Chapter XX
Diagrams of Learning Flow Patterns’ Solutions as Visual Representations of Refinable IMS Learning Design Templates

Davinia Hernández-Leo
University of Valladolid, Spain

Eloy D. Villasclaras-Fernández
University of Valladolid, Spain

Juan I. Asensio-Pérez
University of Valladolid, Spain

Yannis Dimitriadis
University of Valladolid, Spain

ABSTRACT

This chapter introduces the use of diagrammatic representations of learning flow patterns as a means of visualizing refinable IMS Learning Design (IMS LD) templates. It argues that the incorporation of pattern-based IMS LD templates in authoring tools, which graphically guide users to create their own learning designs, offers a solution to the problem of IMS LD constructs not being familiar to educators because of its technical nature and text-based notation. Furthermore, this solution facilitates the reuse of good practices formulated as patterns, permitting a design process that promotes potentially effective results. This issue is especially important in collaborative learning designs, in which elicitations of desired social interactions are planned beforehand. Based on these ideas, the chapter also presents Collage, an IMS LD editor which provides templates based on collaborative learning flow patterns (CLFPs), and includes an example drawn from a real scenario that shows the feasibility and usefulness of the approach.
INTRODUCTION

Learning Technology standards and specifications are used increasingly to support the professional activity of instructional designers (Rodríguez-Estévez, Caeiro-Rodríguez, & Santos-Gago, 2003). Significantly, the recent IMS Learning Design (IMS LD) specification (IMS, 2003) highlights the use of computers to facilitate the teaching-learning processes instead of for delivering educational content (Burgos & Griffiths, 2005). That is to say, IMS LD provides a machine-readable notation (XML language) to formalize learning scenarios that can be automatically interpreted by compliant players or Learning Management Systems (LMS) (Koper & Tattersall, 2005). These formalized scenarios or Units of Learning (UoLs) are abstract representations of any individual or group-based learning situation (a course, a lesson, etc.). The main benefits of using a specification like IMS LD are twofold. First, teachers, instructors and trainers can specify the behavior and functionality of a computer-supported learning system by providing a UoL (Bote-Lorenzo et al., 2004). And, second, because UoLs are interoperable and reusable, they can be executed in various compliant systems and in various settings with different participants (Tattersall et al., 2005).

However, as pointed out in Chapter 15, IMS LD is a notation system that is layered, formal, textual and of a single perspective (Waters & Gibbons, 2004; Botturi, Derntl, Boot, & Figl, 2006). Due to its textual and formal nature, IMS LD does not look familiar (i.e., cannot be understood intuitively) by the majority of educators, as argued by Harrer & Malzah (2006) and in Chapter 14 of this handbook. For this reason IMS LD is not really practical as a teacher-friendly design and authoring approach for UoLs, nor is it useful as a means of communication in participatory design teams involving different stakeholders (Griffiths & Blat, 2005). To overcome these drawbacks, visual design languages and tools are envisaged as a solution for the reflective communication and creative generation of designs (see Chapter 21).

This chapter focuses on the design of IMS LD-formalized collaborative learning scenarios (or scripts) (Hernández-Leo, Asensio-Pérez, & Dimitriadis, 2005). Collaborative learning is a relevant educational approach, whose learning benefits are positively recognized by practitioners (Johnson & Johnson, 1999). This approach highlights the importance of social interaction and active learning processes. Computer-supported collaborative learning (CSCL) is the domain that studies the use of computers and networks as mediation technologies to support collaborative learning (Dillenbourg, 1999). It is worth noticing that the context dependency and social variables involved in CSCL make this domain particularly complex. This fact motivates the agreement of many authors on that CSCL settings need more flexibility (or less rigidity) than traditional, individual instructional sequences (Karagiorgi & Symeou, 2005; Dillenbourg, 2002; Oliver, Harper, Hedberg, Wills, & Agostinho, 2002). On the other hand, it is generally accepted that unplanned collaboration does not necessarily lead to learning outcomes (NISE, 1997; Dillenbourg, 2002). Efficient collaborative learning greatly depends on the effort employed at design time (Goodyear, 2005). Particularly required are systematic approaches to designing CSCL that focus on the elicitation of well-known effective desired interactions (Strijbos, Martens, & Jochems, 2004).

