CSCL Scripting Patterns: Hierarchical Relationships and Applicability

Davinia Hernández-Leo[1], Eloy D. Villasclaras-Fernández[1], Juan I. Asensio-Pérez[1],
Yannis A. Dimitriadis[1], Symeon Retalis[2]
[1]University of Valladolid, Spain, [2]University of Piraeus, Greece
{dherleo, evilfer, juaase, yannis}@ulises.tel.uva.es, retal@unipi.gr

Abstract

The use of patterns in e-learning is being recently proposed with different purposes and scopes. This paper provides a unifying view of several representative proposals in order to situate and introduce the types of patterns that can be used for generating collaboration scripts that are suitable of being computationally represented and interpreted by Learning Management Systems (LMSs). The paper also presents a hierarchical structure that relates these patterns according to their focus and granularity. The structure shows how the different types of patterns are complementary and should be considered in the process of creating a script. In addition to authoring tools and LMSs, there are other types of tools in which the application of patterns may be useful. This paper points out these tools locating them along the "life cycle" of the scripts.

1. Introduction

The design of Computer-Supported Collaborative Learning (CSCL) situations should include positive well-known forms of interaction among participants. A way to deliberately promote productive interactions that enhance the effectiveness of collaborative learning (CL) is the use of collaboration scripts, which describe the structure of CL processes [4]. Moreover, in the context of technology-enhanced learning, the behaviour and functionality of an LMS (Learning Management System) can be controlled by a computer-interpretable collaboration script [3]. The computational representation of a collaboration script is called hereafter a CSCL script [13].

Nevertheless, conceiving collaboration scripts and their computational representation is not a trivial problem. From a pedagogical perspective, designing potentially-effective CSCL scripts is challenging because it is not easy to find the appropriate trade-off between coercion and free collaboration [4]. This challenge is even stronger due to the difficulties involved in modelling CL processes from a technical point of view (formally specifying groups, CL flows, etc.) [13] and because of the fact that computer-interpretable notations (e.g. XML) are not familiar to educators. An example is the IMS Learning Design (LD) educational modelling language that enables the specification of learning processes [12].

A promising solution to this problem is related to (re)using generalizations of successful collaboration scripts formulated as patterns. These patterns can be provided as templates formalized using a computer interpretable notation, thus reducing the technical complexity of authoring CSCL scripts [10]. These patterns for creating collaboration scripts that are suitable of being formalized as CSCL scripts are what we call "CSCL scripting patterns".

Despite the fact that the word "pattern" has been used for centuries with slightly different meanings, their use is more known in the fields of Architecture [1] and Software Engineering [6]. A pattern provides a means of organizing information regarding a contextualized common problem and the essence of its broadly accepted solution, so that it can be repetitively applied. A collection of interconnected (related) patterns which enables the generation of a coherent whole (e.g. a town) is called a Pattern Language (PL) [1]. Recently other domain specific patterns have been proposed, including e-learning [2,5,7,8,11,15,17]. However, not all the e-learning patterns follow the same approach, nor have the same purpose.

The aim of this paper is threefold. The first goal is to provide a unifying view of several approaches with different scopes regarding the use of patterns in e-learning. That allows us to situate the types of CSCL scripting patterns that we present in this paper. We have already proposed the use of a particular type of patterns for the design of CSCL scripts in [10]. These patterns (Collaborative Learning Flow Patterns or CLFPs) describe in their solutions general structures of CL activity flows, which can be adapted to numerous educational scenarios. However, in the design process of scripts based on CLFPs other types of patterns may be involved. Therefore, the second goal of this paper is to describe the space of possible CSCL scripting patterns and to relate them by means of a hierarchical structure according to their granularity and focus.

At the beginning of the introduction we highlighted that CSCL scripts can influence the behaviour of an LMS since they are interpretable by computers. Nevertheless, this is not the only type of tools that is needed in order to facilitate the use of CSCL scripts that are based on patterns. In this sense, the third goal of this paper is to discuss these types of tools and, additionally, the types of tools that may benefit from the use of patterns in the field of CSCL scripting.

The rest of this paper is structured as follows. Section 2 exposes a unifying view of some proposals of patterns in e-Learning. Section 3 is devoted to the description of the space of patterns for conceiving CSCL scripts. The computer-supported applicability of these patterns is discussed in section 4. To finish, section 5 presents conclusions and our future work.

