of specifying collaborative learning activities that use tools that mediate collaboration appears.

Therefore, how to specify collaborative learning activities is another key issue. A group-based activity can be described by associating a role played by several people or/and multiple roles to learning activities that provides at least one tool (IMS-LD *service*) that mediates collaboration. IMS-LD only defines two basic (to some extent) collaborative services: e-mail and conferencing. It is clear that collaborative learning scenarios require more services. That is the case, for instance, of the scenario proposed in [15], which is to be applied in a course on *Computer Architecture* at our University. The scenario is based on the Jigsaw CLFP and includes an activity in which students need a *collaborative task assignment tool* in order to distribute different machines among them, so that each student benchmarks a group of machines. IMS-LD does not provide any means to specify this collaborative tool. The IMS-LD service extension for collaborative services proposed in [8] is a solution to this problem. The possibility of specifying these services and its characteristics enlarge the set of collaborative learning activities that can be described using IMS-LD.

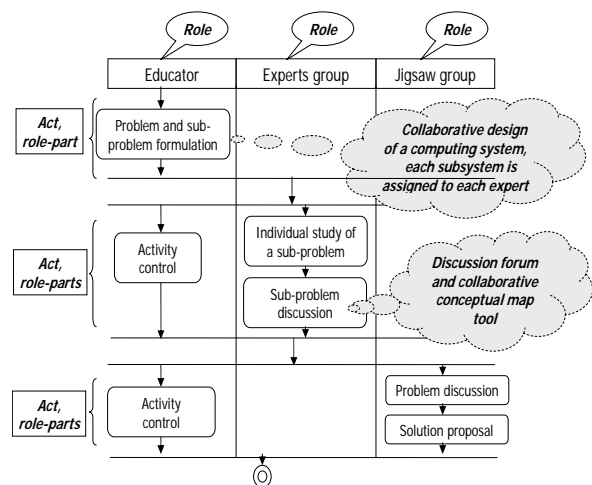


Figure 5. IMS-LD description and customization of the Jigsaw CLFP

3.3 TAPPS CLFP-based LD: Dynamic Roles

It is noteworthy that in collaborative learning scenarios it is often possible to identify different roles depending on their variability (static and dynamic) [21]. The roles that explicitly can be defined with IMS-LD are static roles that remain invariable during the whole Learning Design. Rotation of roles or temporal roles (dynamic roles) cannot be explicitly specified using IMS-LD (although it might be possible to support them if the runtime provides a mechanism that allows roles switching [7]).

Fig. 6 shows a way of implicitly describing with IMS-LD the rotation of the TAPPS-CLFP roles: problem solver and listener. The generic roles “Student-A” and “Student-B” are alternatively related to the explanation and listening activities within each act (that is the reason why in Fig. 1 these generic roles were used instead of the natural roles of the TAPPS CLFP: *problem solver* and *listener*). These activities are connected through a conferencing service. Nevertheless, it is not clear to which extent this could be always a solution to the specification of dynamic roles: A possible example is when in the Jigsaw-CLFP sub-problem activity, experts use simultaneously a discussion forum where one of the experts is a moderator and the rest are participants, and a collaborative conceptual map tool where one of the experts is the leader writer and the others are annotators. Here the solution may be to specify these temporal roles within the service. Moreover, it is not difficult to imagine scenarios in which definition of temporal roles are also needed within an activity.

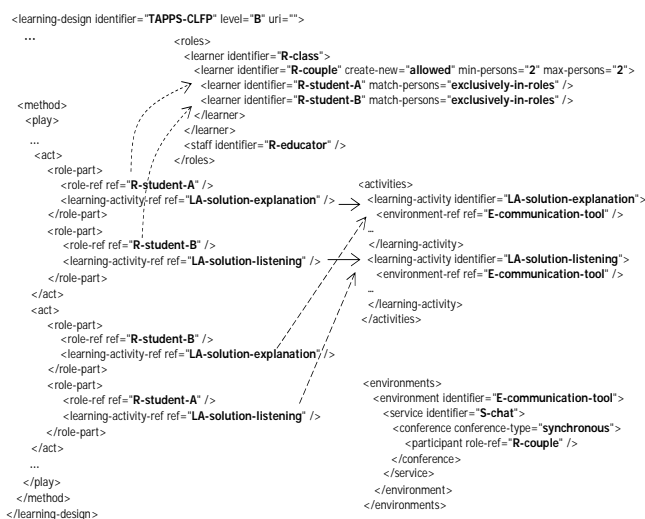


Figure 6. Excerpt of a TAPPS CLFP-based Learning Design (Arrows point referenced element definitions)

