Adaptation Patterns in Systems for Scripted Collaboration

Anastasios Karakostas, Stavros Demetriadis
Aristotle University of Thessaloniki, POBOX 114, 54124, Thessaloniki, Greece
Email: akarakos@csd.auth.gr, sdemetri@csd.auth.gr

Abstract: This work presents a prototype method (DeACS) for identifying useful adaptation patterns to be embedded in systems for adaptive collaboration scripting. Collaboration scripts are didactic scenarios that guide and support the collaborative learning activity while adaptive collaboration scripting is the idea that computer-supported collaboration scripts can be adapted during run time, to provide learning experiences tailored to individual and group characteristics. An adaptation pattern is described as a well-defined adaptation process that can be initiated by the system when specific conditions are identified during script implementation. In order to model the proposed method twelve postgraduate students were engaged in a pyramid-type collaboration script and the analysis of the learning experience provided the basis for identifying a number of possible adaptation patterns. The paper discusses also next steps for advancing the design and evaluation of adaptation patterns in systems for scripted collaboration.

Introduction

Collaborative learning is important for students both for social and cognitive reasons (Dimitracopoulou & Petrou, 2003). However collaborating students might fail to engage in productive learning interactions when left without teachers' support (Hewitt, 2005). Scripted collaboration is the idea that collaboration can be guided by didactic scenarios, aiming to engage students in fruitful learning interactions. Lately there have been efforts for the formalization of collaboration scripts (Kobbe et al., 2007) and the development of computer-based environments to support scripted collaboration (e.g. Bote-Lorenzo, Gomez, Dimitriadis, Asensio, & Jorrin, 2008).

Nevertheless, adjusting the script level of granularity and flexibility emerges as an important issue that affects the outcome of scripted collaboration (Dillenbourg & Tchounikine, 2007). Although a teacher can be flexible enough and adjust various collaboration parameters during script run-time, CSCL systems for scripted collaboration are far from exhibiting a comparable level of flexibility. We have argued elsewhere (Demetriadis & Karakostas, 2008; Karakostas & Demetriadis, 2008) that a possible solution to the script flexibility issue could be the integration of adaptive characteristics to systems for scripted collaboration. "Adaptive collaboration scripting" is, in general, the idea that computer–based environments for scripted collaboration can operate in an adaptive mode in order to significantly improve the learning conditions. Based on the Adaptive Educational Systems (AES) and the Computer-Supported Collaborative Learning systems (CSCL systems), adaptive scripting tailors the learning experience to the needs and characteristics of the individual learner or teams and implements the beneficial adjustments during the specific learning situations emerging from script implementation.

In this work we present a prototype method for identifying possible adaptation patterns within the context of a collaboration script. An adaptation pattern is described as an adaptation process that can be initiated by the system when specific conditions are met during script runtime. The objective is to enhance the flexibility of script implementation and offer adaptive support to students based on their personal and/or group profile. In order to model the proposed method twelve postgraduate students were engaged in a pyramid-type collaboration script and the analysis of the learning experience provided the basis for identifying a number of possible adaptation patterns. In the following, after concisely presenting a conceptual framework for adaptive collaboration scripting, the paper focuses on the implementation of the method and identification of adaptation patterns. Next steps for advancing the design and evaluation of adaptation patterns in systems for scripted collaboration are also discussed.

Background

Scripted Collaboration: Promises and Pitfalls

Researchers have systematically emphasized that collaborating students might fail to engage in productive learning interactions when left without teachers' consistent support and scaffolding (Hewitt, 2005). It has been suggested that the instructor guides the learners' interactions within the group, by implementing an appropriate collaboration script (Kobbe et al., 2007). Collaboration scripts are didactic scenarios that aim to structure and guide the collaborative learning process by specifying the way in which learners interact with one another. Every script can be seen as a sequence of phases with five major attributes each: (a) the kind of task that has to be performed, (b) the composition of the group, (c) the task distribution (among group members), (d) interaction and communication mode, and (e) script's time duration (Kobbe et al., 2007). Scripts can also be distinguished in (a) macro-scripts focused on the organization and the structure of the collaborative activity, and (b) micro-scripts focusing on a more psychological perspective, providing support to individuals for specific activities in order to enhance their socio-cognitive skills (Dillenbourg & Tchounikine, 2007). Furthermore, CSCL scripts are computer-based representations of collaboration scripts where the role of the computer is to provide participants with communication tools and manage script's time and information flows (Tchounikine, 2008).

