Looking Into Collaborative Learning: Design from Macro- and Micro-Script Perspectives

Eloy D. Villasclaras-Fernández[a,1], Seiji Isotani[b], Yusuke Hayashi[b], Riichiro Mizoguchi[b]
[a] *GSIC-EMIC group, University of Valladolid, Spain*
[b] *Mizoguchi Lab, ISIR, Osaka University, Japan*

Abstract. Design of collaborative learning (CL) scenarios is a complex task, but necessary if the goal of the collaboration is learning. Creating well-thought-out CL scenarios requires experience and knowledge about different learning/ instructional theories and practices related to collaboration. Inexperienced designers, who may not have all the needed knowledge to formulate pedagogically sound collaborative learning plans, may have difficulties to design CL activities. Thus, although CL has shown to be effective, it is not often used in classroom adequately. To tackle this problem, this paper presents an ontology for CL, that aims at modeling not only the learning activities, but also the rationale for their configuration in pedagogical terms. This ontology allows modeling the expected interactions among students as well as the mechanisms that promote or enhance collaboration structured across several sessions and involve individual and collaborative activities. Finally, the paper discusses an application of the presented ontology to support the design of pedagogically sound CL scenarios.

Keywords. Collaborative learning, ontology engineering, script design, intelligent authoring systems

Introduction

Design of collaborative learning scenarios is currently one of the relevant topics in the e-Learning research community. In spite of the potential benefits of collaborative learning activities, there is no guarantee that unstructured collaboration will result in the expected learning outcomes [1]. This problem has been tackled by the use of scripts, both with computer support (as in CSCL settings) or without it. By structuring collaboration and guiding participants through learning activities, collaboration scripts have shown to be an effective mechanism to improve the chances of fruitful interactions among participants [1].

Scripts describe a model of the collaborative processes that are expected to take place in the learning setting. Indicating the complexity of collaborative scenarios, different models have been proposed to provide guidelines for script design or analyze the components that compose a collaboration script, such as [2], [3], [4]. With the aim of fostering productive interactions, scripts may describe a range of different features of collaborative activities. With respect to this, scripts have been classified in micro-

[1] Corresponding Author. Tel. (+34) 983 423 696 . E-Mail: evilfer@ulises.tel.uva.es

and macro-scripts. Micro-scripts describe the communication process that students are expected to engage on. Examples of micro-scripts include those for the construction of arguments or argumentation sequences [5]. Macro-scripts, on the other hand, deal with the organization of coarser-granularity sessions, including the description of groups, roles, activities or phases [6]. Examples of macro-scripts are the Arguegraph [1], or those based on well known structures such as Jigsaw or Pyramid [7].

Configuring these components is a complex task, which demands careful planning [1]. This issue is specially challenging in the case of non-expert designers, such as teachers engaged in the design and application of collaborative activities to their own practice. Design of CL scenarios is the objective of different software tools, by means of which designers may generate collaborative (or individual) learning scripts, according to different design approaches. Available tools include LAMS [8], Collage [7]; CHOCOLATO [9], DialogPlus [10], etc.

Besides generating formal representations of scripts, for instance using IMS-LD (IMS-LD) [11], authoring tools can provide support to designers in the design task, for instance by encouraging and facilitating the designer to reflect about the pedagogical model encompassed in the script. This paper is focused on this issue, and on the use of design knowledge about both macro- and micro-scripts. In fact, there is a lack of models and tools which combine elements from macro- and micro-scripts. We argue in the following section that combining these two perspectives can help the understanding of collaborative processes represented in a script. Thus, in order to enable software tools to make use of such design knowledge, from both macro- and micro-script perspectives, we adopt the approach of ontological engineering (OE) [12]. OE is used to build an ontology that represent, explicitly and formally, the main concepts related to the design knowledge about collaborative interactions and mechanisms that structure collaborative learning sessions. This paper describes the work undertaken to develop such ontology and a usage example to support the design of structured collaboration.

The rest of the paper is organized as follows: next section discusses in more detail the problem of designing CSCL scripts. Section 2 describes the features of the ontology presented in this paper, and Section 3 illustrates these features by means of an application. Finally, Section 4 presents the conclusions and future work.

