Representing Computer-Supported Collaborative Learning macro-scripts using IMS Learning Design

Davinia Hernández-Leo[1], Daniel Burgos[2], Colin Tattersall[2], Rob Koper[2]

[1] School of Telecommunications Engineering, University of Valladolid
Campus Miguel Delibes, 47011 Valladolid, Spain
dherleo@ulises.tel.uva.es
[2] Educational Technology Expertise Centre, Open University of the Netherlands,
PO Box 2960, NL-6401DL, Heerlen, The Netherlands
{daniel.burgos, rob.koper}@ou.nl

Abstract. IMS Learning Design (LD) is a specification that aims at computationally representing any learning process. However, the possibilities of LD to represent collaborative learning scenarios are being questioned by the Computer-Supported Collaborative Learning (CSCL) community. In this paper we analyze the LD support to realize CSCL macro-scripts, which describe flows of coarse-grained activities. We first identify the requirements of the scripts for their representation using LD and, then, study the possibilities of LD to support each of these needs by means of two significant scripts that representatively feature the requirements. The paper indicates the conclusions from this analysis showing the capacity of LD notation to express CSCL macro-scripts but also considering the support of related specifications and tools.

Keywords: IMS Learning Design, Computer-Supported Collaborative Learning, Collaboration Scripts, Learning Management Systems

1 Introduction

Computer software for supporting scripted Collaborative Learning (CL) is designed with the aim of scaffolding social interactions among participants [1]. CSCL macro-scripts structure CL scenarios by defining the composition of groups, the distribution of roles and resources as well as the coordination of the activities that make up the learning process [2], [3], [4]. Up to now, these scripts are “hardwired” in specific CSCL applications. This approach has clearly many drawbacks mainly related to time and cost efforts in development. To overcome these problems, a promising approach is to formalize the scripts so that they are automatically interpreted by an engine integrated in a learning management system (LMS). This paper focuses on the computational representation of CSCL macro-scripts (hereafter “scripts”).

In order to computationally represent the scripts we propose the use of IMS Learning Design specification (LD). LD is broadly accepted as de facto standard to formally model interoperable Units of Learning (UoL). The specification was designed so that UoLs can describe any teaching-learning process [5]. However, the

LD support for implementing CSCL scripts is not clear. Since LD is a recent specification (2003), there are not significant examples and efforts that show the possibilities of LD for CSCL. Besides, there is a lack of clarity regarding which characteristic of the scripts should be expressed by the notation itself as opposed to which requirements can be supported by tools or even other related specifications. Although partial work has been already accomplished [3], [6], a more complete and systematic analysis is needed. As a consequence, some researchers are proposing alternative languages to describe CL scenarios [7].

In this paper we systematically analyze the support of LD to implement the main requirements of CSCL macro-scripts. Therefore, Section 2 identifies the educational design requirements of scripts and illustrates them by means of a significant script: Universanté [8]. Two scripts that extensively feature the requirements are used to analyze the implementation of the requirements with LD. The section collects the results of this analysis concerning Universanté script. Finally, Section 3 concludes the paper by confronting the differences between the requirements that can be satisfied by the LD notation and the needs that can be solved using tools or related specifications.

2 Expressing CSCL macro-scripts requirements with IMS LD

This section presents the educational design requirements of CSCL scripts, which we have identified in the CSCL literature: mostly current research on framing the components and mechanisms of scripts [2] and other complementary sources such as [3], [9], [10], [11]. Table 1 describes the requirements which include common collaborative learning mechanisms related to group composition, role/resource distribution and coordination. Significantly, all the requirements are representatively featured in two of them: Universanté and ArgueGraph [1], [2], [8]. Universanté, which exploits socio-economics and cultural differences for teaching community health to students of different countries [8], is used in Table 1 to illustrate the requirements.

Altogether, the main drawback of scripts is their associated "risky" flexibility restrictions [1]. Inflexible extrinsic constraints, such as the duration of activities, can spoil a satisfactory enactment of the learning scenario [9]. It requires modifications on the fly regarding the time structure, the resources or even the activities themselves and their order. These flexibility requirements are being deeply analyzed concerning adaptive situations for individual learning [6], [12]. Nonetheless, a common flexibility-demanding characteristic that significantly appears in CSCL scripts refers to *flexible group composition*. A typical problem of CL is the variability of students' participation. It is often impossible to guess the precise amount of participants that are attending a particular session, if they will be an even or odd number, whether some of them will join the class afterwards or cannot participate in a specific moment [9]. These situations require unexpected group composition modifications.