2. Patterns in e-learning

Figure 1 aims to relate existing patterns proposals that refer to different scopes of the e-learning field. In general two main types of patterns can be distinguished depending on their use. Firstly, “**patterns for analysis**” deal with analyzing the usage of e-learning systems in training or academic contexts, in order to help teachers to continually improve them [14]. The PoInter project is concerned with investigating the appropriateness of patterns as a means of communicating information about how people interact with each other through technology [14]. This type of patterns may be also classified as **patterns of interaction** or **patterns of behavior**.

Secondly, “**patterns for design**” are devoted to the design of e-learning systems. This is the wide scope of E-LEN project [5], which proposes patterns for implementing an institutional e-learning centre. Within this scope, patterns for designing learning scenarios (pedagogical patterns) and patterns for designing technological solutions that supports these scenarios (technological patterns) can be differentiated. On the one hand, pedagogical patterns try to capture expert knowledge of the teaching/learning practice. These patterns propose solutions for problems such as motivating students, choosing and sequencing materials, or evaluating students [15]. Some patterns for (CS)CL, the focus of TELL project [17], can be classified as a type of pedagogical patterns. For example, [8] illustrates a pattern language for “Debate”. On the other hand, technological patterns proposed in [2] record design experience with regard to the construction of LMSs. Focusing on learning objects reusability, [11] proposes the use of patterns to produce reusable designs for creating learning resources that are adaptable.

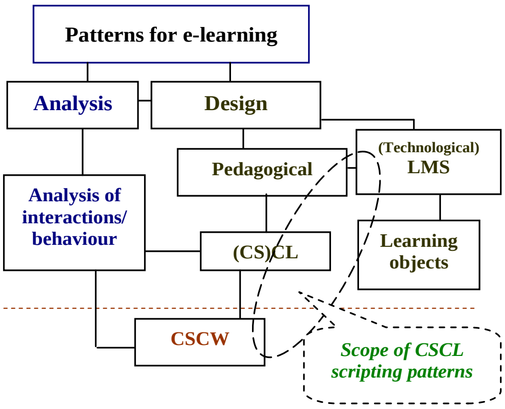

Figure 1. Relationships between some proposals of patterns in e-learning

Patterns from other disciplines or fields can also be useful in the design of e-learning. For example, CSCL can be greatly benefited of knowledge in “groupware” or CSCW (Computer-Supported Cooperative Work), since CSCL and CSCW have much in common (both support group work, provide shared interfaces, etc.). An example of patterns for groupware is GAMA, a PL that provides patterns for supporting dynamic teams using computer technology [16].

It is necessary to remark that there are not clear boundaries among the different approaches. Patterns that result from interaction analysis are related (or can belong) to the domain of CSCL and CSCW. Patterns for designing ANSCL (Asynchronous Network Supported Collaborative Learning) systems [7] can be considered both patterns for CSCL and patterns for LMSs. A similar consideration may be made for CSCL scripting patterns, presented in next section, since they are to be interpreted by LMSs and may include learning objects (Figure 1 illustrates their scope).

3. CSCL scripting patterns

This section describes the space of possible patterns for creating collaboration scripts that are suitable of being computationally represented as CSCL scripts: CSCL scripting patterns. These patterns capture the essence of best (or good) educational practices when creating CSCL script. The practices can be grouped at different granularity levels: set of activities that are organized in CL flows vs. single activities vs. the resources (materials and tools) that supports the single activities. Patterns at the different levels are complementary and need each other for completeness, so they can be related forming a hierarchical structure (which represents the structure of pattern languages for CSCL scripts). That is to say, in order to generate a CSCL script, different types of patterns should be considered.

3.1. Hierarchical structure

Some authors already distinguish between macro scripts and micro scripts [9]. Coarse-grained (or macro) scripts describe general flows of collaborative (or not) learning activities (e.g. those following the Jigsaw technique [10]). Fine-grained (or micro) scripts give detailed support within specific activities (e.g. scripts for argumentative knowledge construction [19]). In the same sense, the first (coarser) granularity level for CSCL scripting patterns is related to the **CL flow**: the sequence of activities that make up a learning process. Some examples of patterns at this level are JIGSAW and PYRAMID (or SNOWBALL) CL Flow Patterns (CLFPs) [10]. Another granularity level refers to the **activities** themselves (see Figure 2). An example of a pattern at this level is DISCUSSION GROUP [8]. Figure 2 illustrates how a PL for CSCL scripts may include other pattern languages, for example “Debate” PL [8]. In addition, we propose a third granularity level that includes the **resources** (materials and tools) needed to support the activities. Some examples are the patterns proposed in [7] such as ANNOTATION ON POSTED MESSAGE. These tree types of granularity correspond to three horizontal levels whose patterns are connected vertically.

