FROM GRAPHICAL LEARNING DESIGNS TO COMPUTER-SUPPORTED COLLABORATION SCRIPTS
A Rapid Development Process

Sara Streng

Media Informatics Group, Ludwig-Maximilians-Universität, Amalienstr. 17, 80333 Munich, Germany

Karsten Stegmann, Frank Fischer

Department of Psychology, Ludwig-Maximilians-Universität, Leopoldstr. 13, 80802 Munich, Germany

Heinrich Hussmann

Media Informatics Group, Ludwig-Maximilians-Universität, Amalienstr. 17, 80333 Munich, Germany

Keywords: Learning designs, CSCL, Functional framework.

Abstract: In the field of collaborative learning, instructional designs are often used to structure learning processes. In order to support educationalists during the design of such learning scripts, graphical learning design editors have been developed, e.g. MoCoLaDe. The learning designs can then be exported using a formal specification, such as the markup language IMS-LD, for further use. This paper presents the concept, development and evaluation of a functional framework, which interprets these learning designs to automatically generate an executable server and an extensible client. These components can immediately be used to run simulations, which help educationalists and programmers to discuss the design of the user interface (which is the only programming task left). Furthermore, the framework provides a user interface for tutors to easily control the applications in real scenarios.

1 INTRODUCTION

Research on technology-based collaboration scripts has been very successful in terms of the development of a broad range of scripts that effectively foster processes and outcomes of computer-supported collaborative learning (e.g. Baker & Lund, 1997; Weinberger et al., 2010; Kollar et al., 2007). However, technology-based collaboration scripts are usually tested exclusively within the environment for which they were developed. Neither the transfer to other experimental platforms nor the transfer into practice has been managed systematically so far.

To overcome this gap between research and practice, Kobbe and colleagues introduced a framework to describe computer-supported collaboration scripts (Kobbe et al., 2007) using a small but still comprehensive number of components

and mechanisms of collaboration scripts. The components are participants, activities, roles, resources, and groups; the mechanisms comprise task distribution, group formation, and sequencing. The idea behind this approach is that each single collaboration script from a broad variety of script types can be described as a specific configuration of these components.

On the basis of the descriptive framework from Kobbe and colleagues (2007), the graphical modelling tool MoCoLaDe for designing collaboration scripts has been developed (Harrer & Malzahn, 2006). This modelling tool produces an IMS-LD file, i. e. a file that can be read by all learning platforms that support the IMS Global Learning Consortium (2003) Standards. Another tool that creates a computer readable script description in IMS-LD is Web Collage (Villasclaras-Fernández et al., 2009). However, instead of an

implementation based on script formalization, specific script patterns (e.g. a jig saw pattern) and assessment capabilities are offered to the user. Finally, CeLS has to be mentioned as tool that comprises capabilities for designing as well as enacting collaboration scripts on the basis of IMS-LD (Ronen & Kohen-Vacs, 2010).

Wecker and colleagues (2010) propose a different approach. Instead of creating an abstract machine-readable description of a collaboration script, S-COL provides ready-made instances of collaboration scripts for any kind of web-based environment. S-COL does this by means of a general interface to any kind of web pages (cf. Wecker et al., 2010). Thereby, collaboration scripts can easily be used on different learning platforms.

MoCoLaDe, Web Collage and CeLS enable educationalist to design their collaboration scripts and enact them in IMS-LD compatible learning environments. The great benefit of this approach is the low barrier for educationalist. Collaboration scripts can be implemented without extensive programming skills. However, this implies losing control over the user interface implementation. This can be a strong disadvantage in terms of human computer interaction (HCI). Empirical studies on HCI aspects of collaboration scripts provide evidence that the design of the user interface (e.g., the workspaces of learners on a tabletop display) has significant effects on how learners follow a collaboration script (Streng et al., in press).

The S-COL approach provides full control over the GUI. However, educationalists have to assign a programmer to realize the script. The complete 'logic' of the collaboration script as well as the GUI has to be programmed.

Furthermore, current approaches mainly address web-based learning environments. Applications for novel devices like smartphones, tabletop displays or tablet computers are not considered. In addition, IMS-LD based learning environments are usually full-featured platforms rather than simple-to-use 'out of the box' solutions that are needed in classroom settings. For instance, while user management is needed in university-wide learning environments, such features are unnecessary for teachers who realize a peer-review script in class.