Considering the difficulties of finding an appropriate trade-off between coercion and free collaboration, re-inventing and designing collaborative UoLs for each specific learning situation is costly. A solution to this problem is unraveling content, tools, specific learning tasks, etc., from the structure (the learning flow) of the “CSCL script” (hereafter referenced as scripts) so that this structure can be applied and reused in different settings, in a similar spirit to the approach presented in the previous Chapter 19. The effort involved in developing separated generic structures of scripts is justified if these learning
Diagrams of Learning Flow Patterns’ Solutions as Visual Representations

flows have been extensively tested and applied in a broad range of different situations. Examples of deeply evaluated collaborative learning flows are the well-known Jigsaw (Aronson & Patnoe, 1997), Simulation or Role-play and Think-Pair-Share strategies (NISE, 1997). Therefore, scripts exhibiting these commonly used structures (good or best practices) represent potentially effective scripts.

This chapter is based on our proposal (Hernández-Leo et al., 2005) related to the formulation of these generalized learning flows as patterns (Alexander et al., 1977), so that they are written in structured formats (including a diagrammatic representation of the solution) that make them easier to understand, (re)use, classify, etc. The resulting patterns, what we call CLFPs (Collaborative Learning Flow Patterns), offer a conceptual common ground among instructional designers and technology developers and a way of communicating collaborative learning expertise. In particular, this chapter provides an answer to the question: how can these types of patterns be integrated into instructional design practice? The pedagogical patterns collected in repositories are not broadly or satisfactorily (re)used (PPP, 2000; E-LEN, 2004). Our approach relies on the incorporation of patterns in authoring tools that graphically guide users to creatively obtain their own collaborative IMS LD compliant designs (Hernández-Leo et al., 2006). From a pedagogical angle (Strijbos et al., 2004), this assistance aims at reducing the complexity of the learning design process, as well as at guaranteeing potential effective results, since the guidance is based on the reuse of best practices in collaborative learning. To incorporate the patterns in authoring tools, we make use of refinable templates, i.e. partly completed re-usable designs (Hernández-Leo, Harrer, Dodero, Asensio-Pérez, & Burgos, 2006). With the aim of reducing the complexity from the technical angle (Griffiths et al., 2005), we use a specific graphical notation intended to be more intuitive: the diagrams of the patterns’ solutions. Remarkably, we have developed an IMS LD compliant editor called Collage, whose implemented design process incorporates combinable CLFPs as IMS LD templates (Hernández-Leo et al., 2006). This tool employs the CLFPs’ diagrams as the visual representation of the templates.

The structure of the chapter is as follows. The next section is devoted to introducing the idea of pattern-based design. It focuses on CLFPs, as a type of patterns for designing scripts, and discusses the role of patterns when they are integrated in authoring tools. Then, we face the implementation of CLFPs as IMS LD templates and their visual incorporation in the Collage authoring tool for the production of scripts (collaborative UoLs). Then follows an example that illustrates the creation of a script based on a combination of tree CLFPs. The chapter ends with a discussion and some concluding remarks.

**PATTERN-BASED DESIGN**

The word “pattern” has been used for centuries (e.g. patterns for dressmaking Holkeboer, 1987). However, it is broadly accepted that the idea of “pattern-based design” comes originally from architecture (Alexander et al., 1977), and has been also successfully used in software engineering (Gamma, Helm, Johnson, & Vlissides, 1995). Recently other domain specific patterns have been proposed, including (technology-supported) education (Derntl & Botturi, 2006). On page X of Alexander et al. (1977) Alexander defines a pattern as follows:

Each pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.

Therefore, patterns provide a structure for integrating the analysis and solution that can
be found to a problem, in a way that is sensitive to context. They offer guidance, but require embellishment. Alexander introduces 253 patterns in the architecture domain. He presents patterns for everything from designing independent regions, to cities, buildings and even single rooms. By connecting these patterns with common forces and other relations he transforms this collection of patterns into a pattern language, which provides a consistent way of creating a comfortable environment for people to live in. Apart from being a vehicle of creativity and communication in which structured ideas of good practices can be discussed, shared and modified, Alexander (1999) points out three essential features of a pattern language: a moral preoccupation (the ability to improve an environment), an aim of creating (morphological) coherence, and generativeness (e.g., the power to create morally sound and coherent objects).

For clarity purposes and in order to present each pattern connected to other patterns, Alexander uses the same general format for each pattern (Alexander et al., 1977; E-LEN, 2004). The relations with other patterns are indicated in the introductory paragraph, setting the context of the pattern, and the final paragraph, explaining how it can be embellished or completed. The essence of the problem and the solution are highlighted in bold type. The solution is also shown in form of a diagram, with labels to indicate its main components. This diagram is key to promoting a good understanding of the solution.