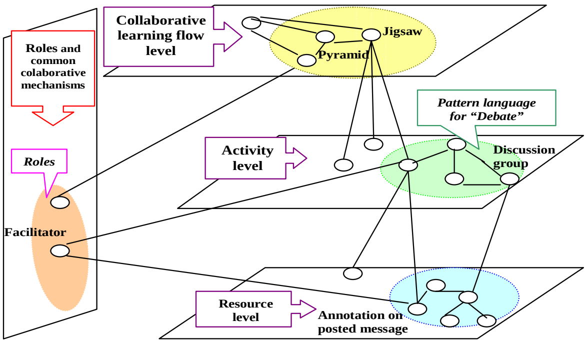
Roles and common colaborative mechanisms
Collaborative learning flow level
Jigsaw
Pyramid
Pattern language for “Debate”
Activity level
Discussion group
Roles
Facilitator
Resource level
Annotation on posted message

Figure 2. Hierarchical structure of CSCL scripting patterns. Patterns are identified with nodes, which are related by edges

Besides, there are some aspects (such as **roles** or **common collaborative mechanisms**, namely group formation, roles/resources distribution, floor control, awareness) that can be connected directly to some of the patterns at any of the aforementioned granularity levels. For example, roles can be defined globally at the level of the whole learning flow, within activities or/and within collaborative tools (e.g. usage of the FACILITATOR pattern [8]). Thus, the patterns that state a principle about these aspects are at a vertical level.

The following example of a collaboration script, which is designed according to patterns at the different levels, aims to clarify the relationships between them. It also illustrates how the hierarchical structure conceptually guides the process or sequence that can be used when applying the patterns. In the example, it is a refinement process: from top (coarse grain) to bottom (fine grain). The reader should also note how the patterns used in this example form themselves a small pattern language. Other patterns useful in the creation of CSCL scripts can be also located at the levels considered in the hierarchical structure, potentially forming other different PLs.

3.2. Example

The context of the example is a university course about “computer architecture”. Four students have to collaboratively propose a computing system for a client with particular requirements according to a three-level PYRAMID CLFP. Each participant studies the problem and proposes a solution. After that, they discuss asynchronously in pairs according to the DISCUSSION GROUP pattern in order to propose a new common solution. Finally, the four students have to agree on a shared solution following the same pattern. The teacher should guide both discussions playing the role suggested by the FACILITATOR pattern. This role should be supported by the asynchronous collaborative system used for the discussion. It would be also useful that this system implements the ANNOTATION ON POSTED MESSAGE pattern in order to let the user know of actions taken by other participants.

3.3. Discussion

Not only are the patterns related to others at a different granularity level but they may complement other patterns at the same level of granularity (see Figure 2). CLFPs, for example, may be combined or concatenated forming what we call CLFPs hierarchies [10]. The precise identification of relations between patterns is still an open research issue. In addition, CSCL scripting patterns are completed with other PLs (e.g. PLs of didactics for specific subject matters [15]), contain other PLs (see Figure 2), and are embraced by other PLs. For example, higher level patterns of CSCL scripting patterns are the patterns related to high level pedagogy (e.g. collaborative knowledge building) [8].

We should also distinguished, as [1] does, the terms *design patterns* and *construction patterns.* Alexander's design patterns refer to understanding the geometry of a building and the relationships between parts, while construction patterns examine the materials and processes needed in order to put the designs into practice. We distinguish between *design CSCL scripting patterns*, which are used to devise the educational design of a script (patterns at the CL flow and the activities levels); and *construction CSCL scripting patterns*, which supports the implementation of the designs in actual practice (patterns at the resource and vertical level). The audience of design CSCL scripting patterns are mainly teachers and learning designers, who construct CL plans. Nevertheless, these patterns may be used by systems designers in requirements analysis tasks. In contrast, construction CSCL scripting patterns are more intended for system developers (or content providers), although they should be also considered by teachers and even students, who are the actual users of the scripts and its computer-supported application.