Table 2 presents the lessons learned when expressing the requirements of the Universanté script with LD. The table also includes selected excerpts of suggested coding (the complete LD package, also of ArgueGraph, their use case narratives and activity diagrams are available on-line at http://gsic.tel.uva.es/collage/scripts).

Table 1. Requirements of CSCL macro-scripts, Universanté script

Requirements	***Description***	***Illustration: Universanté script***
Hierarchy of groups	CSCL scripts typically make use of groups forming hierarchies, i.e., groups may be composed of other (smaller) groups or different (individual) roles	There are country groups and thematic groups. Each thematic group is composed of case groups
Group size	Defining the desired number of group members is perhaps the most common suggestion of scripts regarding group composition. They usually recommend keeping group size small for short activities because, for example, there is not enough time for the group to become effective. However, larger groups are adequate in long scenarios	(See amount of groups and group formation policies. Since there are at least two countries, each case-group has (at least) 2 persons of different countries. Since there are 2 cases per theme, each thematic-group has (at least) 4 persons. Since there are four cases, each country-group has (at least) 4 persons)
Amount of groups	Many scripts require a certain number of groups or at least a minimum or maximum amount so that the dynamics they propose are afforded	At least two different thematic groups, two case groups (per thematic) and two country groups
Group formation policies	Depending on the scenario groups should be heterogeneous or homogenous to be more effective. The groups can be formed either by the students themselves or by the teacher by referring to existing common features (e.g. gender, age) or simply using a random assignment policy	Each case group is formed of at least 1 participant per country
Dynamic group formation	Some CSCL scripts need some groups to be formed at runtime. That is, group assignment may depend on the result of a previous activity	(Not applicable)
Role distribution	In a CSCL script participants may assume one or more roles at the same time (e.g. one of the students in a group is assigned to the role "scribe"). In addition, participants can switch their roles with other participants (e.g. rotation of roles)	Each person belongs to three different types of groups, which implies playing specific roles depending on their "case", "theme" and "country"
Resource distribution	The amount of resources and their distribution may depend on the number of groups, roles and participants	All case descriptions are distributed evenly among all case groups
Flow of CL activities	The main problem of the activity coordination falls into the synchronization of groups and roles through the activities: a person may belong to a group in a certain activity and to another group in the following one (then she probably needs to wait for the rest of the members of her second group in order to start the second activity)	- Within each case group, all participants discuss a clinical case using a discussion forum; regularly the case groups with the same thematic gather in the same discussion forum and identify common points and differences between the cases; […] - Within each country group, the members of each thematic group in turn present (face to face) a synthesis of their case experience; - Within each thematic group, the members of each country group create a fact sheet concerning the thematic status in their country; […] - Within each country group, the members of each thematic group in turn present their fact sheets; […] - Within each country group, the member of each thematic group modify the fact sheet according to the methodological comments; […]
Floor control	While working together in the same activity, learners' actions are sometimes guided or constrained according to floor control mechanisms (e.g. a model of turn-taking when modifying an artifact)	Fact sheets and health strategies are shared artefacts that require floor control mechanism to ensure data consistency.
Flow of artefacts	Artifacts (e.g. a document) are often created by an individual or a group. They may be used in different activities and by different individuals or groups of the same script	Since the fact sheets are created until they are finally made available to the teacher, they are used in discussions within theme groups, presented within country groups, commented by the teacher, and modified by their authors.

Table 2. Computationally representing the requirements using LD, Universanté Script