Implementing CSCL scripts has been reported to result in improved learning outcomes (e.g. Rummel & Spada, 2007; Weinberger, Fischer, & Mandl, 2002). However, CSCL scripting has been criticized for its loss of flexibility (difficulty of modifying a script in run time according to the needs of the instructional situation) (Dillenbourg & Tchounikine, 2007), and also the danger of "over-scripting" collaborative activity (the pitfall of overemphasizing script imposed interactions and constraining natural collaboration) (Dillenbourg, 2002).

Adaptive Collaboration Scripting: Maximizing the Script Benefits

Against this background, we argue that adaptive scripting techniques can be beneficial for learners when embedded in CSCL scripting systems. Such systems could adaptively tailor the learning experience to the needs and characteristics of both the individual learner and the group so that the benefits from the scripted collaboration are maximized. We call such systems "Adaptive Collaboration Scripting systems" or ACS systems and, in general, we suggest that the adaptation methods in ACS systems focus on two major objectives: (a) *enhance peer interaction*: the implementation of user modelling techniques and respective adaptation rules, when appropriate, can help establish conditions of more effective peer interactions (e.g. when recording learners' profile during group formation process to create hetero- or homogenous groups) and (b) *establish conditions of flexible scripting*: adjusting the level of scripting to avoid conditions of over- or under-scripting considering the learners' and/or the groups' needs and characteristics.

Efforts to implement adaptive techniques in CSCL in order to improve the learning experience have already been reported to the literature. For example, Harrer, Malzahn, and Wichmann (2008) combine aspects from CSCL, pedagogical design, and Intelligent Tutoring Systems (ITS) to provide an integrated architecture for supporting collaborative learning activities. Our point of view, however, is to emphasize the need for a generalized conceptual framework of adaptive scripting, considering not only the learner's (or group's) characteristics but also the specific characteristics of the implemented script.

Designing an Adaptive Collaboration Scripting System

Basics of an ACS system

An ACS system should satisfy three major criteria: (a) it is a CSCL system, that is, it somehow supports collaborating groups of students; (b) it includes a user model (learner's cognitive characteristics and preferences), a group model (data relevant to the synthesis and the dynamics of the group) and a script model (computer-based script representation comprising information on specific script characteristics); (c) based on these models, it adapts the representation of the CSCL script by initiating some adaptation pattern when necessary. An adaptation pattern is a process that by taking into account the user, group and script models, aims to adjust the collaboration activity in order to maximize student engagement, satisfaction and, consequently, learning outcomes. For an adaptation pattern at least three issues should be defined: (a) conditions of initiation, (b) aspects of script to be adapted, and (c) processes to be executed.

However we maintain that not any script feature can be candidate for adaptation. The script "intrinsic" constraints (Dillenbourg & Tchounikine, 2007), that is, the core features that give to the script its specific pedagogical character and value, should not be adapted in any way. Only "extrinsic" constraints can be adapted in order to enhance scripted flexibility. Extrinsic constraints can be considered as belonging to either of two categories: (a) "Non-pedagogical", that is constraints without any pedagogical relevance. These can be altered by the teacher and/or the students simply to make the script to better accommodate the conditions of the specific implementation (for example, extending the duration of a phase because of a learner's temporal inability to meet a deadline). (b) "Pedagogical" constraints that can (should) be adapted in order to provide a well suited learning experience (for example, increasing the level of support when diagnosing learners' misconceptions). From this perspective therefore, computerized script representations should clearly define intrinsic and extrinsic (also pedagogical and non-pedagogical) script features, and adaptation patterns should affect only those features characterized as extrinsic.

