Chapter 11 – Discussion

ROLES OF COMPUTATIONAL SCRIPTS

Daniel D. Suthers
University of Hawaii

Abstract: This chapter, which was solicited as a commentary upon the chapters of the computer science perspectives on scripting in the present volume, analyzes different roles that computational scripts are expected to play in collaborative learning. Three roles of computational scripts are identified and discussed: offloading some of the work of managing a collaborative interaction so that learners can focus on the learning task, guiding learners into types of interactions that are expected to be productive for learning, and communicating instructional designs. Several problems for further research are identified, including exploration of the synergy between scripting and representational aids, and investigation of the conditions under which spontaneity of patterns of behavior is a factor in the association of these patterns of behavior with learning. Given issues of learner control and the situated nature of learning, a synthesis of the roles of scripting is suggested that views a script as a proxy by which an instructional expert can participate, along with learners who draw upon the script as a resource, in the accomplishment of a successful collaborative learning episode.

For the purposes of this chapter, collaboration scripts are devices by which participants' actions are regulated towards some ideal. In general, the concept of scripting is independent of computer technology, and can be studied without involvement of technology beyond using (for example) verbal or printed instructions. There are clear advantages to using computational technology, such as support for distance interaction and automated prompting, but the primary variables being studied are not intrinsically properties of computational technology. However, since the section of this book on which the author was asked to comment consists of four chapters on computational approaches to scripting, this chapter treats scripting specifically as a form of computer support for collaborative learning (CSCL).

The author has identified two major strategies for using technology in CSCL (Suthers, 2005; 2006). The *computer-mediated communication* (CMC) strategy treats the technology as a communication channel and tries to increase the richness of that channel as much as possible. The *guide-and-constrain* strategy uses technology to direct the potential actions of participants towards some benefit. Scripting clearly falls in the second strategy, although one chapter in this section (Lauer & Trahasch, this volume) also makes a contribution towards the first strategy, which will be discussed briefly later. Two sub-strategies were identified for guiding or constraining participants' actions in order to benefit learning.

One guide-and-constrain strategy is to remove obstacles to learning. Benefits of collaboration for learning may not be realized because CSCL introduces the additional task of managing the group via CMC. Scripting "offloads" this management, freeing up participants to focus on the problem-solving task. The chapter by Haake and Pfister (this volume) is motivated by this strategy.

The other guide-and-constrain strategy builds on research showing that some specific patterns of interaction are effective for learning, and tries to lead participants into these patterns of interaction. All of the chapters in this section exemplify this strategy to some extent. They differ along a continuum from setting up the conditions from which such interactions are hoped to emerge to explicitly imposing them upon participants. *Macro* or *static* scripts set up the conditions. *Micro* or *dynamic* scripts typically try to enforce the forms of interaction.

Scripts are not just ways to control people or computers. They are designs, and we need ways to communicate designs. By providing computational support for scripts as designed artifacts, we gain advantages such as the ability to easily edit, communicate (transmit, copy), and provide multiple perspectives on these designs. The chapters by Miao, Harrer, Hoeksema, & Hoppe (this volume) and by Haake & Pfister (this volume) make contributions towards computer supported authoring of scripts. Scripts are also resources for learners, a perspective that is not reflected in the target chapters.

The next three sections of this commentary will consider each role of scripting in turn: offloading tasks, fostering productive interactions, and communicating designs. The concluding discussion will point out some limitations of current work and discuss scripts as resources for learners.

1. OFFLOADING TASKS

One motivation for using scripts is to resolve a paradox of collaborative learning. There are known benefits of group learning: cooperation on a divisible task can reduce task load on each individual (Steiner, 1972), and collaboration can increase learning effectiveness through activities that are more difficult to do alone, such as argumentation, explanation, and reflection

(Andriessen, Baker, & Suthers, 2003; Slavin, 1995). However, collaboration imposes an additional task on the learners: in addition to choosing actions within the problem domain and attending to what they are learning from those actions, they must also manage interpersonal relations and group functioning (Whitworth, Gallupe, & McQueen, 2000). Learning may be reduced if less cognitive resources are dedicated to the learning task (Sweller, van Merriënboer, & Paas, 1998). Also, Haake and Pfister (this volume) note that the task of collaboration in a distributed environment itself is unfamiliar. Computer-mediated communication lacks the cues of face-to-face interaction (Clark & Brennan, 1991), increasing the difficulty of managing distributed collaboration. Scripts can help reduce collaborative effort by imposing regularities or providing guidance on how to collaborate based on a theory of how to realize the advantages of collaborative learning.

