Conceptual and Computational Issues in the Formalization of Collaboration Scripts

Andreas Harrer, University of Duisburg-Essen, harrer@collide.info
Lars Kobbe, Knowledge Media Research Center, l.kobbe@iwm-kmrc.de
Nils Malzahn, University of Duisburg-Essen, malzahn@collide.info

Abstract: Collaboration scripts aim at facilitating social and cognitive processes of collaborative learning by shaping the way learners interact with each other. Computer-supported collaboration scripts generally suffer from the problem of being restrained to a specific learning platform and learning context. Researchers are therefore aiming for a formalization of collaboration scripts on both a conceptual and a computational level. A recently developed framework allows to describe collaboration scripts using a small number of components (participants, activities, roles, resources and groups) and mechanisms (task distribution, group formation and sequencing). Based on these, a formal, graphical modelling tool has been developed and tested with several example scripts.

Introduction

Collaboration scripts are based upon the scripted cooperation approach, originally developed by O'Donnell, Dansereau, Hall and Rocklin (1987), which differs from other collaborative learning approaches in that it aims at facilitating social and cognitive processes of the learners by means of providing a carefully designed structure for small group interaction. Collaboration scripts have become fairly popular within educational science, especially in the domain of computer-supported collaborative learning (CSCL), where they have been used in various settings, including face-to-face, web-based as well as mobile contexts. So far, collaboration scripts are hardwired to a specific learning environment and context, limiting their portability, scalability and adaptability as well as making it difficult to research the specific effects of scripts on learning processes and outcomes. In order to resolve these issues, researchers aim for a formalization of scripts both on a conceptual and a computational level.

The current de facto standard for the specification of executable learning processes is IMS Learning Design (IMS/LD) (Koper and Tattersall, 2005). The editing level of IMS/LD is mainly a textual that is close to the technical specification: there are some (mostly) tree based editors, which try to abstract from the XML-based language, but there is no "native" graphical representation. The suitability for formalization of complex collaborative learning scenarios has been critically discussed in the CSCL area (Hernandez et al., 2004; Miao et al., 2005), which might imply that for the CSCL researcher an adapted and specialized conceptual model with explicit concepts like groups and dynamic assignment of resources is needed.

As a conceptual and computational formalization goes hand in hand, this task was recently approached by an interdisciplinary research team of computer scientists, psychologists and educational scientists, leading to a common framework for the specification of scripts (Kobbe et al., subm.). Consolidating the conceptual analyses of Dillenbourg (2002), Dillenbourg and Jermann (in press) and Kollar, Fischer and Hesse (in press) with new insights into the composition of scripts, the specification proposes a comprehensive yet economic description of collaboration scripts, featuring five basic components and three basic mechanisms. The components include the individuals that participate in a script (*participants*), the *activities* that they engage in, the *roles* they assume, the *resources* that they make use of and the *groups* they form. Script mechanisms help to describe the distributed nature of scripts, that is, how activities, roles and resources are distributed across participants (*task distribution*), how participants are distributed across groups (*group formation*) and how both components and groups are distributed over time (*sequencing*).

Based on these components and mechanisms, we have started to develop a tool for the formal, graphical modelling of scripts. A formal, graphical model combines the advantages of computer-readable, formal data structures for issues of portability, scalability and adaptability with those of graphical representations: A graphical model usually is much easier to read and analyze (Shu, 1998), in particular by non-computer-scientists whose judgement and input is indispensible for the conceptual validation of the computational model. Furthermore, some features of collaboration scripts are best conveyed graphically, such as parallel activities, repetitions, and conditional branches. And last but not least, a graphical modelling tool serves as a prototypical authoring environment for the design of new collaboration scripts. To a target user group of educational researchers and practitioners with little programming knowledge, a graphical authoring tool will be much more user-friendly than a programming editor.

Modelling Scripts

Today, a large variety of highly sophisticated graphical notation systems especially in computer science exist. Yet none of these have found to be fully appropriate for the modelling of collaboration scripts (Harrer & Malzahn, 2006). Inspired by the semantics of statecharts diagrams, we have developed a graphical notation which allows a formal modelling of collaboration scripts based on the conceptual framework mentioned above. Since the framework concepts are familiar to the CSCL researcher, we decided to provide a graphical construct for each of the theory based concepts. The components are represented by iconic objects while the dynamic aspects (i. e. the mechanisms) are represented as links between these objects. The inscription of the links is used to define the mechanim more specifically, such as different group formation strategies (e. g. by group size, by number of groups, or by values attributed to the participants, like gender or skills). This direct mapping from conceptual to computational level is further supported by syntactical constraints imposed by the editing tool.