The DeACS method

Overall the goal of the DeACS method (*De*signing *ACS* systems) is to provide guidance for teachers and designers on how to identify useful adaptation patterns emerging in the context of scripted collaboration, of which a computerized form can presumably be embedded in an ACS system. Developing an ACS system can be a complex process if one wishes to define beforehand and embed in the system all possible adaptation patterns that might occur during script runtime. Furthermore it is difficult to predict if these adaptation patterns would be beneficial for learners when enacted in a specific script implementation. To tackle this issue the DeACS method proposes to implement a number of iterations of the script with a twofold objective: first, to implement and evaluate a certain number of adaptation patterns that the teacher might consider as necessary for the collaborating students (based on the literature or his/her prior teaching experience), and, second, to identify possible helpful adaptation patterns emerging from the expressed needs of students during the scripted collaboration.

Generally the DeACS method proposes three major processes: (a) a top-down process: integration of selected adaptation patterns in the ideal script (the form of the script that the teacher initially wishes to put into practice) based on particular activation conditions, (b) a bottom-up process: identification of adaptation patterns that emerge from students' needs for help, support, adjustments, etc. during script runtime, (c) an evaluation process aiming to assess the added value of the adaptation patterns in the previous two categories. If the evaluation of patterns reveals beneficial impact on student learning then these patterns can become part of the script representation embedded to the ACS system.

Method Implementation in a Pyramid Script

In order to model the DeACS method twelve postgraduate students were engaged in a case-based learning (CBL) pyramid-type collaboration script for two weeks. Generally in a pyramid-type learning each participant works first individually studying any learning material and then participates in a workgroup of a gradually increasing size, to collaboratively process the material from a certain perspective. The objective of our "pyramid" script is to help students develop a satisfactory conceptual understanding of a complex domain by collaboratively analysing relevant cases. In this implementation we used the Moodle LMS to support students from distance and provide an asynchronous discussion board for them to collaborate. The CBL pyramid script had four main and one pre-scripting phase.

Pre-scripting phase: In this phase we identified a number of users' characteristics, administering appropriate questionnaires. The objective was to record (a) the students' experience with CSCL systems and the CBL method, and (b) the students' prior knowledge about the technology-supported pedagogical innovations in secondary education, which was the domain of instruction. The participants' answers showed that two of them were experienced, eight of them had little prior experience and two of them had no experience at all as teacher or learner using CSCL activities and systems. Regarding domain knowledge, three students could be characterized as experts, two had advanced knowledge, and seven of them were novices. We used this information to form the six groups in the second phase of the script and also to assign group moderators in the third phase.

1. Individual study phase (three days): In this phase each student selected and studied two case studies from an online case study database (SITES: M2, 2008) and wrote a report analyzing the cases that he/she had studied.

2. Small group synthesis phase (five days): In this phase, six groups of two were formed (three were heterogeneous and three were homogeneous according to students' prior domain knowledge). The group members discussed the cases that have studied in the first phase in order to expand and also deepen their perspective in the domain and they also submitted a deliverable (answers to a series of questions) recording their common understanding. The group collaboration implemented a "reciprocal contribution" approach: the first group member (acting as "rapporteur") suggested an answer to a question posed by the instructor while the other member (acting as a "reviewer") commented afterwards on the answer. Thus the two students reached a final common answer and they moved on to answer the next question in the list (their roles were changed).

3. Expanded group synthesis phase (five days): In this phase two larger groups of six were formed. Each large group included one member of each small group (dyad) of the second phase. New material was offered to the groups and the task was to discuss asynchronously this new material (question prompts were given) and provide their own proposal on how they would apply technology-supported innovations in secondary education. Two of the more advanced students (one for each larger group) were assigned to be moderators during group asynchronous discussion.

4. Debriefing phase: The debriefing phase was conducted in the classroom. The instructor led the discussion making comments about the quality of the deliverables and of the group collaboration.