On the other hand, design patterns in software engineering are formulated using a different format, and tackle different types of problems. They describe how to solve particular problems that come up in software development. Gamma (Gamma et al., 1995) defines design patterns as, “Descriptions of communicating objects and classes that are customized to solve a general problem in a particular context” (p. 3).

Nevertheless, these patterns also target the provision of a common vocabulary, a common base of understanding regarding what is important in the area (programming vs. architecture) and a large corpus of solutions that makes users (developers vs. architects) more effective. With the same goal of collecting best practices, proven solutions and lessons learned, there are several initiatives that aim at applying the patterns approach to the educational domain, especially to technology-enhanced learning. Hernández-Leo, Villasclaras-Fernández, Asensio-Pérez, Dimitriadis, and Retalis (2006) expose a unifying view of some proposals for patterns in e-learning. It differentiates “patterns for analysis,” which deal with analyzing the usage of e-learning systems in order to improve them, from “patterns for design,” which are devoted to the design of e-learning systems (Retalis, Georgiakakis, & Dimitriadis, 2006) and pedagogical scenarios (PPP, 2000). Within the CSCL domain, there are also efforts that try to take advantage of the idea of pattern-based design. Currently, the TELL project represents the main initiative of collecting design patterns for teachers and educational CSCL (system) designers (TELL, 2005). Some of these patterns are devoted to the design of scripts, which is the focus of the following subsection.

**CSCL Scripting Patterns: Collaborative Learning Flow Patterns**

A “CSCL scripting pattern” (hereafter *scripting pattern*) describes a common problem and its corresponding broadly-accepted solution which can be used repeatedly in the design of collaboration scripts that are suitable of being formalized using a computer-interpretable notation. In other words, the final goal is that the resulting pattern-based scripts can be interpreted by an LMS. The main users of scripting patterns are practitioners and instructional designers, who construct collaborative learning plans. Scripting patterns can be formulated from general design ideas or structures based on any best/good practice when
Diagrams of Learning Flow Patterns' Solutions as Visual Representations

designing scripts to their reusable specialization. These practices can be grouped at different levels of granularity: set of activities that are organized in CL flows (coarser granularity level), the single activities themselves (actions within activities), and the resources (materials and tools) that support the single activities. Besides, there are some granularity level, we focus the next paragraph on this type of patterns: Collaborative Learning Flow Patterns, or CLFPs (Hernández-Leo et al., 2005). As we advanced in the introduction, CLFPs capture the essence of broadly accepted well-known techniques for structuring the flow of activities involved in collaborative learning.

<table>
<thead>
<tr>
<th>Table 1. Pyramid CLFP (short version)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TITLE</strong></td>
</tr>
<tr>
<td><strong>CONTEXT</strong></td>
</tr>
<tr>
<td><strong>PROBLEM</strong></td>
</tr>
<tr>
<td><strong>FORCES</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>SOLUTION</strong></td>
</tr>
<tr>
<td><strong>DIAGRAM</strong></td>
</tr>
<tr>
<td><strong>RATIONALE</strong></td>
</tr>
<tr>
<td><strong>EXAMPLE</strong></td>
</tr>
<tr>
<td><strong>ADDITIONAL INFORMATION</strong></td>
</tr>
</tbody>
</table>
This pattern gives the collaborative learning flow for a context in which several small groups are facing the study of a lot of information for the resolution of the same problem.

If groups of students face resolution of a complex problem/task that can be easily divided into sections or independent sub-problems, an adequate collaborative learning flow may be planned.

The flow of collaborative learning activities to be followed in order to solve a complex divisible task should promote the following educational benefits:

- To promote the feeling that team members need each other to succeed (positive interdependence)
- To foster discussion in order to construct students’ knowledge
- To ensure that students must contribute their fare share (individual accountability)

However, the solution for structuring collaboration in order to tackle this problem may be complex and probably more appropriate for collaborative learning experienced teachers and learners.

Each participant (individual or initial group) in a group ("Jigsaw Group") studies or works around a particular sub-problem. The participants of different groups that study the same problem meet in an "Expert Group" for exchanging ideas. These temporary focus groups become experts in the section of the problem given to them. At last, participants of each "Jigsaw group" meet to contribute with its "expertise" in order to solve the whole problem.

The Jigsaw structure, first introduced by Aronson et al. (Aronson & Patnoe, 1997), derives from practice (didacticism used in the practice) rather than from general learning theories (Johnson & Johnson, 1999), i.e. it represents a method that has been extensively tested and applied in a broad range of different settings and on which there are abundance of research (see, for instance, the references pointed out in the additional information section).