Require-ments	***Involved LD elements and attributes***	***Illustrative excerpts, supposing that there will be 2 countries and 4 participants per country, i.e. 2 thematic groups comprising 2 case groups***
Hierarchy of groups	Groups are modeled using roles, which can be bound to several persons. Roles can be nested, indicating that a role is divided in sub roles.	Each thematic group comprises several case groups. <roles> <learner identifier="R-thematic-group-cancer" min-persons="4"> […] <learner identifier="R-case-group-breast_cancer" min-persons="2"> […] </learner> <learner identifier="R-case-group-lung_cancer" min-persons="2"> […] </learner> </learner> <learner identifier="R-thematic-group-aids" min-persons="4"> <learner identifier="R-case-group-pregnant" min-persons="2"> […] </learner> <learner identifier="R-case-group-drug_addict" min-persons="2"> […] </learner> </learner> <learner identifier="R-country-group-switzerland" min-persons="4"> […] </learner> <learner identifier="R-country-group-cameroon" min-persons="4"> […] </learner> […] </roles>
Group size	Role attributes min-persons and max-persons specify the required minimum and maximum numbers of persons bound to the role.	
Amount of groups	Each group can be modeled as a role. An alternative is using the role attribute create-new, which indicates that multiple instances of the role (and their sub roles) can be created during runtime [3].	
Group formation policies	This requirement cannot be formally specified but it can be added as information of the role in the referenced resource.	<learner identifier="R-case-group-breast_cancer" min-persons="2">[…] <information> <tem identifier="I-relation-case-country-groups" identifierref="R- relation-case-country-groups " /> </information> </learner> […] The resource "R- relation-case-country-groups" can be a text file indicating "Each case group is composed of at least 1 participant per country"
Role distribu-tion	Persons can be bound to one or several roles in the same run of the UoL. In this example each person should be bound to one country group and one case group (and thus to one thematic group). The moment in which they are playing each role is specified in the learning flow using the role-part element. To explicitly indicate that persons can be matched exclusively to one of the sub roles (e.g. case groups within a thematic group) LD provides the role attribute match-persons.	
Resource distribu-tion	The resources can be associated to activity-descriptions or to environments, referenced in turn by other LD elements depending on the distribution needs.	In this example, an environment per "case description" (a learning-object) is defined. An activity-structure per case is also defined. Each activity-structure references one of the environments and a common learning-activity explaining the task. Each activity-structure is bound to a role in different role-parts of the same act.
Flow of CL activities	The flow of activities is expressed in the method. A method contains one or more plays, which are modelled according to a theatrical play with acts and role-parts. The plays run in parallel. Acts together with conditions (and also notifications) determine whether, when, and for what roles activities and resources need to be available.	This example requires a method with five acts. Each act contains a role-part per role of the "type" of role that corresponds to each phase. In the cases that the activities are performed by persons belonging at the same time to two groups (E.g. "within each thematic group, the members of each country group create a fact sheet"), it is necessary to add conditions with two expressions of type is-member-of-role (see figure 1) <if> <and> <is-member-of-role ref="R-thematic-group-cancer" /> <is-member-of-role ref="R-country-group-switzerland" /> </and> </if> <then> <show> <class class="C-fact-sheet-cancer-switzerland" /> </show> <hide> <class class="C-fact-sheet-cancer-cameroon" /> <class class="C-fact-sheet-aids-switzerland" /> <class class="C-fact-sheet-aids-cameroon" /> </hide> </then>
Flow of artefacts	Properties can be used to model individual and shared artefacts. Global-elements and monitor services are used to set and view the value of their own or that of others properties. These elements are referenced by the different activities that require the artefacts.	The value of the properties modelling the facts sheets can be set and viewed by the participants by means of global-elements in the several activities. In this excerpt users can modify their fact sheet. <html […]> <div class="C-fact-sheet-cancer-switzerland"> <p>Please modify the fact sheet (Cancer status in Switzerland) according to the methodological comments: </p> <ld:set-property ref="L-fact-sheet-cancer-switzerland"/> </div> […] </html>

3 Conclusions

CSCL macro-scripts aim at structuring collaborative learning processes of coarse-grained activities. Their requirements are shaped around the fact that they involve groups and multi-roles characteristics. The many possibilities of LD to support the identified needs have been tested by means of two scripts (Universanté and ArgueGraph Scripts) that significantly feature the requirements. Returning to the different types of requirements, we summarize now how they are addressed by the notation itself and/or by other specifications and tools:

– The LD roles component and its related elements and attributes together with the joint use of properties and conditions provide constructs to computationally represent several group composition requirements (mainly hierarchy of groups, group size and dynamic group formation). The notation provides limited support for the formal specification of the number of groups and group formation policies. However, these requirements as well as an enhanced realization of the others can be supported by related *administration tools* (and also supporting tools such as grouping services) in combination with eventual group composition specifications.
– Similarly, role distribution relies on the constructs offered by the roles component, in this case complemented with the coordination of role-parts, in each of which a participant may play different roles. In addition, *supporting tools* may define specific roles implying different privileges when using the tools. Rotation of roles can be realized by rotating activities or by using mechanisms eventually provided by the *players*. The distribution of resources is facilitated by the coordination of role-parts but also through the possibility of referencing resources to different elements of LD such as activity-descriptions or environments. The use of properties or *supporting tools* also provides another means of resource distribution.
– Coordinating the flow of CL activities is feasible using the LD method and conditions. The flow of artifacts between activities can be attained by employing properties, global-elements and monitor services (as well as other specialized *supporting tools*) conveniently referenced by other LD elements. The consistency of shared artifacts is ensured by jointly held properties. Moreover, sophisticated floor control mechanisms can be realized by using *supporting tools*.
– Flexibility requirements are also tackled by both the LD notation and its implementation in tools. The main attributes of roles that enable flexible group compositions are min-persons, max-persons and create-new. Further flexibility is provided by the capabilities of LD to support adaptation and the distinction between abstract descriptions (UoLs) and specific instantiations (runs). This distinction affords new developments allowing modifications to runs in progress.