After the end of the activity, we organized semi-structured interviews with all participating students in order to evaluate the impact of the pre-selected adaptation patterns (top-down process) and also identify possible adaptation patterns emerging from learners' needs during script implementation (bottom-up process). The content analysis of interviews transcriptions indicated both benefits and drawbacks regarding the implemented adaptation patterns and also the need for two specific new adaptation patterns. In the following we analyze in detail three of the major adaptation patterns (one embedded, two emerged).

Adaptation Patterns Analysis

Pattern 1

Name: Group Heterogeneity based on Prior Domain Knowledge
Key-idea: Formation of heterogeneous groups based on partners' prior domain knowledge is expected to foster improved collaborative learning conditions.
Activation conditions: When students need to work in groups to broaden their understanding in a domain.
Action: Form heterogeneous groups regarding students' prior domain knowledge. Heterogeneous groups should comprise learners whose prior domain knowledge should not be extremely unequal (Wang, Lin, & Sun, 2007).
Specific Implementation and Results: This is a previously known idea for group formation, embedded to the script. Therefore, this pattern implementation is considered as a top-down process. In our activity three heterogeneous groups were formed with one participant expert and one participant novice each. The rest of the groups were homogeneous (two groups with novice participants and one group with medium knowledge participants). Evaluation data, based on students' interviews, showed that two of the three heterogeneous groups had fruitful interactions and their members considered the whole phase as very efficient for them. An expert user mentioned: "*…it was very interesting and useful to see how my partner was thinking…*". The third heterogeneous group showed very inefficient communication, however, we have reasons to believe that this was for reasons unrelated to the group formation process. By contrast, in the three homogeneous groups at least one member in each group believed that the activity was not beneficial for them at all. Overall, our evaluation confirmed that what was initially perceived as helpful adaptation (i.e. forming heterogeneous groups) was indeed beneficial for students (at least for the majority of them) during the collaboration activity.
System Prerequisites: An ACS system that will support this pattern should consist of: (a) an instrument to record prior domain knowledge, (b) a module to form groups based on the principle of heterogeneity.

Pattern 2

Name: Lack of Confidence
Key-idea: Support the novice learners in larger groups in order to be more confident to participate.
Activation Conditions: Whenever a domain novice learner participates in an extended group (more than three teammates).
Action: Provide the novice learners with extra support (adaptive scaffolding) in order to motivate them and improve the quality of their contributions. A possible action is to assign specific roles to novice learners in order to make their participation in larger groups more clear and understandable.
Specific Implementation and Results: This adaptation pattern emerged during the script runtime and, therefore, it refers to the bottom-up process of the DeACS method. During script implementation we observed that the novice learners' participation in the third phase of the script was very low (as opposed to other participants). The novice learners mentioned that this phase was rather complicated for them and, also, that they were not sure how correct their opinions and thoughts were. As a result they preferred most of the time not to actively participate but rather passively attend what other group members contributed. By contrast, expert learners evaluated the third phase as the most beneficial for them (both socially and cognitively) exhibiting high participation and quality of contributions.
System Prerequisites: A system that supports the "lack of confidence" pattern should include a repository with supportive material to be presented to novice learners. For example, guidelines on how to contribute, further explanations to better understand their role during the script phase or other relevant roles to be assigned to the novice participants.

Pattern 3

Name: Advance the Advanced
Key-idea: Adjust the level of script challenge for the sole expert participant in a group in order to offer an interesting learning experience to him/her too.
Activation Conditions: A group (regardless of the number of the members) has one and only expert.
Action: Provide the domain expert participants with extra and advanced domain learning material and/or activities.
Specific Implementation and Results: During the implementation of the script we observed (bottom-up process) that the second phase of the script was not adequately challenging for the advanced learners. The three domain experts in the three heterogeneous groups mentioned that although the collaboration procedure was socially beneficial, it did not add anything new to what they already knew about the domain. It seems that for the sole advanced learner in a group, the lack of same level teammates minimizes the opportunity for interesting interaction and exchange of ideas. A possible adaptation pattern to cope with this situation would be to provide advanced learners with additional and more challenging material and/or activities.