4. Computer-supported applicability

Different types of tools may benefit from the use of patterns for CSCL scripts or are needed to facilitate their use. This section discusses the characteristics of some of them classified according to the different stages (designing, instantiating, executing) of a CSCL script "life cycle" (see Figure 3).

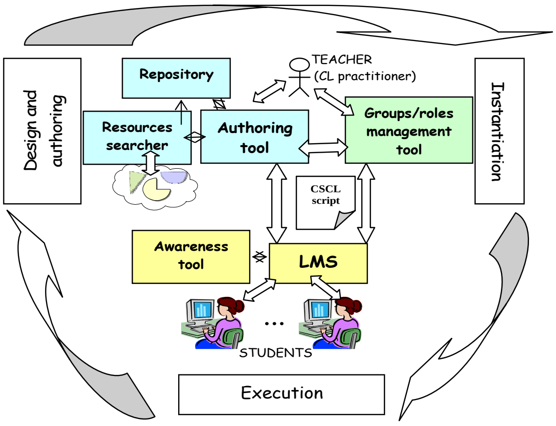

Figure 3. "Life cycle" of CSCL scripts and related tools

Within the phase in which CSCL scripts are conceived and created (design and authored), four types of tools may be distinguished. CSCL scripting patterns and scripts (generated by applying the patterns) may be collected in **repositories**. The main challenges for this type of tool is related to facilitating collaboration for the joint development of scripts, and to "labeling" patterns and scripts to ease their sharing. The reuse of CSCL scripting patterns can be fostered by incorporating them in **authoring tools** so they may provide advice along the design process. In addition, CSCL scripting patterns whose solutions propose structures of scripts can be represented computationally and implemented in authoring tools as a kind of templates that can be easily completed in order to create computer-interpretable scripts. *Collage* [10] is an example of an editor that uses CLFPs as the basis for generating new CSCL scripts (formalized with LD). This idea can be also applied to patterns at the activity and resource level (e.g. learning objects patterns [11]).

To create CSCL scripts, practitioners also need to select tools (not to generate them) that are to support the activities. In this line, semantic search of tools using ontologies is being researched by [18]. Therefore, some patterns at the resource level (this is applicable to tools and learning materials) can be implemented as "advisors" that can act as a mediator between the **resource searchers** and the user.

During the instantiation of a CSCL script [10], **tools for managing roles and groups** are also necessary. This type of tools should easily enable the creation of multiple groups or roles and the further binding of individuals according to the knowledge captured in the patterns and the pattern-based structure of a script, which may be quite complicated.

Regarding the interpretation (i.e. execution) of CSCL scripts, the most important types of tools are players and **LMSs**. A system that interprets CSCL scripts should consider the information collected in the patterns. That is, it should be able of interpreting scripts at the learning flow level or/and at the activity level, provide the needed resources, etc. In addition, CSCL scripting patterns can be used by **awareness tools**. For instance, a CL flow awareness tool (based on CLFPs) will allow participants to be aware of the collaborative learning flow during execution: which activities have been accomplished, which are the next ones, in which activities are involved the rest of the participants, etc. In many CL situations, having such awareness is crucial since participants may change their groups depending on the phase of the learning flow and may need to know the progress of their future team partners.

5. Conclusion and future work

Within a unifying view of several different pattern-based approaches in e-learning, this paper has positioned and presented the types of patterns that can be used in order to generate CSCL scripts. These types of patterns are related according to their granularity and focus forming a hierarchical structure so that the process for understanding and applying the patterns in order to create scripts is conceptually facilitated. In this sense, we are currently working on a meta-model (conforming to the proposed hierarchical structure) for describing CSCL scripting pattern languages, on describing a method for the further generation of scripts, as well as on analyzing the possibilities and limitations of computationally representing CSCL scripts using IMS-LD.

On the other hand, the paper has introduced some tools required along the “life cycle” of CSCL scripts in which CSCL scripting patterns may be useful. We have implemented an authoring tool (Collage) and an LMS (Gridcole). Furthermore, we are currently developing a group management tool, which will allow us to integrate Collage and Gridcole towards their use in real practice.

Acknowledgments

This work has been partially funded by the e-Learning project EAC/61/03/GR009, Kaleidoscope NoE FP6-2002-IST-507838, Spanish Ministry of Education and Science projects TIC-2002-04258-C03-02 and TSI2005-08225-C07-04 and Autonomous Government of Castilla and León, Spain, projects VA009A05, UV46/04 and UV31/04. The authors would like to thank the rest of the members of the GSIC-EMIC research group.