Yet, scripts do not come without a cost. Dillenbourg (2002) claims that scripts increase the cognitive load of the learner, as the learners have to process the script as well as the rest of their task. If reduction of load is the primary motivation for scripts, then they should be implemented in an unobtrusive manner not requiring the attention of the learner-participant to benefit. However, this recommendation must be thought through carefully. As Sweller et al. (1998) point out, cognitive load can either be extrinsic, intrinsic or germane to the learning task. If the script were intended to offload matters that the learners need not attend to in order to learn (extrinsic load), unobtrusive support is desirable. If the script were intended to focus attention on that which is to be learned or model strategies to be acquired (intrinsic load), then participants would benefit from explicit reflection on and manipulation of scripts (germane load). In general, cognitive resources must be allocated to that which is to be learned, and scripts may be a resource towards such an end.

2. FOSTERING PRODUCTIVE INTERACTIONS

The other use of scripting is based on the belief that some forms of interaction are more effective for learning than others, and that it is worthwhile to make some arrangements for the appearance of such interactions in a session, as they may not occur naturally. These arrangements differ on degree of coercion, which corresponds roughly to the distinction between "macro" scripts, in which the conditions for a session (group organization, task assignment) are set up at the outset of the session (hence remain "static" for the duration of the session), and "micro" scripts, that prompt for or even constrain participants to these effective forms of interaction during a session. To the extent that scripting is coercive or imposed, it relies on the assumption that spontaneity in the production of these patterns of interaction is not a factor in their association with learning. Since this assumption is at the crux of the scripting enterprise, it is worth examining carefully with both theoretical and empirical tools. Further work is required in this area.

The following subsections discuss both the management of conditions for interaction and of the interactions themselves, and include a brief comment on one chapter's contribution towards improving computer-mediated communication.

2.1 Setting up the conditions for interaction

Much work on macro scripting (and some work on micro-scripting) assumes a situation in which a number of students are available (working online) and a decision needs to be made concerning who collaborates on what task. The granularity can range from pairing up students who are working independently for the purposes of one helping the other (micro-scripting) to setting up groups to work on a task in the first place (macro-scripting), our present focus. An overlay student model approach is common in the literature: If a student needs help, a helper is chosen who has recently solved the same problem (e.g., Ikeda, Go, & Mizoguchi, 1997). Ayala (this volume) describes a variation that compares individual to group models (rather than to other individual models), and uses Vygotsky's (1978) "zone of proximal development" (ZPD) as a unifying concept to both group configuration and task assignment at the macro level and pairing of individuals for help at the micro level. The learner's ZPD – the "structural knowledge frontier" – is identified as those knowledge elements "pedagogically related" (a prerequisite relation?) to those believed to be internalized. This set is matched to knowledge elements believed to be already internalized by others – the "social knowledge frontier" – in order to form collaborative groups at the intersection of the structural and social knowledge frontiers. The social knowledge frontier indicates how others in the group can act as mentors for present purposes. Therefore, this assignment is not based on an a-priori identification of teacher and learner: anyone can be mentor as well as learner.

The matching just described is conducted from the point of view of an individual learner: one or more participants are chosen for their ability to help the learner, rather than considering what they can achieve jointly. Vygotsky's claim that every intra-psychological function appears first on the inter-psychological plane suggests a richer basis for group formation that we might explore. Any potential accomplishment of the group can be internalized, so we might consider not just knowledge elements as articulation points between individuals and groups, but also capabilities of the group as a whole.