Against this background, we propose a solution that provides full control over the GUI, but incorporates the capability to read and enact IMS-LD descriptions of collaboration scripts. The solution is a functional framework that can be used to create lightweight learning applications from collaboration scripts, which were authored with the

graphical modelling tool MoCoLaDe.

This paper presents the XSS framework along with the resulting development process and several auxiliary tools that support the user interface design as well as classroom operation.

2 XSS FRAMEWORK

2.1 Development Process: From Learning Designs to Executable Distributed Applications

Our goal is to provide a framework that supports the development of scripted collaborative learning applications and provides adequate tools for educators, programmers and tutors throughout the development process as well as in classroom operation. The application uses a client/server-based approach. Each learner interacts with one client, which provides the graphical user interface. The server is the central component. Knowing the collaboration script as defined in the imported IMS-LD document, it controls the sequencing of learning phases, is in charge of phase transitions and responsible for the communication between server and clients.

The framework automatically generates as much of the functionality as possible based on the information specified in the IMS-LD. Figure 1 illustrates the development process starting from the creation of the learning design. First, an educator creates a learning design using a graphical editor (in this case MoCoLaDe). The learning design is then exported as IMSL-LD document. Using the XSS Framework, the learning design can then be imported and transformed into a fully functional server and a preliminary client ('*Dummy Client*'). The Dummy Client shows the task description and learning material for each phase and is capable of communicating with the server and other clients. However, the means to execute the learning task are not provided, yet. For example, if the learner is asked to write a summary of a text in a given phase, the text and the instructions are shown in the dummy client, but the capability of writing the summary is still missing. Thus, during the user interface design phase programmers or user interface designers need to finalize the Dummy Client.

Once the client is fully functional, the application is ready for classroom operation. A tutor or administrator can start and control the server whereas the learners are provided with clients.
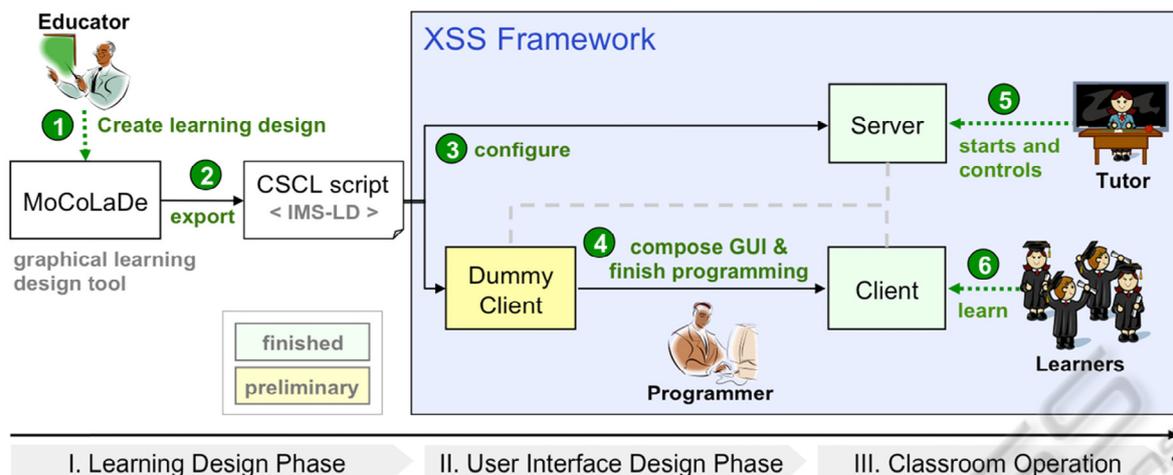
Figure 1: eXtremely Simple Scripting (XSS) development process.

The framework is implemented in the programming language C#. Client and server applications are executable (.exe) files, which can easily be started by double clicks on any Windows machine. No prior installation or configuration is required on the target devices except for IP address and port numbers, which can be entered on the welcome screen.

2.2 Framework and Application Controls

The XSS Framework provides a set of auxiliary tools that support the involved parties in controlling the framework or finalized application. These tools are used during the user interface design phase (step 4 in Figure 1) as well as in classroom operation (step 5 in Figure 1).

The entry point is the *Simple Starter* (Figure 2), which guides the user along a sequence of tools. Depending on the development stage, the Simple Starter is either used by the programmer and/or educationalist responsible for designing the learning application, or the tutor once the application is fully developed.

The first step is starting the server. This step will load a script (in IMS-LD), interpret it and automatically generate an executable server as well as a dummy client for simulation purposes. Afterwards the server is started and ready to accept client requests.