Collaborative understanding of a paper where each subsection (excluding the summary and introduction) is assigned to each member or every “Jigsaw Group.” More information available at http://gsic.tel.uva.es/collage.

(Aronson & Patnoe, 1997; Clarke, 1994; Johnson & Johnson, 1999)
Diagrams of Learning Flow Patterns’ Solutions as Visual Representations

Table 3. Think-Pair-Share CLFP (short version)

<table>
<thead>
<tr>
<th>Title</th>
<th>Think-Pair-Share Collaborative Learning Flow Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CONTEXT</strong></td>
<td>This pattern gives the collaborative learning flow for a context in which students are paired to solve a challenging or open-ended question. ***</td>
</tr>
<tr>
<td><strong>PROBLEM</strong></td>
<td>If groups of students face resolution of a challenging or open-ended question.</td>
</tr>
<tr>
<td><strong>FORCES</strong></td>
<td>The flow of collaborative learning activities to be followed in order to solve a challenging or open-ended question:</td>
</tr>
<tr>
<td></td>
<td>• To promote the feeling that team members need each other to succeed (positive interdependence).</td>
</tr>
<tr>
<td></td>
<td>• To foster discussion in order to construct students’ knowledge.</td>
</tr>
<tr>
<td></td>
<td>• To focus students’ attention on a particular topic.</td>
</tr>
<tr>
<td></td>
<td>• To give a chance to formulate answers by retrieving information from long-term memory.</td>
</tr>
<tr>
<td><strong>SOLUTION</strong></td>
<td>Each participant has time to think about the question. They pair and discuss their ideas about the question. Then, they comment or take a classroom “vote.”</td>
</tr>
<tr>
<td><strong>DIAGRAM</strong></td>
<td>They comment or take a classroom “vote”</td>
</tr>
<tr>
<td></td>
<td>They pair and discuss their ideas about the question</td>
</tr>
<tr>
<td></td>
<td>Each participant has time to think about the question</td>
</tr>
<tr>
<td><strong>RATIONALE</strong></td>
<td>Students are much more willing to respond after they have had a chance to discuss their ideas with a classmate because if the answer is wrong, the embarrassment is shared. Also, the responses received are often more intellectually concise since students have had a chance to reflect on their ideas with one another. See also the references pointed out in the additional information section).</td>
</tr>
<tr>
<td><strong>ADDITIONAL INFORMATION</strong></td>
<td>(NISE, 1997; Millis &amp; Cottell, 1998)</td>
</tr>
</tbody>
</table>
scenarios. These good practices pre-structure collaboration at a macro-level (Dillenbourg & Tchounikine, 2006) in such a way that they promote productive interactions, so that the potential effectiveness of the educational situation is enhanced. Tables 1, 2 and 3 show short versions of three CLFPs (complete versions of Pyramid and Jigsaw CLFPs are available in TELL, 2005), which are represented according to a structure based on the use of natural language. The format is similar to Alexander’s form, including also the diagrammatic representation of the solution.

It is not realistic to consider that collaborative learning scenarios are always structured as indicated by a unique CLFP. Like Alexandrian patterns, CLFPs can be used collectively in order to define richer collaborative learning flows. CLFPs can be combined: a particular phase of a CLFP can be structured according to another CLFP (that can be eventually the same). Or, they can be concatenated: some phases of a learning design are structured according to one CLFP and other (separated but consecutive) phases of the learning design are structured using another CLFP (that can be eventually the same).

Similarly to the patterns for Architecture or Software Engineering, the main advantages of CLFPs (and scripting patterns in general) are that they provide a conceptual common ground and a way of communicating, in this case, collaborative learning expertise of best practices to others, eventually novice practitioners. Instead of trying to create their own scripts from scratch, practitioners can use these patterns as a starting point. Consequently, the potential effectiveness of the scripts is enhanced, as is the design processes in terms of complexity and time demands. At this point, the following question arises: How can scripting patterns be effectively integrated into design processes? Since the patterns collected in repositories are not broadly and satisfactorily reused, our approach relies on the incorporation of patterns in authoring tools. An authoring tool based on patterns can guide educators to obtain effective scripts for their specific collaborative learning situations. In other words, the selection of a CLFP as the basis for the desired script would guarantee, to a great extent, the achievement of a set of objectives as dictated by previous experiences from which the CLFP emerged.

Therefore, the following question arises: How should the patterns be implemented in authoring tools? In other words, what is the role of patterns in authoring tools?