4. Implementation Approach

In order to validate all the proposals that have been introduced in the previous sections, the authors are currently involved in the development and testing a CSCL system in which CLFP-based IMS-LD scripts play a central role. That CSCL system, whose structure is depicted in Fig. 7, has two main components: a CLFPs-based Learning Design authoring tool (see section 2.1) capable of guiding collaborative learning designers in the process of creating their own IMS-LD based collaboration scripts by starting from existing CLFPs; and *Gridcole*, a tailorable collaborative learning system capable of interpreting IMS-LD collaboration scripts and setting up the technological environment (based on the so-called *grid services* [22] that follow Service-Oriented Computing paradigm [23]) needed to support all the collaborative learning activities included in the script. Both components, which separately face the design and the enactment problems respectively, are detailed in the following subsections.

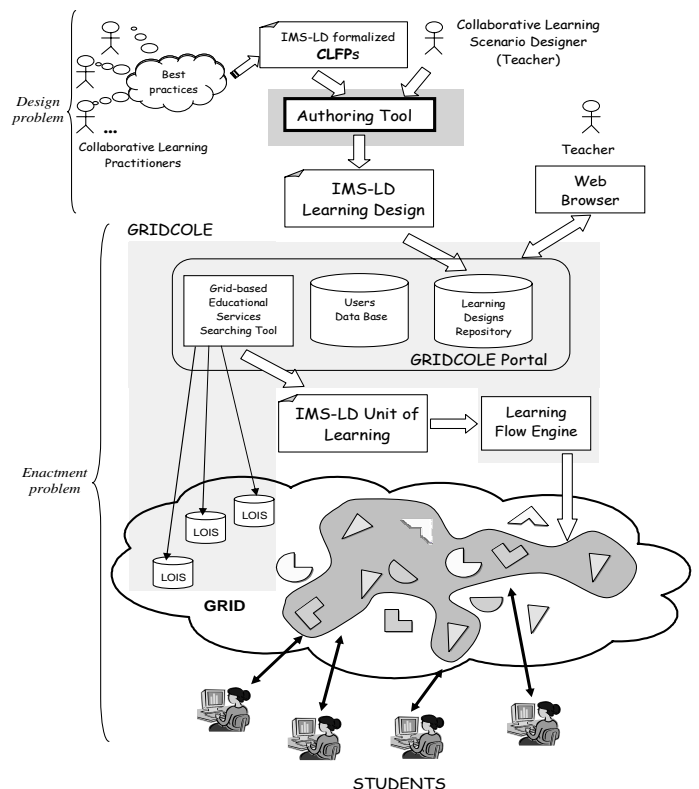


Figure 7. Schema of the implementation approach adopted for validating the use of IMS-LD and CLFPs (design and enactment problems are treated separately)

4.1 CLFP-based Learning Design Authoring Tool

In order for educators to become directly involved in authoring Learning Designs and Units of Learning, IMS-LD authoring tools are needed. Three possible classifications of IMS-LD authoring tools are the following. Two classifications are open source vs. commercial implementations and general vs. specialized (support for specific learning theories) editors. In addition, authoring tools can be close to the specification or can hide the syntax and terminology of IMS-LD. Two significant authoring tool proposals are: RELOAD LD editor [10], a general open source authoring tool which is close to the specification, and, LAMS [9], an easy-to-use commercial authoring tool which is IMS-LD “inspired” (not at present IMS-LD compliant) and which allows to choreograph a whole learning scenario around a chosen topic according to a sequence of activities created from a template called “What is Greatness (in a human being)”.