The second step is starting the client. Depending on the development process, this is either a Dummy Client or a fully functional client that is ready for classroom use. *Dummy Clients* are click-through prototypes, which show (1) the instructions and (2)

learning material for each phase. The client-server-communication, including exchange of material between clients as specified in the script, is already implemented. For instance, in a script for two learners, who build a dyad, a common task for each learner could be reviewing the partner's outcome of the previous phase. The functionality of exchanging learning materials (e.g. summaries and reviews) between the clients is already integrated in the automatically generated Dummy Client. Therefore, by starting multiple clients the entire collaboration script can be run through. Instead of opening one window for each client, which can get very confusing, the *Multi Client* tool can be used to clearly arrange and manage multiple client windows. Running simulations allows creators of learning designs to verify that their model was correctly interpreted. Furthermore, as the Dummy Clients are closer representations of the final application than any model, the simulation mode supports interdisciplinary teams of educationalists, designers and programmers to discuss the functionality and design of the user interface in each phase.
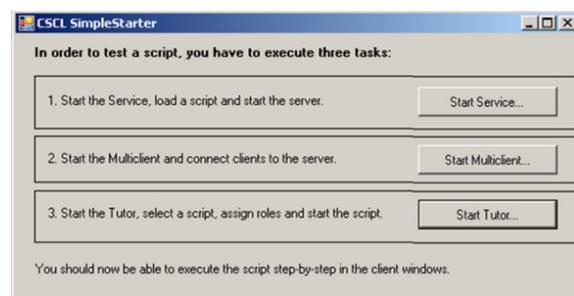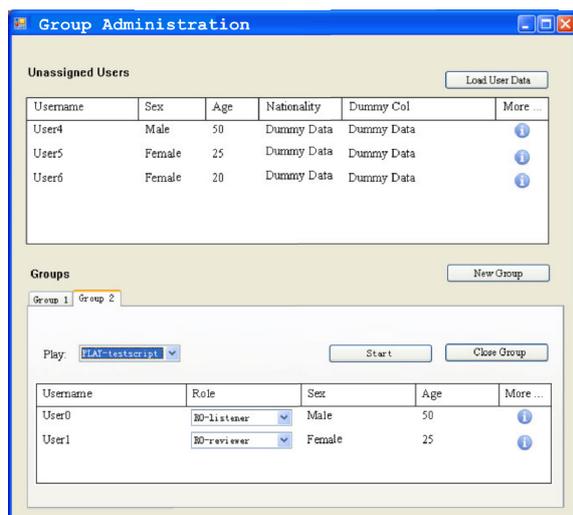

Figure 2: SimpleStarter.

Figure 3: Group Administration Tool for tutors.

The third and final step is starting the *Group Administration Tool*, which supports the group building process. The Group Administration Tool is split into two sections: the (unassigned) user section at the top and the group section at the bottom (see Figure 3). In the user section, the tutor can import user data from a csv-file (using the button 'Load User Data') or enter data manually (a click on the 'i' button in the 'More…' column opens a form). The types of information (e.g. sex, age, nationality, etc.) are those specified in the learning design (see Figure 4).

3 EVALUATION

A user study was conducted to assess the usability and functionality of the framework. Furthermore, the study should show whether the framework is flexible enough to support different types of learning designs. During the development of the framework several learning designs were kept in mind and regular tests were conducted using the MURDER script. However, as a proof of concept, we needed to see if the framework works with other learning designs as well.

In this study, steps four and six of the development process (see Figure 1) were omitted. Thus, the dummy client was not finalized and the simulation was used to test framework and application.
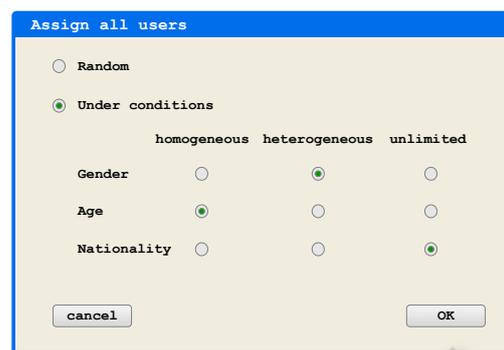

Figure 4: Automatic assignment of users to groups.

3.1 Case Study Design

We invited four doctoral students (aged between 26 and 34, three of them female) of the learning sciences, who each work with different collaboration scripts. In the study, the participants modelled their individual learning design using MoCoLaDe and then imported the design in order to create an executable learning application.