The Role of Patterns: Assistant vs. Template

As mentioned above, incorporating patterns in authoring tools potentially facilitates their reuse and the design of experience-founded scripts. The incorporation of patterns in authoring tools can be accomplished in two (complementary) ways so that the patterns act as assistants or as templates:

- Pattern-based assistant: a context-aware advising mechanism based on the knowledge formulated in the patterns. It is possible to imagine this type of assistant similar to, for example, the animated Microsoft Office Assistant that becomes visible whenever the system detects the user could benefit from its advice.
- Pattern-based template: a ready-made skeleton based on the knowledge formulated in a pattern that can be refined to create finished designs. This idea may remind readers, for example, of Web page templates (e.g. for Adobe Dreamweaver or Microsoft Front Page) that enable the easy production of Web pages by separating their presentation from the content.

The traditional dressmaking patterns are very close to this idea of template while more abstract rules, such as what types of buttons should be employed for a certain type of costume, are
more related to the idea of assistant. Depending on the nature of the different types of scripting patterns they can be implemented as assistant or as templates. In the case of the patterns at the collaborative learning flow level, when the degree of specialization of the pattern is such that its solution offers a learning flow (e.g. the CLFPs of Table 1, 2 and 3), this pattern is suitable of being provided as a template. On the contrary, if the pattern offers abstract advice, for example, to enrich the learning flow (see Enriching the learning process pattern of TELL, 2005), it may be implemented as an assistant. A similar analysis is applicable to the patterns at the activity level. On the other hand, there are already some proposals of the use of pattern-based templates for the creation of learning objects (Jones, 2004). Also at the resource level, patterns concerning educational tools may be offered as assistant that can act as a mediator between tool searchers and the teacher or instructional designer (Vega-Gorgojo et al., 2006).

The implementation of pattern-based assistant in authoring tools is outside the scope of this chapter. However, the next section faces the implementation of CLFPs whose solutions are flows of activity types as templates. Similar ideas could be adopted for the patterns at the activity level that provide ways of organizing single activities.

**IMPLEMENTING CLFPS AS IMS LD TEMPLATES**

The approach of providing CLFPs as templates of scripts can be accomplished with any educational modeling language that enables the computational representation of the scripts. With this purpose, we use IMS LD specification since it is currently accepted as the de facto standard for formalizing teaching-learning processes, such as the learning flows suggested in the CLFPs.

**IMS LD for Computationally Representing CLFP-Based Scripts**

IMS LD was realized by the IMS Global Consortium in 2003 (IMS, 2003). Chapter 2.10 describes the IMS LD notation in detail and reviews experience with its use. A Learning Design (LD) 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. A method contains the play, which is modeled according to a theatrical play with acts and role-parts. Since the specification provides a standard language for formally expressing learning situations based on different pedagogical theories (Koper & Tattersall, 2005), the reuse and interoperability of LDs in different systems is facilitated. A Unit of Learning (UoL) is a content package including an LD and a set of physical resources, or their location. It is an abstract representation of any learning event (a course, a lesson, etc.).

Although certain limitations have been observed, describing collaborative learning scenarios using IMS LD is feasible (Hernández-Leo et al., 2005). The main features of the specification that should be considered when describing group-based characteristics are the following. First, 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. Since roles can be nested (Hernández-Leo et al., 2005), indicating that a role is divided into sub-roles, it is possible to specify groups composed of other (smaller) groups or different roles. Second, IMS LD enables activities to be specified in coordinated collaborative learning flows in the method using mainly the act and the activity structure elements as well as conditions. Moreover, the play and
the monitoring service are also helpful when coordinating activities. Nevertheless, the teacher (or instructional designer) does not need to be familiar with the technical IMS LD constructs if high-level editors, such as the authoring tool presented in next section, are provided.

**Collage, a CLFP-Based IMS LD Editor of CSCL Scripts**

*Collage* (Hernández-Leo et al., 2006) is a high-level IMS LD authoring tool specially created to allow teachers to create their own collaborative UoLs (scripts) for collaborative learning by reusing CLFPs as templates. The scripts produced are IMS LD (level A) compliant. Therefore, they can be interpreted by any IMS LD player (Bote-Lorenzo et al., 2004; Tattersall et al., 2005), so that the participants of the educational design (learners and teachers) are automatically guided through the sequence of activities planned in the learning flow. Participants are also provided with the required resources in each activity.