As it has been mentioned, our proposal is a CLFP-based IMS-LD authoring tool: An editor that will allow educators to design effective IMS-LD-compliant collaborative learning scenarios that will be structured according to one or several CLFPs (CLFP hierarchies, see section 2.1). We may affirm that our approach is a collaborative-learning specialized IMS-LD authoring tool in between LAMS and RELOAD LD editor.

The tool would enable the selection of CLFPs using CLFP metadata, which includes: CLFP objectives regarding cognitive skills (e.g. TAPPS CLFP encourages analytical reasoning skills), types of problems that can be solved (e.g. Brainstorming CLFP generates a large number of ideas in a short period of time), complexity and collaborative learning experience needed (e.g. Jigsaw CLFP is complex and is probably more appropriate for experienced students or educators) and advice (e.g. Students could not contribute if they cannot generate an idea in the brainstorming, but number of times they are allowed not to contribute may be limited).

Following the three-stage process mentioned in subsection 2.2 (explained in detail in [8]), the authoring tool would guide the educator using the IMS-LD formalization of the selected CLFP(s) in order to achieve a customized CLFP-based Learning Design for a particular scenario. A CLFP-based Unit of Learning would consist of a CLFP-based Learning Design and a set of actual resources that depend on a particular learning situation. Any IMS-LD player could eventually interpret these Units of Learning (e.g. Coppercore [13]) or these Learning Designs (e.g. Gridcole).

4.2 Tools Integration Using IMS-LD Scripts

Gridcole [15] is a collaborative learning system that can be easily tailored in order to support collaborative learning scenarios described in the form of IMS-LD documents. These documents, which will eventually be generated using the CLFP-based authoring tool described above, must include a generic description of every tool that makes up the collaborative environment required for the realization of each activity.

Our system provides a repository (see Fig. 7) in which Learning Designs can be stored for later retrieval and use. Educators can then choose one of the existing Designs so that Gridcole looks for suitable tools according to the descriptions included in the selected document. Searches are performed in an educational grid in which third-party providers supply a large pool of tools exposed as grid services. Significantly, the use of grid technology enables access to all kinds of tools that may be employed for educational purposes. Examples of tools that may be found in an educational grid include collaborative editors, discussion forums, structured chats, 3D virtual environments, remote laboratories of electronics, etc. (Note that Gridcole also enables the use of tools with supercomputing or specific hardware requirements that can be integrated in collaborative environments if desired.)

Once this operation is completed, the Learning Design is packaged in an IMS Unit of Learning along with the references to the tool grid services that will support its realization. All this information is employed by Gridcole learning flow engine at the time of running the Unit in order to determine the sequence of activities that must be performed by each participant as well as to provide students with customized environments to support each activity. These environments integrate the tool grid services referenced in the Unit of Learning.

It is noteworthy that the authors have already developed a prototype of Gridcole system. The main limitations of this prototype concern search facilities and script interpretation. Nevertheless, the prototype has been employed to support a short version of the collaborative learning scenario for the *Computer Architecture* course mentioned in subsection 3.2 and explained deeply in [15].

5. Conclusions and Future Work

CLFPs capture expert practice in collaborative learning strategies with regard to structuring collaboration. These patterns suggest rather than prescribe a solution. They could be understood as incomplete scripts that represent best educational practices and that offer guidance but require customization. The paper has pointed out the advantages of the CLFPs as well as the lessons learned from its formalization using IMS-LD.

IMS-LD does support the description of the aspects of CLFPs (group formation, learning flows). However, it is necessary to warn about some detected limitations of the specification for describing dynamic roles (rotation of roles, temporal roles) and for specifying learning activities involving groups which require particular tools that support collaboration (collaborative task assignment tool, collaborative editor, etc.). Nevertheless, a more exhaustive analysis should be accomplished in order to generalize these conclusions for all the possible patterns in collaborative learning structuring.

Therefore, future work includes also additional validation of the CLFP approach, the analysis of the flexibility of its formalization using IMS-LD and the study of the trade-offs among instructional design and open-dynamic learning. In this sense, a short-term activity under way is related to the formalization of CLFPs hierarchies.