System Prerequisites: A system implementing the "advance the advanced" pattern should include some type of repository with advanced learning material and activities.

Conclusions and Future Work

In this work we presented (a) an introductory conceptual framework for adaptive collaboration scripting and (b) a prototype method (DeACS) for identifying and evaluating the impact of adaptation patterns to be embedded in CSCL systems for adaptive scripting. In order to model the proposed method twelve postgraduate students were engaged in a pyramid-type collaboration script and the analysis of the learning experience provided the basis for identifying a number of possible adaptation patterns. This work analyzed some of these patterns that emerged both from a "top-down" and a "bottom-up" strategy, specifying the characteristics that a system should have to support this type of adaptations.

We are currently working on implementing specific adaptation patterns in a CSCL system for supporting students in scripted collaboration. Our goal is first, to explore the important technical issues related to the implementation of adaptation patterns in existing CSCL systems, and second, to examine the impact of these patterns on student learning when applied in larger scale and in an automated runtime mode.

References

Bote-Lorenzo, M. L., Gomez-Sanchez, E., Vega-Gorgojo, G., Dimitriadis, Y. A., Asensio-Perez, J. I., & Jorrin-Abellan, I. M. (2008). Gridcole: A tailorable grid service based system that supports scripted collaborative learning. *Computers and Education, 51*(1), 155-172.

Demetriadis, S., & Karakostas, A. (2008). Adaptive collaboration scripting: A conceptual framework and a design case study. In F. Xhafa & L. Barolli (Eds.), *Proceedings of the CISIS 2008: 2nd International Conference on Complex, Intelligent and Software Intensive Systems* (pp. 487-492). Los Alamitos, CA: IEEE Computer Society.

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: Open Universiteit Nederland.

Dillenbourg, P., & Tchounikine, P. (2007). Flexibility in macro-scripts for computer-supported collaborative learning. *Journal of Computer Assisted Learning, 23*(1), 1-13.

Dimitracopoulou, A., & Petrou, A. (2003). Advanced Collaborative Distance Learning Systems for Young Students: Design Issues and Current Trends on New Cognitive and Meta-cognitive Tools. *THEMES in Education International Journal*.

Harrer, A., Malzahn, N., & Wichmann, A. (2008). The remote control approach - An architecture for adaptive scripting across collaborative learning environments. *Journal of Universal Computer Science, 14*(1), 148-173.

Hewitt, J. (2005). Toward an understanding of how threads die in asynchronous computer conferences. *The Journal of the Learning Sciences, 7*(4), 567-589.

Karakostas, A., & Demetriadis, S. (2008). Systems for Adaptive Collaboration Scripting: Architecture and Design. *Proceedings of the Adaptive Collaboration Support Workshop of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 7-12). Retrieved March 12, 2009, from http://www.ah2008.org/files/resourcesmodule/@random4875cac18955e/1215679251__Proc_AH2008_WS1_Adaptive_Collaboration_Support.pdf

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), 211-224.

Rummel, N., & Spada, H. (2007). Can People Learn Computer-Mediated Collaboration by Following A Script? In F. Fischer, I. Kollar, H. Mandl & J. Haake (Eds.), *Scripting Computer-Supported Collaborative Learning* (pp. 39-55). New York: Springer.

SITES: M2 (The Second Information Technology in Education Study: Module 2) (2008). Retrieved November 7, 2008, from http://www.sitesm2.org/

Tchounikine, P. (2008). Operationalizing macro-scripts in CSCL technological settings. *International Journal of Computer-Supported Collaborative Learning, 3*(2), 193-233.

Wang, D.-Y., Lin, S., & Sun, C.-T. (2007). DIANA: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups. *Computers in Human Behavior, 23*, 1997-2010.

Weinberger, A., Fischer, F., & Mandl, H. (2002). Fostering computer supported collaborative learning with cooperation scripts and scaffolds. In G. Stahl (Ed.), *Computer support for collaborative learning: Foundations of a CSCL community* (pp. 573-574). Mahwah, NJ: Erlbaum.