With this tool, the user can create a UoL by specifying a set of objectives and prerequisites, the resources and tools that will be available, and the learning flow, i.e., the sequence of activities that the participants will perform and the roles that they will play. In fact, the edition of the collaborative learning flows by means of visual representations is Collage’s key feature. This edition is accomplished through the selection and configuration of whichever graphical CLFP-based IMS LD templates is best suited for a particular scenario. The learning activities specified by the templates can then be set up with a description and a set of resources and supporting tools. An activity may also be structured according to another CLFP. Thus, a single script may be composed by several CLFPs organized hierarchically as mentioned in a previous section.

The resulting learning flow (suggested by a single CLFP or a combination of several CLFPs) is presented in an intuitive way to help the user keep in mind a global view of the design, and to facilitate their comprehension and authoring. This presentation consists in two views. First, a global view of the LD structure is shown as a tree graph. When LD embraces several CLFPs, each pattern constitutes a node. Users can browse through the different sections that compose the LD by clicking on the nodes. The second view allows the user to zoom in to a particular section of the activity flow. This detailed view, is a graphical representation of the CLFP-based template to which the selected section belongs. This graphical representation corresponds to the diagram of the pattern’s solution so that the good practice offered by the CLFP is easily recognized and understood. In addition, it is worth mentioning that the implementation of the CLFP-based templates in Collage has been done so that their visual representations are highly interactive and including pattern-specific functions (e.g. selection of levels in a Pyramid CLFP-based template).

Thanks to these graphical representations, Collage reduces the complexity of the design process of potentially-effective collaborative UoLs. Given that we can create quite complex collaborative designs, in which several roles participate through a number of phases, it is useful to clearly represent each of these elements. While the resulting IMS LD document contains just a series of textual-notated acts with different activities, the graphical representation provided by Collage helps understanding the significance of each phase within the whole LD. This is possibly due to the CLFP-based design process, which takes advantage of previous knowledge that the practitioners may have (or may acquire) about the reusable good practices depicted in the patterns. At the same time, the use of these visualizations hides some difficult to understand IMS LD elements in such a way that the user does not need to know about their existence and function.

These benefits reflect a trade-off between constrained design options, and good reuse and particularization of CLFPs and an easy edition.
of collaborative LDs. These characteristics are not considered in more generic approaches, such as RELOAD LD editor (University of Bolton, 2004; Milligan, Beauvoir, & Sharples, 2005) and CopperAuthor (van der Vegt, 2005), which are close to the technical source of the specification and use text-based notation, or MOT+ (de la Teja, Lundgren-Cayro, & Paquette, 2005; see also Chapter 2.3) and ASK-LDT (Sampson, Karampiperis, & Zervas, 2005), which use graphical notations but do not incorporate abstractions easier to understand by teachers. The main features of Collage are illustrated in the following example drawn from a real case study.

**AN EXAMPLE BASED ON A COMBINATION OF THREE CLFPS**

This example illustrates how to use Collage for the CLFPS-based design of a script that belongs to an undergraduate engineering course on “Operation, Administration and Maintenance of Communication Networks.” The course is part of the 5th year (out of five) of the “Telecommunications Engineering” curriculum at the University of Valladolid, Spain.

The teacher (also playing the role of an instructional designer) wants his 12 students to collaboratively read a long, technical paper on a particular subject related to the course contents. The teacher, by setting this scenario, aims at achieving several learning objectives that include: improving the knowledge of the paper’s contents, using scientific and technical literature, and being able to synthesize what has been read. Also, the teacher expects that the collaborative nature of the setting helps the students to achieve other objectives not directly related to the course contents such as promoting positive interdependence among group members, fostering discussion, and individually accountability (each participant should be responsible for his/her contribution to the group work).

For fulfilling the above objectives, the teacher wants to design a blended collaborative learning setting spanning two two-hour face-to-face synchronous sessions at the lab; and a virtual (or distant) asynchronous collaborative session in between. For both types of sessions, several supporting telematic tools are available: document viewers and editors, a Web-based shared document repository, a Web-based questionnaire tool, and a chat tool.

Taking into account the context and the number of students as well as the time restrictions, the teacher has to face the complex task of designing a set of sequenced collaborative learning activities, eventually supported by the available tools, which effectively guide the students towards the achievement of the desired learning objectives. As it was mentioned throughout the chapter, this task becomes even more complex if the resulting learning design needs to be formalized using IMD LD so that it can be interpreted by a LMS. How might the Collage authoring tool, as well as the underlying CLFP-based design process it enforces, help the teacher?