2.2 Anchored discussion as a context for scripting

Lauer and Trahasch (this volume) make a contribution in the first category of computer support for collaborative learning: improving CMC. These authors identify several limitations of online review of lecture recordings from a learning standpoint, including the optional and limited nature of interaction with the materials and the lack of interaction with other learners. These critiques also apply to the face-to-face lectures themselves: we can question why the authors would want to replicate this problematic didactic tradition online. Replication of a face-to-face genre online might miss the opportunity to design "beyond being there" (Hollan & Stornetta, 1992), leveraging the unique opportunities of the online medium. However, Lauer and Trahasch do seem to have succeeded in making online lectures more attractive than their face-to-face counterpart. Some technical issues are addressed to enable either synchronous or asynchronous "anchored discussion" of the video lecture materials. Parallel, embedded and linked designs are considered following Suthers (2001). In the process of interacting, learners create value in the form of annotations and "script views" that can be exploited for their own or others' learning. So far, this is an expressive media solution, but scripting is brought in to address a perceived problem with unsupervised student interaction.

2.3 Micro-managing interaction

Work on micro-scripting, Lauer and Trahasch's included, typically assumes a situation in which students are already interacting online. There is evidence (or at least the worry) that their interactions will be ineffective without guidance, and the instructor cannot be "present" to provide this guidance (e.g., the interaction is asynchronous, or there are too many students interacting for the number of available instructors). To address this problem, an epistemological commitment as to what constitutes effective learning through collaboration is identified (e.g., Jermann & Dillenbourg, 2003; Weinberger, Reiserer, Ertl, Fischer, & Mandl, 2005) and restrictions on communicative actions in the interface are written to guide or constrain learners to desired interactions (e.g., Baker & Lund, 1997; Robertson, Good, & Pain, 1998).

Lauer and Trahasch's chapter does not offer a theory of what constitutes an effective interaction, but provides an example script in which one learner is the *analyzer* and the other the *critic*, the two alternating as the critic provides constructive criticism on the analyzer's analysis of the lecture materials and the analyst responds to these critiques. The epistemological assumption of this work seems to be that people learn by cognitive processing of content such as exemplified by the analyst and critic roles. Scripting is needed because students do not naturally assume these roles. As previously noted, there is a critical assumption that students will engage in deeper processing even if coerced into doing so.

In this author's view, a promising direction to pursue in Lauer and Trahasch's program is the interplay between the representational solutions (interlinking discourse and content representations) discussed in the previous subsection and scripting that provides guidance and structure to the interaction using these representations. There may be a synergy between scripting and representational aids. For example, Toth, Suthers, and Lesgold (2002) showed that peer assessment rubrics – which may be seen as a form of script – have their greatest effect when designed in conjunction with representational aids (Belvedere's evidence maps). The Toth et al. work made a commitment to epistemological scripting, but the results of Weinberger et al. (2005) suggest that exploration of representational and scripting aids for social interaction may also be valuable.

Ayala's chapter also discusses micro-level coordination of interaction. Students are paired up for short exchanges using reasoning about the learner's ZPD similar to that discussed above, and allowed to interact via a restricted interface that does "not allow discussion out of context of the joint problem" and limits communication to a set of pre-scripted messages. This approach will be discussed further in the concluding section.

3. COMMUNICATING DESIGNS

As computer scientists, the authors are concerned with identifying appropriate computational formalisms for scripts. Finite state automata (FSA) are popular devices due to their simplicity and easily grasped graphical representations. FSA are sets of states connected by transitions that correspond to "input" symbols, which in scripting applications typically stand for actions taken by collaborating participants. For example, participants are classified into roles, and these roles are constrained to make certain moves depending on the state of the interaction. Lauer and Trahasch use nondeterministic finite state automata to formalize scripts. Nondeterminism allows for ambiguity concerning which states result from a given action, providing economy of expression but adding no descriptive power: any nondeterministic FSA can be converted into a deterministic one, usually with the addition of states. Haake and Pfister also use an FSA representation, but they are primarily concerned with the inflexibility of scripts. Scripts that are "hard coded" into a computer program are not easy to change. Their solution is to provide a set of "atomic scripts" and a means for teachers as well as programmers to compose them into larger scripts. "Flexible scripting" focuses on flexibility in authoring, rather than flexibility from the learners' point of view.