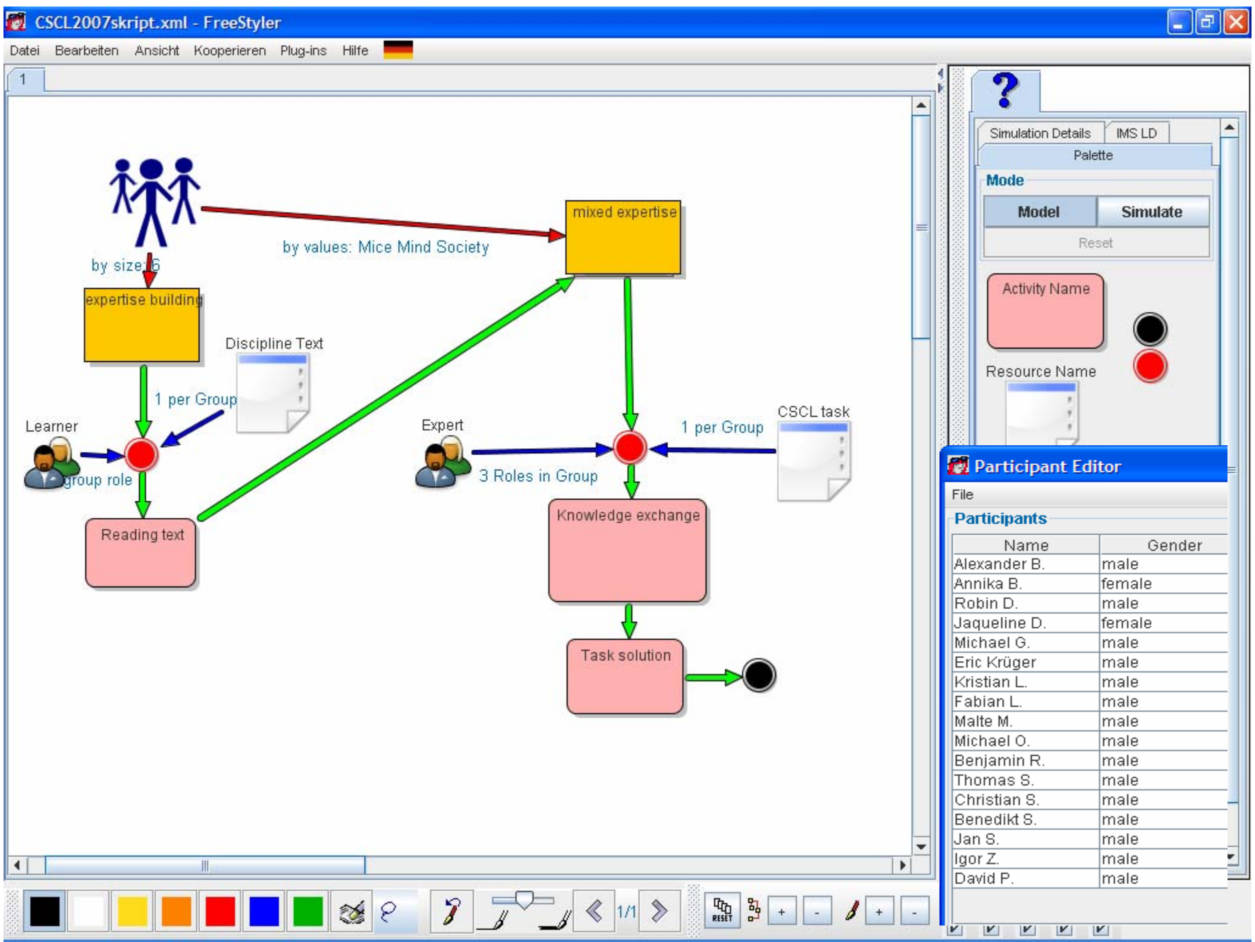

Figure 1. Jigsaw script modelled with the proposed graphical modelling language in the editing tool.