The first step when using Collage (Hernández-Leo et al., 2006) consists of selecting the set of CLFPs that, according to the desired learning objectives and the type of learning task to be faced by the students (in this case, reading a technical paper), which might be more suitable for structuring the flow of learning activities. Although this first step avoids having the teacher/instructional designer start the design process from scratch, it is not a straightforward step: Not all CLFPs are suitable for all types of learning tasks and objectives, and a bad selection might result in unsuccessful design.

Taking into account the description of Table 1, the Pyramid CLFP promotes positive interdependence and fosters discussion with the final goal of achieving gradual consensus on the
Diagrams of Learning Flow Patterns’ Solutions as Visual Representations

resolution of a task without a concrete solution. Within the context of this example, the teacher might opt for requesting the students to read the same paper and reach a final consensus, following the structuring of the Pyramid, on the eight most important ideas of the paper and two questions they would like to be solved. In this way the teacher does not simply limit the learning task to reading the paper, he is proposing a collaborative way of solving a problem (without a concrete solution) that might help to achieve further learning objectives, according to the experience from which the good practice collected in the CLFP has been drawn.

But the Pyramid CLFP by its own does not seem to provide all the confidence in achieving all the desired learning objectives. In this sense, and according to the descriptions of Table 2, the Jigsaw CLFP promotes individual accountability when a general problem is divided into sub-problems to be faced by collaborating small groups. How could the teacher combine the potential advantages of the Jigsaw CLFP with those of the Pyramid CLFP? And, how could this CLFP be applied to the particular task that has been proposed above? (i.e., reading the paper and agreeing in a set of ideas and questions). One potential solution would be to consider that at the first level of the Pyramid (“Each participant studies the problem and proposes a solution...” in Table 1) instead of individual students, groups of them propose a solution after having worked together according to the Jigsaw CLFP. For applying this pattern, the general problem (reading the paper) is divided into sub-problems (each member of the group reads just one part of the paper). Then “experts” of the different groups exchange ideas (those students that have read the same part of the paper) in order to contribute to the resolution of the general problem within their groups (again, agreeing in a set of ideas and questions). Using the Jigsaw CLFP in this case reduces the workload (each student just reads one part of the paper) without precluding the overall understanding of the paper contents (thanks to the “expertise” provided by other group members).

According to the Pyramid CLFP solution, the final expected learning activity (last level of the Pyramid) implies that all the students discuss and agree on a final proposal of ideas and questions on the paper. For structuring and fostering this final discussion, the Think-Pair-Share CLFP may also be useful (see Table 3). In this case, each participant of the CLFP would be a group of students (not an individual). The first phase of the Think-Pair-Share (Think) would not be needed as each participating group has its own proposal on ideas and questions about the paper (derived from the previous Pyramid activities). The Pair phase might be carried out by means of a speaker from each group that explains each proposal. Finally, the Share phase (overall discussion) might be mediated by the teacher.

Therefore, the first teacher’s decision might result in structuring the learning design according to the Pyramid CLFP but using the Jigsaw CLFP for the first level and the Think-Pair-Share CLFP for the last one. As we detailed in the previous section, the Collage tool incorporates the diagrams of the CLFPs (see Table 1, 2, and 3) as a graphical notation that facilitates users’ (teacher/instructional designers) selection and combination. Figure 1 illustrates how this graphical notation is used for the proposed example.

After selecting and combining CLFPs, the next step in the Collage CLFP-based design process implies refining those CLFPs (implemented as IMS LD templates) so as to obtain the final UoL. This step includes design activities such as describing the learning activities (e.g., reading a part of the paper, discussing questions/ideas, etc.), providing information about roles (including groups), establishing group-size limits, and determining and configuring the resources needed to support each activities. Table 4 summarizes a potential result of this refining step for the number of students of the example (12), the available tools, and the time frame.

The described refining step is graphically supported by the Collage tool. Figure 2 shows how the teacher provides details on the “pair” phase of
Diagrams of Learning Flow Patterns’ Solutions as Visual Representations

Figure 1. Planning the learning flow: combining Pyramid, Jigsaw and Think-Pair-Share CLFPs in form of IMS LD templates using a graphical notation in which the visual representation of each template corresponds to the diagram of its related pattern’s solution.
the TPS CLFP that structures the activities at the level 3 of the Pyramid CLFP. (The complete UoL is available at http://gsic.tel.uva.es/collage.)