References

[1] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., and Angel, S. *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, NY, 1977.

[2] Avgeriou, P., Papasalouros, A., Retalis, S., and Skordalakis, M., "Towards a Pattern Language for Learning Management Systems", *Educational Technology & Society*, vol. 6, no. 2, June 2003, pp. 11-24.

[3] Bote-Lorenzo, M.L., Hernández-Leo, D., Dimitriadis, Y., Asensio-Pérez, J.I., Gómez-Sánchez. E., Vega-Gorgojo, G., and Vaquero-González, L.M., "Towards Reusability and Tailorability in Collaborative Learning Systems Using IMS-LD and Grid Services", *Advanced Technology for Learning*, vol. 1, no. 3, Sept. 2004, pp. 129-138.

[4] Dillenbourg, P., Over-Scripting CSCL: The Risks of Blending Collaborative Learning with Instructional Design. In: Kirschner, P.A. ed. *Inaugural Address, Three Worlds of CSCL. Can We Support CSCL?* Heerlen: Open Universiteit Nederland, 2002, pp. 61-91.

[5] ELEN project website, http://www2.tisip.no/E-LEN/, last visited: Jan. 2006.

[6] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison- Wesley, USA, 1995.

[7] Georgiakakis, P. and Retalis, S., "Demystifying the Asynchronous Network Supported Collaborative Learning Systems", *International Journal of Computer Applications in Technology*, vol. 25, no. 2/3, 2006.

[8] Goodyear, P., "Educational Design and Networked Learning: Patterns, Pattern Languages and Design Practice", *Australasian Journal of Educational Technology*, vol. 21, no. 1, 2005, pp. 82-101.

[9] Harrer, A., Malzahn, N., Jermann, P., and Dillenbourg, P., "Bridging the Gap - Different Level of Granularity for CSCL Scripts", Workshop position paper, *Computer Supported Collaborative Learning 2005: The Next 10 Years!*, Taipei, Taiwan, May 2005.

[10] Hernández-Leo, D., Villasclaras-Fernández, E.D., Jorrín-Abellán, I.M., Asensio-Pérez, J.I., Dimitriadis, Y., Ruiz-Requies, I., and Rubia-Avi, B., "COLLAGE, a Collaborative Learning Design Editor Based on Patterns", *Educational Technology and Society,* vol. 9, no. 1, Jan. 2006, pp. 58-71.

[11] Jones, R., "Designing Adaptable Learning Resources with Learning Objects Patterns", *Journal of Digital Information*, vol. 6, no. 1, Dec. 2004, pp. Article No. 305.

[12] Koper, R. and Tattersall, C., *Learning Design, a Handbook on Modelling and Delivering Networked Education and Training*, Springer-Verlag, Heidelberg, 2005.

[13] Miao, Y., Hoeksema, K., Hoppe, H.U., and Harrer, A., "CSCL Scripts: Modelling Features and Potential Use", *Computer Supported Collaborative Learning 2005: The Next 10 Years!*, Taipei, Taiwan, May 2005, pp. 423-432.

[14] Pointer project, Patterns of Interaction: a Pattern Language for CSCW, http://www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/, last visited: Jan. 2006.

[15] Pedagogical patterns project, http://www.pedagogicalpatterns.org/, last visited: Jan. 2006.

[16] Schümmer, T., "GAMA, a Pattern Language for Computer Supported Dynamic Collaboration", *Conference on Pattern Languages of Programs,* Irsee, Germany, 2003.

[17] TELL project website, http://cosy.ted.unipi.gr/tell/, last visited: Jan. 2006.

[18] Vega-Gorgojo, G., Bote-Lorenzo, M.L., Gómez-Sánchez, E., Dimitriadis, Y., and Asensio-Pérez, J.I., "A Semantic Approach to Discovering Learning Services in Grid-Based Collaborative Systems", *Future Generation Computer System,* 2006, (in press).

[19] Weinberger, A., Fischer, F., and Stegmann, K., "Computer-Supported Collaborative Learning in Higher Education: Scripts for Argumentative Knowledge Construction in Distributed Groups", *Computer Supported Collaborative Learning 2005: The Next 10 Years!,* Taipei, Taiwan, May 2005, pp. 717-726.