The user study was divided in two parts. In the first session the experimenter gave a MoCoLaDe training. In the beginning of the training session there was a lecture on MoCoLaDe's functionality and components. Afterwards the participants were given laptops and modelled an exemplary collaboration scripts following a step-by-step tutorial given by the instructor. Finally, the participants modelled their own collaboration scripts and exported it to IMS-LD.

The second part of the study was scheduled for the following day. At first there was a short introduction to the framework. Then the development process was demonstrated using the exemplary script from the day before. Afterwards each participant imported the IMS-LD file created in the last session, ran the application and tested the Multi Client as well as the Group Administration Tool. In the end, there was a short questionnaire, followed by an open discussion about missing features as well as other suggestions for improvement.

3.2 Results

3.2.1 Functionality

The doctoral students modelled a peer-questioning (King, 1997), two peer-review (Weinberger et al., 2005) and an argumentative collaboration (Stegmann et al., 2007) script. All of these learning

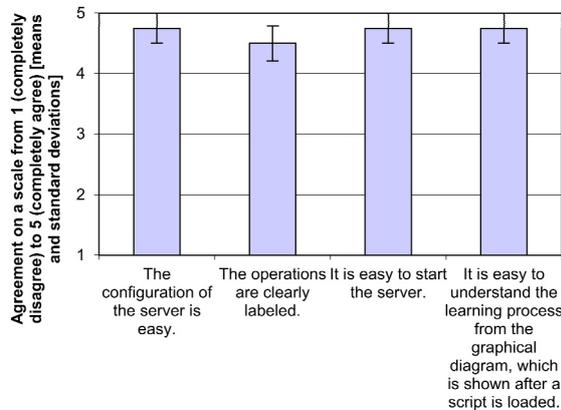scripts were successfully modelled in MoCoLaDe, imported and run as simulation using the XSS framework.

Figure 5: Usability assessment of the server.

However, IMS-LD currently does not support iterative cycles of activity sequences or conditional behavior (if … then do activityX else do activityY), both of which occurred in the user study. To solve the problem the participants' collaboration scripts were simplified. Multiple cycles were reduced to one cycle and instead of conditional behaviour a default activity was defined.

The participants were overall satisfied with the functionality. However, they asked for additional features during classroom operation. Similar to the Multi Client, which provides an overview during simulations, a similar tool could allow observations in classroom settings. Instead of arranging multiple Dummy Clients, an *Observation Tool* should provide an overview of all running client windows and update changes in real time. In addition, it should be visualized which clients are in the same group.

Furthermore, in the final questionnaire three participants suggested adding the functionality to export data. In the open discussion this suggestion was further discussed. As the individual steps as well as the results of the learning session often need to be evaluated, it should be possible to export a protocol and the learning results.

3.2.2 Usability Assessment

To assess the usability of the user interfaces, the final questionnaire contained several questions regarding the server (see Figure 5) and Group Administration Tool (see Figure 6). The questions (for example 'The operations are clearly labelled') had to be answered on a Likert scale from 1

(completely disagree) to 5 (completely agree). All questions were answered very positively with averages of 4.5 or higher.
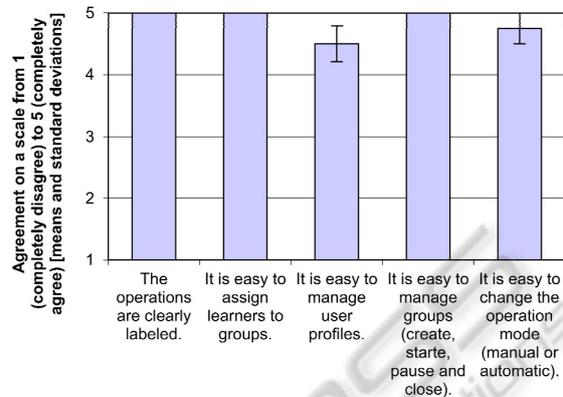

Figure 6: Usability assessment of the Group Administration Tool.

In the discussion one participant proposed to combine the three windows (Simple Starter, Group Administration Tool and Multi Client) into one single window. Whether this is still feasible once additional features (e.g. the Observation Tool) are implemented, and how the windows could be merged to maintain a clear arrangement needs to be considered carefully.