1. Design of collaboration scripts

In order to capture design knowledge to aid designers in the creation of collaboration scripts, we need a conceptual framework to describe not only the expected behavior of the participants, but also the rationale or purpose of each component of the script. This section analyzes the design principles that we take as the base of this design rationale.

First, the design process is driven by the selection of the learning objectives pursued by the script. Learning objectives, such as knowledge acquisition or cognitive/meta-cognitive skill development, should be the first aspect considered by the designer [2], since they determine the adequate interactions that should occur among students. With respect to this, we already indicated above that micro-scripts can be used to describe the interaction process that students are expected to follow.

However, while interactions are the main mechanism to promote learning [13], structuring CL scenarios involves further issues. According to [14], promotive interaction is actually only one of the essential elements of collaboration. Other necessary elements are positive interdependence (PI) and individual accountability (IA)

[2], which affect students' commitment towards group work and their own individual contributions. These elements contribute to structuring collaboration and minimizing its potential negative effects [15]. Other aspects should also be considered to understand how CL can result in learning, such as social cohesion and motivational issues [16]; adequate motivating goals regulate the collaborative process [1].

During design, the designer should aim at creating the needed mechanisms and conditions for collaboration. For this task, macro-script components need to be taken into account. For instance, PI and IA may be created by macro-script collaborative structures such as Think Pair Share or Jigsaw [7], [17]; on the other hand, social cohesion may be created by additional activities that affect or prepare collaboration, such as team-building activities or group self-evaluation [16]. Thus, macro-script components, which organize collaborative activities across task distribution, possibly changing groups and time schedules [13], create the context to situate and promote the socio-cognitive processes that collaborative interactions are expected to elicit.

Therefore, while micro- and macro-scripts have been distinguished in the literature [6], the objective of this paper is to support the creation of complete collaboration scripts including information about both granularity levels. This will be achieved by providing designers with guidelines, based on available design knowledge, about the configuration of scripts by combining macro- and micro-level components.

The approach of using design knowledge to support designers has already been adopted in different research works. For instance, a certain type of design patterns, Collaborative Learning Flow Patterns (CLFPs) [7], has been proposed as a way of capturing design knowledge about the configuration of macro-script. On the other hand, [9] provides a conceptual framework to describe design knowledge about interaction and interaction patterns, with respect to the students' learning state. Therefore, this ontology can be used to provide guidelines for the configuration of micro-scripts.

These two research works and others have shown that it is possible to use software tools, capable of using design knowledge, to support designers in the configuration of CL sessions. Figure 1 shows these works and different components of macro- and micro-scripts. As some concepts may appear in both types of scripts, these are overlapping, such as *activities*. This paper is focused on the use of design knowledge from micro- and macro-script levels, combining both perspectives in the design task.

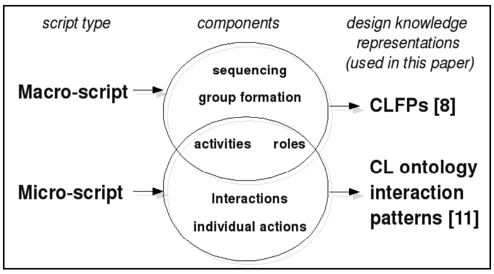

Figure 1. Components of macro- and micro-scripts.

2. Ontology support for the design task

The design problem described in Section 1 can be tackled by the use of CL script authoring tools, as those cited above. In order to capture design knowledge that can be used by such tools, this paper takes the approach of ontological engineering, which has been shown to be useful for design of learning scenarios [12]. This paper proposes an integrative ontology for macro- and micro-scripts based on the studies of CLFPs [7] and the CL ontology [9]. This integrative ontology is used to model: (a) the configuration of different types of components that compose a collaboration script. These components range from micro-script level to macro-script level, and can be understood as building blocks, according to [18]. And (b) the description of complete collaboration scripts, created by configuring those components.