In order to achieve these objectives, it is necessary to apply the ideas exposed in the paper in real settings. With this purpose, we are developing a system that consists of an authoring tool that uses CLFPs as guides for educators to obtain effective collaborative Learning Designs (scripts) customized according to their specific learning situations, and a system that includes a learning flow engine that integrates tools, which are grid services, using a CLFP-based IMS-LD script. Nevertheless, the implementation in real settings of computer-supported scripted CLFP-based situations is not trivial. This CSCL system facilitates the process of implementing CLFPs; but more research is necessary to systematically analyze and evaluate this problem.

Some open questions to explore are how to manage unexpected situations during the running of a script: Could they be considered beforehand in the Learning Design? Could the LD be changed during run-time? Are the scaffolding and awareness CSCL necessities envisaged in the Designs?

With regard to the formulation of best practices in collaborative learning as patterns for CSCL, the authors are currently involved in the TELL project [24], in which the patterns are being identified following a bottom-up approach, i.e., using as a starting point real case studies.

Acknowledgements

This work has been partially funded by European Commission e-Learning TELL Project EAC/61/03/GR009, Kaleidoscope Network of Excellence (contract 507838) and Spanish Ministry of Science and Technology TIC-2002-04258-C03-02. The authors would also like to acknowledge the contributions from other members of the GSIC/EMIC Group (Intelligent & Cooperative Systems Research Group / Education, Media, Information, and Culture).

References

- [1] A. Dimitracopoulou & A. Petrou, Advanced collaborative distance learning systems for young students: Design issues and current trends on new cognitive and metacognitive tools, *THEMES in Education International Journal*, 2005, (in press).
- [2] P. Dillenbourg, *Collaborative learning: Cognitive and computational approaches* (Oxford, UK: Elsevier Science, 1999).
- [3] M. Muller & S. Kuhn, Participatory Design, *Communications of the ACM*, 36(4), 1993, 25-28.
- [4] Y. Dimitriadis, J. I. Asensio-Pérez, A. Martínez-Monés, & C. A. Osuna-Gómez, Component based software engineering and CSCL: Component identification and dimensioning, *Upgrade (digital journal of European Professional Informatics Societies), special issue on e-learning: Boarderless education*, 4(5), 2003, 21-28.
- [5] P. Dillenbourg, Over-Scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Inaugural Address, Three Worlds of CSCL. Can We Support CSCL?* (Heerlen: Open Universiteit Nederland, 2002).
- [6] P. Jermann, A. Soller, & A. Lesgold, Computer software support for CSCL. In P. Dillenbourg (Ed.), *What we know about CSCL and implementing it in higher education* (Norwell, MA, USA: Kluwer Academic Publishers, 2004) 141-166.
- [7] IMS Learning Design v1.0 Final Specification, <http://www.imsglobal.org/learningdesign>, last visited: 2004.
- [8] D. Hernández-Leo, J. I. Asensio-Pérez, & Y. Dimitriadis, IMS Learning Design support for the formalization of Collaborative Learning Patterns, *Proc. 4th International Conference on Advanced Learning Technologies*, Joensuu, Finland, 2004, 350-354.
- [9] J. Dalziel, *Implementing Learning Design: The Learning Activity Management System (LAMS)*, 2003, whitepaper, accessible in <http://www.lamsinternational.com/documents/ASCILITE2003.Dalziel.Final.pdf>.
- [10] RELOAD, Reusable eLearning Object Authoring