**DISCUSSION**

Our proposal of our using the diagrams of learning flow patterns’ solutions as visual representations of refinable IMS Learning Design templates adds intuitiveness (as characterized by Waters et al., 2004) and familiarity (as described in Chapter 2.9) to the IMS LD notation system. This intuitive intelligibility results from the visual similarity to the mental images and ideas that the potential users have or build regarding the patterns. On the other hand, this approach represents a pragmatic solution grounded in practice to apply moderate constructivism assumptions compatible with ID practices and with the support of technology tools, as demanded in (Karagiorgi et al., 2005).

There are currently only six patterns incorporated in Collage. However, since they can be creatively combined in order to create new richer learning flows, it offers some degree of flexibility. As categorized in Botturi et al. (2006), Collage provides generative patterns which can then communicate with a more finalist language (IMS LD) using the graphical representations of the patterns’ solutions. However, one of the
Diagrams of Learning Flow Patterns’ Solutions as Visual Representations

limitations of Collage is that the addition of new templates based on other patterns is laborious: each template has its own representation and includes pattern-specific functions. To overcome these drawbacks, it would be interesting to study the possibility of using visual languages for instructional design, such as E²ML (Botturi, 2006; see also Chapter 7) or the others VIDLs presented in Section 2 of this handbook, to graphically represent patterns’ solutions. This may not only facilitate the addition of new patterns (using the same graphical notation) to Collage, but it may also afford more flexible editing possibilities of the learning flow (e.g., using drag-and-drop elements of the visual language that can be added or removed from the templates).

CONCLUSION

With the aim of integrating technology in education, instructional designers and teachers can use Learning Technology standards, such as IMS LD. However, these standards and specifications provide technical-oriented, text-based notation systems that do not offer visual representation of instructional activities and do not enforce design processes that support the creation of pedagogically sound designs. This chapter has offered an approach that provides good practices formulated as patterns and visualized through graphical diagrams within authoring tools. These tools support users in the creation of IMS LD compliant designs (UoLs) by means of reusable
Diagrams of Learning Flow Patterns’ Solutions as Visual Representations

pattern-based templates. The three different ideas linked in this approach are:

• **The use of patterns for the design of experience-founded learning scenarios.** Our particular focus is on CSCL scripts, i.e., collaborative learning scenarios that can be interpreted by LMSs. That is the reason why we describe the different types of patterns that can be used in the design of scripts, and analyze in more detail the CLFPs, which suggest flows of learning activity types. In addition, we argue that some patterns can act as assistants (advising mechanisms) while others are more suitable as refinable templates (partially completed designs).

• **The incorporation of pattern-based templates in authoring tools using visualizations inspired by the patterns’ solutions.** Collage authoring tool represents an example of a pattern-based editor that facilitates teachers (and instructional designers) in the creation of scripts based on combinations of CLFPs. The pattern-based design process implemented in Collage relies on the use of a visual notation in which the graphical representation of each learning flow template corresponds to the diagram of the related CLFP. Apart from the reuse of good practices that facilitate planning elicitations of desired social interactions, the Collage design process reduces the technical complexity associated with creating computer-interpretable collaboration scripts.

• **The production of CSCL scripts compliant with IMS LD.** IMS LD specification can be used to computationally represent CLFPs-based templates. In fact, the scripts produced by Collage are IMS LD (level A) compliant.

These ideas have been thoroughly illustrated with an example drawn from a real course, which shows the feasibility and usefulness of the whole approach. We are currently carrying out several case studies with teachers, who use Collage creating pattern-based scripts, and students, who experience these scripts. The conclusions of these case studies will provide further insights about the possibilities of the approach presented in this chapter. Future work includes the study of using visual languages to represent patterns’ solutions (and thus pattern-based templates). This work envisages a potential chance to face the challenges of easily adding new patterns to Collage and providing more flexible refinement possibilities in the authoring.

**ACKNOWLEDGMENTS**

The authors wish to thank the rest of the members of the GSIC/EMIC group at the University of Valladolid, Spain. Special thanks to Iván M. Jorrín-Abellán, Bartolomé Rubia-Avi and Inés Ruiz-Requies. This work has been partially funded by the Spanish Ministry of Education and Science project TSI2005-08225-C07-04, the Autonomous Government of Castilla and León project VA009A05 and the Kaleidoscope Network of Excellence FP6-2002-IST-507838.

**REFERENCES**


Johnson, R. (2004). Designing adaptable learning resources with learning objects patterns. *Journal of Digital Information, 6*(1), Article No. 305 Available at http://jodi.ecs.soton.ac.uk/Articles/v06/i01/Jones/


Diagrams of Learning Flow Patterns' Solutions as Visual Representations


TELL (2005). Design patterns for teachers and educational (system) designers. TELL project deliverable.