To model both independent components and complete scripts, the ontology is focused on two types of information. The first refers to the description of the expected tasks and behavior of the participants. With respect to this, the main concepts on the ontology, shown in Table 1, describe the events of different granularity that compose the time structure of the script. These concepts are a conceptualization of those shown in Figure 1 and enable the seamless integration of micro-scripts in macro-script structures. This type of knowledge has already been modeled with existing ontologies. For instance, [19] propose an ontology to describe CSCL scripts, which uses semantic information to promote interoperability between different CSCL systems.

Table 1. Ontology concepts to model the time structure of CSCL scripts.

Macro-script components	Phase	A phase describes the activities for the involved groups [1]. We assume that groups may only change between phases. Only one phase can be active at any given time in a CL group.
Macro-script components	Group activity	Assigns activities to groups: each group may have a different activity [11].
Macro-script components / Micro-script components	Session	A unit with specific learning objectives. Represents an abstraction of a learning activity, whose specific communication model is not given.
Micro-script components	Interaction pattern	A sequence of necessary or desired interactions [9]. Describes the communication model of a session.
Micro-script components	Interaction	Individual interactions that compose an interaction pattern [9].

The second type of information deals with the script's pedagogical model: an explicit description of the purpose and expected benefits of each component of the script, which explain the rationale for their concrete configuration, according to the design principles described in Section 1. Again, this information refers to both independent components and complete scripts. In this way, the ontology can be used also to document the design decisions taken during the construction of the script. While the first type of information may be regarded as the core of a script, the ontology presented here is focused mainly on the rationale of the script's design: the ontology explicitly indicates the CL mechanisms that foster collaboration and interactions between participants. Such CL mechanisms include creating adequate conditions related to the environment, grouping policies, etc.; or establishing certain goals that promote PI and AI. The ontology is intended to may be used regardless of the final formalization (such as IMS-LD) of scripts or CLFPs.

This connection between macro-script components (which create CL mechanisms) and micro-script components (interactions and interaction patterns) is explicit in the ontology and the scripts created with it. Therefore, the objective of the ontology presented here is twofold: first, propose a structure to describe CL pedagogic mechanisms in scripts; second, capture design knowledge about a number of scripts components and CLFPs, as discussed below. Section 3 illustrates, through an example, how these mechanisms have been included in the ontology. However, before we examine them, we will describe the sources of design knowledge used in the current version of the ontology, created with the Hozo ontology editor [20].

Design knowledge can be extracted from different sources, including educational theories and learning patterns [21]. Precisely these two sources have been used to build the ontology presented here: we have integrated design knowledge captured in CLFPs [7] and the aforementioned CL ontology developed by [9]. This ontology describes fine-grained interaction patterns and events that may occur in a CL scenario. Additionally, it provides information about the learning benefits that can be expected from those interactions, and the requirements for participants to engage in them. We use this information to provide guidelines concerning the design at micro-script level (see Figure 1). The design knowledge captured in the CL ontology is extracted from well-established learning theories [9], such as Cognitive Apprenticeship and Anchored Instruction. Currently, eight interaction patterns are available in the ontology.

In order to collect design knowledge about components of coarser granularity (i.e., macro-script components) we have analyzed a type of e-learning pattern: Collaborative Learning Flow Patterns (CLFPs) [7]. CLFPs, which are extracted from best practices in collaborative learning, tackle the problem of organizing phases, groups and activities in order to increase the chances of effective interactions. CLFPs have already been formalized in a computer-interpretable format, IMS-LD, to be used in a CSCL script authoring tool, Collage [7]. However, IMS-LD does not allow capturing, in a computer-interpretable format, the rationale for the configuration of CLFPs. As we will see in Section 3, the ontology proposed in this paper tackles this limitation. Currently the Jigsaw and Think Pair Share patterns are available in the ontology.

Using CLFPs and the discussed CL ontology, we have described and connected design knowledge about micro- and macro-script components. In this way, the integrative ontology proposed here aims at supporting the design of scripts that contain information about both levels. Next section will discuss how to use this ontology.

3. Application example

In order to investigate the feasibility of our approach, we have developed a web application that makes use of the propose ontology. This application enables a designer to select one CLFP (which describe macro-script components) and configure in detail the activities that compose its structure. Therefore, it extends the functionality implemented in Collage [7] by including micro-script component configuration: interaction patterns can be embedded in the activities of the CLFP. To support this task, the application uses the ontology to provide suggestions about the most adequate interaction patterns, according to the pedagogical method of CLFPs.