- & Delivery project website, <http://www.reload.ac.uk/>, last visited: 2004.
- [11] P. Karampiperis & D. Sampson, Designing learning services for open learning systems utilizing Learning Design, *Proc. 4th IASTED International Conference Web-based Education*, Grindelwald, Switzerland, 2005, 279-284.
- [12] CopperAuthor project website, <http://www.copperauthor.org/>, last visited: 2005.
- [13] Coppercore project website, <http://www.coppercore.org/>, last visited: 2004.
- [14] T. Vantroys & Y. Peter, COW, a flexible platform for the enactment of learning scenarios, *Proc. 9th International Workshop on Groupware*, Grenoble, France, 2003, 168-182. (Lecture Notes in Computer Science, 2806)
- [15] M. L. Bote-Lorenzo, L. M. Vaquero-González, G. Vega-Gorgojo, Y. Dimitriadis, J. I. Asensio-Pérez, E. Gómez-Sánchez, & D. Hernández-Leo, A Tailorable collaborative learning system that combines OGSA grid services and IMS-LD scripting, *Proc. 10th International Workshop on Groupware*, San Carlos, Costa Rica, 2004, 305-321. (Lecture Notes in Computer Science, 3198)
- [16] C. A. Ellis & J. Wainer, A Conceptual Model of Groupware, *Proc. 5th Conference on Computer Supported Cooperative Work*, Chapel Hill, North Carolina, USA, 1994.
- [17] Doing CL website, <http://www.wcer.wisc.edu/nise/CL1/CL/doingcl/DCL1.asp>, last visited: 2004.
- [18] IMS Content Packaging v1.1.4 Final Specification, <http://www.imsglobal.org/content/packaging>, last visited: 2004.
- [19] M. Caeiro-Rodríguez, L. Anido-Rifón, & M. Llamas-Nistal, Towards IMS-LD extensions to actually support heterogeneous learning designs. A pattern-based approach, *Proc. 4th International Conference on Advanced Learning Technologies*, Joensuu, Finland, 2004, 565-569.
- [20] Y. Miao, K. Hoeksema, H. U. Hoppe, & A. Harrer, CSCL Scripts: Modelling Features and Potential Use, *Proc. Conference on Computer Supported Collaborative Learning*, Taipei, Taiwan, 2005, (in press).
- [21] J. A. Marcos-García, A. Martínez-Monés, & Y. Dimitriadis, The role of roles in the analysis of interactions in collaborative environments, *Proc. 16th European Conference on Artificial Intelligence, workshop: Artificial Intelligence in Computer Supported Collaborative Learning*, Valencia, Spain, 2004, 23-35.
- [22] I. Foster, C. Kesselman, J. M. Nick, & S. Tuecke, Grid services for distributed system integration, *Computer*, 35(6), 2002, 37-46.
- [23] M. P. Papazoglou & D. Georgakopoulos, Service-Oriented Computing, *Communications of the ACM*, 46(10), 2003, 25-28.
- [24] TELL project website, <http://cosy.ted.unipi.gr/tell/>, last visited: 2005.

Biographies



Davinia Hernández-Leo received her M.S. degree in telecommunications engineering from University of Valladolid, Spain, in 2003. She is currently studying towards a PhD degree in telecommunications engineering at the University of Valladolid. She is a teaching assistant at the

Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. Her research interests include CSCL and modelling techniques applied to the educational field.



Juan Ignacio Asensio-Pérez received the M.S. and the PhD degrees in telecommunications engineering from University of Valladolid, Spain, in 1995 and 2000, respectively. He is currently an Associate Professor at the Department of Signal Theory, Communications and Telematics Engineering, University of

Valladolid. His research interests include distributed systems and, particularly, distributed CSCL applications and integrated systems and network management.



Yannis A. Dimitriadis received the engineering degree from the National Technical University of Athens, Greece (1981), the M.S. from the University of Virginia (1983) and two PhD degrees from the University of Valladolid, Spain (1992 and 1995), both in telecommunications engineering.

He is currently an Associate Professor at the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. His research interests include computer supported collaborative learning and distributed systems.



Eloy D. Villasclaras-Fernández is currently studying towards a M.S. degree in telecommunications engineering at the University of Valladolid. He is working on the development of a CSCL authoring tool for his final project at the Department of Signal Theory, Communications

and Telematics Engineering, University of Valladolid. He is interested in distributed systems, particularly applied to CSCL.



Miguel Luis Bote-Lorenzo received his M.S. degree in telecommunications engineering from University of Valladolid, Spain, in 2001. He is currently studying towards a PhD degree in telecommunications engineering at the University of Valladolid. He is a teaching assistant at the Department of Signal Theory,

Communications and Telematics Engineering, University of Valladolid. His research interests include CSCL systems and grid computing.



Iván M. Jorrín Abellán received his M.A. degree in Psychology and Pedagogy from the University of Valladolid, Spain, in 2001. He is currently studying towards a PhD degree in Pedagogy at the University of Valladolid. He is teaching assistant at the same university. His current research efforts are

devoted to the study of the educational implications of computer supported collaborative learning.