4 CONCLUSIONS

Our case study provided evidence that the XSS framework in combination with MoCoLaDe could be a helpful tool for educationalist. Doctoral students in the field of educational science were able to author their collaboration scripts with MoCoLaDe and test the functionality of the collaboration scripts using the simulation functionalities provided by the XSS framework. In addition, the XSS framework provides not only tools for the development of collaboration scripts, but can also serve as runtime environment for collaboration scripts. Collaboration scripts implemented with XSS can be distributed as out of the box solutions. No additional applications or servers are necessary.

Our results also showed some limitations of the current approach. So far IMS-LD does not allow cycles of activities or conditional sequences of activities. As cycles and conditions are fairly common in collaboration scripts, the framework functionality should be further extended beyond the capabilities of IMS-LD.

The user study further showed that the educationalist require more functionalities from the runtime-environment. During the learning phase, they would like to monitor the activities of the learners. Finally, after the collaboration, archiving the process data and outcomes of collaborative learning was requested.

Our future work has to focus on the transfer to and use in practice more strongly. We have to provide evidence that the development of the GUI for collaboration scripts using the XSS framework is as easy as promised. Finally, evidence is needed for the whole concept that stems from an end-to-end case: how to come from a learning design via dummy clients to an executable out of the box application used in a classroom.

Moreover, the XSS framework and the S-COL approach may be combined. While the XSS provides build-in IMS-LD support, S-COL provides an interface to all common web-based learning environments. The GUI of collaboration scripts can be optimised collaboratively by HCI-designers and educationalists and easily transferred to different learning environments.

ACKNOWLEDGEMENTS

REFERENCES

M. Baker, and K. Lund, "Promoting reflective interactions in a CSCL environment." Journal of Computer Assisted Learning, 13(3), 1997, pp. 175-193.

A. Harrer and N. Malzahn, "Bridging the Gap - Towards a Graphical Modeling Language for Learning Designs and Collaboration Scripts of Various Granularities." Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06), Kerkrade, the Netherlands: IEEE Computer Society Press, 2006, (pp. 296-300).

IMS Global Consortium. "IMS Learning Design XML Binding", 2003. 1.0 Specification.

A. King, "ASK to THINK-TEL WHY: A model of trans-active peer tutoring for scaffolding higher level complex learning", Educational Psychologist, 1997, 32, 221–235.

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

I. Kollar, F. Fischer and J. D. Slotta, "Internal and external scripts in computer-supported collaborative inquiry learning, Learning and Instruction, 17(6), December 2007, pp. 708-721.

M. Ronen, D. Kohen-Vacs and N. Raz-Fogel, "Adopt & adapt: structuring, sharing and reusing asynchronous collaborative pedagogy". In Proc. Conference on Learning sciences (ICLS '06), International Society of the Learning Sciences, 2006, pp. 599-605.

K. Stegmann, A. Weinberger and F. Fischer, „Facilitating argumentative knowledge construction with computer-supported collaboration scripts". International Journal of Computer-Supported Collaborative Learning, 2(4), 2007, pp. 421-447.

S. Streng, K. Stegmann, C. Wagner, S. Böhm, H. Hussmann and F. Fischer (in press), "Supporting argumentative knowledge construction in face-to-face settings: from ArgueTable to ArgueWall", Proceedings of the conference on Computer-Supported Collaborative Learning 2011.

E. D. Villasclaras-Fernández, D. Hernández-Leo, J. I. Asensio-Pérez and Y. Dimitriadis, "Incorporating assessment in a pattern-based design process for CSCL scripts", Computers in Human Behavior, 25(5), 2009, pp. 1028-1039.

C. Wecker, K. Stegmann, F. Bernstein, M. J. Huber, G. Kalus, S. Rathmeyer, I. Kollar and F. Fischer, "S-COL: A Copernican turn for the development of flexibly reusable collaboration scripts", International Journal of Computer-Supported Collaborative Learning, 5(3), 2010, pp. 321-343.

A. Weinberger, K. Stegmann and F. Fischer, "Computer-supported collaborative learning in higher education: Scripts for argumentative knowledge construction in distributed groups", In T. Koschmann, D. Suthers, & T. -W. Chan (Eds.), Computer Supported Collaborative Learning 2005: The Next 10 Years, Mahwah, NJ: Lawrence Erlbaum, 2005, pp. 717-726.

A. Weinberger, K. Stegmann and F. Fischer, "Learning to argue online: Scripted groups surpass individuals (unscripted groups do not)". Computers in Human Behavior, 26, 2010, pp. 506-515.