To illustrate the use of the ontology, we will consider the case of the configuration of a script based on the Jigsaw pattern. The Jigsaw pattern, as defined in [7], proposes a series of phases in which students work on a problem that can be divided into several subproblems. Figure 2 (right) shows the activity flow of a Jigsaw devoted to the acquisition of knowledge concerning two subtopics, as represented by the developed tool (see Figure 2-A). Figure 2 (left) shows an excerpt of the proposed ontology, showing a simplified version (due to space limitations) of the definition of the second and third phases of the Jigsaw. The tree-like representation, created with the Hozo editor, indicates *part-of* relationships among concepts (the text over each box indicates the *role* of a concept within its container). Additional relationships among concepts are depicted by the arrows. Relationships are completed with cardinality information, indicating the number of necessary elements (as suggested by the CLFP). This excerpt

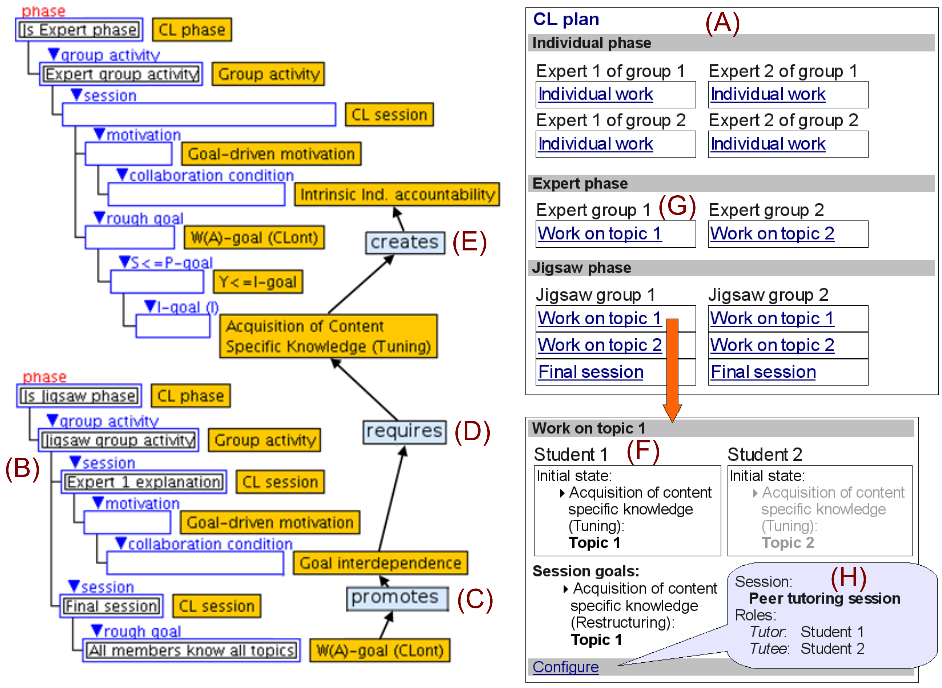

Figure 2. Left: Representation of the Jigsaw model created with Hozo ontology editor [20]. Right: Two representations of the script in the developed application.

of the ontology describes macro-level *phases* and activities (or *CL sessions*), connected to group and individual learning goals.

The description of the rationale of the design, which is the main objective of the ontology, can be illustrated in Figure 2 (left). The last phase of the Jigsaw, or *Jigsaw phase*, consists of a series of CL sessions (see Figure 2-B) in which each *Expert* explains to his/her group mates the topic s/he has previously worked on. In order to motivate students in these activities, the Jigsaw uses the following mechanism: in previous activities, each student became expert only on one topic; thus, students need all their group mates to achieve the global objective of the Jigsaw. In this way, this global objective (see Figure 2-C) *promotes goal interdependence*, a subclass of positive interdependence. Learning objectives, defined as *W(A)-goal* in the CL ontology [9], allow the connection with micro-script components, as discussed below.