Concluding, computationally representing CSCL macro-scripts using the LD interoperable notation provides the following benefits. Firstly, they can be repetitively and automatically processed. In addition, they can be reused in different settings and with different participants. And, furthermore, they can be easily adjusted to support other learning scenarios by using LD-compliant authoring tools. Current development in this area aims at teacher-friendliness and is focused on visual representations and the reuse of learning design solutions [13]. These editors should hide the computer representational details laid out in this article.

Acknowledgments

Davinia Hernández-Leo acknowledges partial financial support to the Spanish Ministry of Education and Science project TSI2005-08225-C07-04, Autonomous Government of Castilla and León project VA009A05, Kaleidoscope NoE FP6-2002-IST-507838 and, especially, the Research Fellowship Program of the University of Valladolid. This program facilitated a visiting Research Fellow at the Educational Technology Expertise Centre (OTEC) of the Open University of the Netherlands. The authors would like to thank the rest of the members of the OTEC centre and of the GSIC/EMIC group at the University of Valladolid.

References

1. Dillenbourg, P.: Over-Scripting CSCL: The Risks of Blending Collaborative Learning With Instructional Design. In: Kirschner, P. A. (eds.): Inaugural Address, Three Worlds of CSCL. Can We Support CSCL? Open Universiteit Nederland, Heerlen (2002) 61-91
2. Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., Fischer, F.: Specifying Collaboration Scripts. International Journal of Computer-Supported Collaborative Learning (submitted)
3. Hernández-Leo, D., Asensio-Pérez, J.I., Dimitriadis, Y., Bote-Lorenzo, M.L., Jorrín-Abellán, I.M., Villasclaras-Fernández, E.D.: Reusing IMS-LD Formalized Best Practices in Collaborative Learning Structuring. Advanced Technology for Learning. 2 (4) (2005) 223-232
4. Dillenbourg, P., Jermann, P.: Designing Integrative Scripts. In: Fischer, F., Kollar, I., Mandl, H., Haake, J. (eds.): Scripting Computer-Supported Collaborative Learning. Cognitive, Computational, and Educational Perspectives. Springer, New York (2007) 277-302
5. Koper, R., Olivier, B.: Representing the Learning Design of Units of Learning. Educational Technology & Society. 7 (3) (2004) 97-111
6. Koper, R., Burgos, D.: Developing Advanced Units of Learning Using IMS Learning Design Level B. Advanced Technology for Learning. 2 (4) (2005) 252-259
7. Vignollet, L., David, J. P., Ferraris, C., Martel, C., Lejeune, A.: Comparing Educational Modelling Languages on a Case Study. IEEE International Conference on Advanced Learning Technologies, Kerkrade, The Netherlands IEEE Computer Society (2006) 1149-1150
8. Berger, A., Moretti, R., Chastonay, P., Dillenbourg, P., Bchir, A., Baddoura, R., Bengondo, C., Scherly, D., Ndubre, P., Farah, P., Kayser, B.: Teaching Community Health by Exploiting International Socio-Cultural and Economical Differences. European Conference on Computer-Supported Collaborative Learning, Maastricht, the Netherlands (2001)
9. Dillenbourg, P., Tchounikine, P.: Flexibility in Macro-Scripts for CSCL. Journal of Computer Assisted Learning. 23 (1) (2007) 1-13
10. NISE: Doing CL: CL Structures. http://www.wcer.wisc.edu/archive/cl1/CL/ Last accessed: http://www.wcer.wisc.edu/archive/cl1/CL/ (1997)
11. Johnson, D. W., Johnson, R. T.: Learning Together and Alone: Cooperative, Competitive, and Individualistic Learning. 5 edn. Allyn and Bacon, Needham Heights, MA, USA (1999)
12. Zarraonandia, T., Dodero, J.M., Fernández, C.: Crosscutting Runtime Adaptations of LD Execution. Educational Technology and Society. 9 (1) (2006) 123-137
13. Hernández-Leo, D., Harrer, A., Dodero, J.M., Asensio-Pérez, J.I., Burgos, D.: Creating by Reusing Learning Design Solutions. International Symposium on Computers in Education, León, Spain (2006) 417-424