The first prototype we developed was tested with several scripts from the literature to show the soundness of our integrated approach. As an example and to introduce the constructs of the modelling language, figure 1 shows a graphical model of a Jigsaw classroom scenario (Aronson et al., 1978). The participants, symbolized by a group of people, are split into groups of six to study together a text which is characteristic for one the disciplines with respect to CSCL (technology, cognition, society) that they are supposed to become an expert of. The assignment of the text to the particular groups is represented by the iconic document associated to the graphical learning flow (green arrows) with red circles. Since all of the students are on equal footing, they are all learners in this phase of the script. So they are assigned the role *Learner* for this part of the script. The reddish boxes with rounded edges on the end of the sequencing (i.e. the learning flow) arrows represent the activity to be conducted by the groups. After reading the text, all participants are reorganized (red arrow) to form groups of mixed expertise. Each of the groups consists of three members with one from each of the expert groups before. This is stated by the inscription *by value Mice Mind Society* on the red arrow and refers to attributes gained by the resource assignment in the previous assignment. A similar attribution with roles would have also been possible if we had distinguished more than one role before. Finally, the mixed expert groups exchange their discipline specific knowledge within their groups and solve the task that could not be solved otherwise. Thus, the script comes to an end which is symbolized by the black circle.

Conclusions

The presented graphical modelling language based on a theoretical framework addresses CSCL's particular needs for dynamic group formation and role assignment. Thereby it allows the non-computer scientists to model the scripts using an iconic expression level without losing conceptual soundness of the approach. This enables researchers and practitioners to use the model in several ways (Miao et al., 2005): planning, refinement, discussion and exchange of models with peers and students to improve the overall performance of learning. The soundness of the modelling language was tested by several researchers of CSCL, asking them to model their favourite scripts as well as observing a teacher modelling her lesson design with the tool.

In our current work, we are extending our tool so that the models can be simulated in the editing tool and an export to IMS/LD is possible. This will improve the quality of CSCL scripts and learning process further because potential (static) problems with the script can be more easily detected and even dynamic problems can be analyzed by "what-if" analyses. An example for this is the detection of group formations that are not fully compliant to the specified formation strategy, such as distributing 17 students to groups of size 6. This will be detected automatically by the simulation and will be highlighted to the designer of the learning activity. The IMS/LD export will enable novices in IMS/LD to create and configure computer-supported learning scenarios using de-facto standard tools available in the LD developer community, yet modelling in a high-level graphical modelling language using the more convenient conceptual model of CSCL scripts.

References

Aronson, E., Blaney, N., Stephan, C., Sikes, J., & Snapp, M. (1978). *The jigsaw classroom*. Sage Publications.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL?* (pp. 61-91). Heerlen, NL: Open Universiteit Nederland.

Dillenbourg, P., & Jermann, P. (in press). Designing integrative scripts. In F. Fischer, H. Mandl, J. Haake, & I. Kollar (Eds.), *Scripting computer-supported collaborative learning: Cognitive, computational and educational perspectives.* New York: Springer.

Harrer, A. & Malzahn, N (2006). Bridging the gap - towards a graphical modelling language for learning designs and collaboration scripts of various granularities. *Proc. ICALT 2006,* edited by Kinshuk, R. Koper, P. Kommers, P. Kirschner, D. Sampson & W. Didderen. IEEE Press, Los Alamitos, CA, pages 296-300.

Hernandez, D., Asensio, J.I., & Dimitriadis, Y., (2004) IMS Learning Design Support for the Formalisation of Collaborative Learning Flow Patterns. *Proc. ICALT 2004*, IEEE Press, Los Alamitos, CA,, pp. 350-354.

Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., & Häkkinen, P. (submitted). Specifying Computer-Supported Collaboration Scripts.

Kollar, I., Fischer, F., & Hesse, F. (in press). Computer-supported cooperation scripts – A conceptual analysis. *Educational Psychology Review*.

Koper, R., & Tattersall, C. (2005): Learning Design: *A Handbook on Modelling and Delivering Networked Education and Training* Springer Verlag.

Miao, Y., Hoeksema, K., Hoppe, H. U., & Harrer, A., (2005) CSCL scripts: Modelling features and potential use. *Proc. of CSCL 2005*, Lawrence Earlbaum Associates, Mahwah, NJ, pp. 423-432.

O'Donnell, A. M., Dansereau, D. F., Hall, R. H., & Rocklin, T. R. (1987). Cognitive, social/affective, and metacognitive outcomes of scripted cooperative learning. *Journal of Educational Psychology, 79*(4), 431-437.

Shu, N. C. (1998): *Visual Programming.* Van Nostrand Reinhold.

Acknowledgements

This research project was partially funded by the Kaleidoscope Network of Excellence.