On the other hand, the objectives of the *Jigsaw phase* also create the motivation for the previous phase, or *Expert phase*, in which experts on the same topic work together. In this case, the mechanism used to promote students collaboration is the following: each *Expert*'s responsibility for the *Jigsaw phase* creates conditions to make each individual accountable for their work. Therefore, the global objective of *Jigsaw phase* *requires* (Figure 2-D) certain individual learning goals (*I-goal* [9]) for the *Expert phase*: each *Expert* must reach the *Jigsaw phase* with the needed knowledge to fulfill their responsibility. These goals *create* (Figure 2-E) the motivation for collaboration: students must solve possible misconceptions that may exist after the *Individual phase.*

Therefore, all the phases are strongly related. Describing the learning goals of each activity and, consequently, the presumed initial state of the involved students (see Figure 2-F) facilitates the connection between macro-script components with micro-script ones. Due to space limitations, we have discussed only relationships based on goals (*W(A)-goal* or *I-goal*), but other mechanisms exist, such as those related to group changes between phases or characteristics of the environment.

The connections between macro- and micro-script components are used in the developed application to provide the designer with recommendations about the most suitable interaction patterns for each activity. For instance, when a designers searches for a suitable interaction patterns for the activities in the *Jigsaw phase* (Figure 2-G), the system suggests to configure them as "peer tutoring sessions" [9], indicating also the appropriate role distribution (represented in Figure 2-H).

This suggestion is based on the knowledge captured by the ontology. The tool compares (a) the activity objectives with each interaction pattern objectives, and (b) the participants' expected initial state with those required by the interaction patterns. The expected initial state depends on each participant's previous activities; in the example, each student is initially expert on a different topic, as shown in Figure 2-F. Thus, the suggestion reflects the best match between the interaction patterns and the activity considering its objectives and the learners' expected initial state. Other interaction patterns, such as "cognitive apprenticeship", would pursue different sets of objectives; in the case of "distributed cognition", the students do not satisfy the required initial state [9]. However, "peer tutoring" requires additional initial skills too: self-expression skill. This information can be used to update the learning objectives of the previous activities (*Expert phase)* or to decide the group formation policy [9].

4. Conclusions

In this paper we have presented an ontology that combines design knowledge from macro-script and micro-script design approaches. Integrating two different sources of design knowledge, this ontology can be used to develop software tools that support inexperienced designers in the task of creating pedagogically sound collaboration scripts. To show how this ontology may be employed, we have developed a simple application that allow designers to configure the detailed interactions expected to occur within complex CL structures such as CLFPs.

The objective of the ontology is to describe the design rationale of collaboration scripts. In Section 3, we show that this information can be used by software tools to provide design support. We have seen one possible kind of support: the selection of interaction patterns suitable for different activities embedded in macro-scripts. Describing micro-script components of a collaboration script can also help understand the learning process of each student, or facilitate the identification of inconsistencies in the design: requisites which are not fulfilled or learning paths that do not reflect the global objectives of the collaborative scenario. Therefore, different approaches could be implemented to process the ontology and the knowledge contained in it, in order to provide different kinds of support for designers.

The tool shown here is only one example of the type of applications envisioned with the ontology. Through an example of usage of this tool, this paper has illustrated the technical feasibility of capturing and using design knowledge in the proposed ontology to support the design task. Based on this, future work will be aimed at integrating the ontology in more sophisticated authoring tools, which provide more flexibility in the configuration of macro-scrip features (i.e., not only using CLFPs) and functionality to export scripts to formats such as IMS-LD. However, further evaluation with target users is still necessary. Finally, several aspects of collaborative learning that have not been discussed in this paper, including the effect of assessment or of social and group-work skills, should be also included in the ontology.

References

[1] P. Dillenbourg. Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner, editor, Inaugural Address, Three Worlds of CSCL. Can We Support CSCL?, pages 61–91. Open Universiteit Nederland, Heerlen, 2002.

[2] J. W. Strijbos, R. L. Martens, and W. M. G. Jochems. Designing for interaction: Six steps to designing computer-supported group-based learning. Computers & Education, 42(4):403–424, 2004.

[3] Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., Häkkinen, P. & Fischer, F. (2007). Specifying computer-supported collaboration scripts. International Journal of Computer-Supported Collaborative Learning, 2(2-3), pp.211-224.

[4] Kollar, I., Fischer, F., & Hesse, F. W. (2006). Computer-supported collaboration scripts - a conceptual analysis. Educational Psychology Review. 18 (2), June 2006 , pp. 159-185(27).

[5] Weinberger, A., Fischer, F., & Stegmann, K. (2005). Computer-Supported Collaborative Learning in higher education: scripts for argumentative knowledge construction in distributed groups. Proceedings of CSCL 2005: The Next 10 Years! (pp. 717-726) Mahwah, NJ: Lawrence Erlbaum Associates.

[6] Dillenbourg, P., Hong, F. (2008), The mechanics of CSCL macro scripts. International Journal of Computer-Supported Collaborative Learning, 3(1), pages 5-23.

[7] Hernández-Leo, D, Villasclaras-Fernández, E.D., Asensio-Perez, J.I., Dimitriadis, Y., Jarrín-Abellán, I.M., Ruiz-Requies, I. and Rubia-Avi, B (2006). COLLAGE: A collaborative Learning Design editor based on patterns. Educational Technology and Society, 9 (1), 58-71.

[8] Dalziel, J. (2003). 'Implementing learning design: the Learning Activity Management System (LAMS)'. Proceedings of the ASCILITE 2003 conference, Adelaide.

[9] Isotani, S. & Mizoguchi, R. (2007) Deployment of Ontologies for an Effective Design of Collaborative Learning Scenarios. In Proceedings of CRIWG, LNCS 4715, 223-238.

[10] Conole, G., Fill, K.: A Learning Design Toolkit to Create Pedagogically Effective Learning Activities. Journal of Interactive Media in Education. Available at http://jime.open.ac.uk/ (2005)

[11] IMS Global Learning Consortium, Inc., IMS Learning Design Specifications, 2003

[12] Mizoguchi, M., Bourdeau, J. (2000). Using Ontological Engineering to Overcome AI-ED Problems. International Journal of Artificial Intelligence in Education, 11(2), 107-121.

[13] Dillenbourg, P. (2004) (Ed). Framework for integrated learning. Kaleidoscope Network of Excellence deliverable D23.5.1.

[14] Johnson,D. W., & Johnson,R. T. (1994). Learning together and alone: Cooperative, competitive, and individualistic learning. Boston: Allyn and Bacon.

[15] Veenman, S., van Benthum, N., Bootsma, D., van Dieren, J., & van der Kemp, N. (2002). Cooperative learning and teacher education. Teaching and Teacher Education, 81, 87 103.

[16] Lehtinen, E., Hakkarainen, K., Lipponen, L., Rahikainen, M., & Muukkonen, H. (n.d.). Computer supported collaborative learning: a review. CL-NET project. Retrieved December 19, 2008, from http://www.comlab.hut.fi/opetus/205/etatehtava1.pdf

[17] Panitz, T. (n.d.). Collaborative versus cooperative learning: a comparison of the two concepts which will help us understand the underlying nature of interactive learning. Retrieved December 17, 2008, from http://pirun.ku.ac.th/~btun/pdf/coop_collab.pdf

[18] Hernández-Leo, D., Harrer, A., Dodero, J.M., Asensio-Pérez, J.I., Burgos, D. A Framework for the Conceptualization of Approaches to "Create-by-Reuse" of Learning Design Solutions Journal of Universal Computer Science. 13(7):991-1001, October 2007.

[19] Papakonstantinou, A., Demetriadis, S., Bassiliades, N. (2007) Ontology development for computer supported collaborative learning scripts, Balkan Conference in Informatics (BCI 2007), vol.1, 2007, pages 491-500.

[20] K. Kozaki, Y. Kitamura, M. Ikeda, and R. Mizoguch (2002). Hozo: An Environment for Building/Using Ontologies Based on a Fundamental Consideration of Role" and "Relationship", Proc. of EKAW2002, pp.213-218, Sigüenza, Spain, October 1-4.

[21] Koper, R. (2005). An Introduction to Learning Design. In Koper, R. & Tattersall, C. (Eds.), Learning Design, a Handbook on Modelling and Delivering Networked Education and Training, Heidelberg: Springer, 3-20.