The FSA representation is well suited for describing desirable sequences of actions, but does not provide a notation for structural descriptions, such as of group composition or the role of artifacts in an activity. Miao, Harrer, Hoeksema, and Hoppe (this volume) use the Unified Modeling Language (UML) for this purpose. Their work is an extension of the Instructional Management System Learning Design (IMS-LD), which is expressed in UML. Miao et al. critique the IMS-LD for several shortcomings that they address, most notably in modeling groups, the dynamic changing of roles in a group, the existence of artifacts as the product of activities independent of individual persons, and complex control flows. Miao et al. note that attempts to fix this problem by defining "group services" without actually fixing the model are inadequate. (The present author found similar limitations in the IEEE Learning Technologies Standards Committee Architecture & Reference Model when advising that working group in the late 1990's.) Models exist not only to generate the desired behavior: they are also used by people for communicating instructional designs. A dialogue between this line of work and those working on authoring systems (Murray, 1999) is in order. Do Miao et al. or Haake & Pfister model pedagogical knowledge in a manner consistent with how educators think about their practice? This question is not just concerned with the usability of the interfaces provided, but also with whether the very assumptions of the modeling languages (independently of their visual representations) match educators' thinking. The answer may depend on who the educators are. For example, university professors and primary school teachers may have different needs.

It is conceivable that after researchers have improved the expressiveness of a given formalism for scripting collaborative learning, other limitations will be found. To what extent are we able to fully specify a learning situation? For example, a person may play different roles at different moments or even the same moment: These roles are not properties of individuals but are emergent from the group interaction. There will always be some aspect of a learning scenario that any given modeling language leaves out. Scripts are guides and a partial solution: we have no choice but to partner with learners' improvisational abilities. Therefore, we might profitably design computationally supported representations of scripts as resources for the participants in a learning situation, in addition to designing them as an educator's notation for a high-level computer program. For example, Carell, Herrmann, Keinle, and Menold (2004) describe a line of work on script-like representations with which participants articulate, negotiate and reflect on their own processes.

4. SCRIPTS AS RESOURCES

As suggested throughout this chapter, a major issue for scripts is their flexibility and degree of coercion. The language of deontic logic is often used in describing micro-level scripting: formalisms are defined indicating which actions are obligatory and which are permitted. Perhaps this is a sign that our technical solutions are heavy handed. Collaborating learners need help in reducing the complexity of simultaneously coordinating the group and interacting via CMC, and they need to be guided towards situations that are likely to be productive. We technologists, influenced by the formal nature of our tools, have responded to these needs by restricting the learner to actions that our formalisms permit or that our artificial intelligences can understand. When people are interacting via computer media that we design, we can attempt to restrict their interactions, but we should distinguish what we can do from what we should.

A well-known result in the field of CSCW showed the limitations of scripting interactions in the workplace. The Coordinator (Flores, Graves, Hartfield, & Winograd, 1988), an FSA-driven script for coordination of work related communications in an office setting, was accepted in formal and hierarchical organizations, but was too rigid for more creative organizations. What does this result portend for scripting of interaction in educational settings, particularly where we seek to foster active inquiry (a form of creativity) on the part of the student? Students may "play the game", complying with the authority of the instructor by using scripted systems if required to do so, but will their interactions be as effective for learning if they are merely following along to play the game posed to them? Furthermore, what have we lost by disallowing "out of context" interactions? Might "off task" conversation contribute to learning, for example through affective means, or unexpected discoveries? These are fundamental issues for scripting, and the lack of empirical evaluation of these and related issues in the present chapters indicates that further research is needed.

It is essential to help the learner with the guidance they need without excessive loss of control on their part. This hypothesis is motivated partly by affective reasons such as learners' sense of control, but is also based on the stance that learning is an interactionally and contingently achieved accomplishment (Koschmann, Zemel, Conlee-Stevens, Young, Robbs, & Barnhart, 2005). Scripts can't capture all the contingencies under which people accomplish learning, because learning comes as a result of a huge variety of situations, so it is unspecifiable in terms of the features of those situations (Suthers, 2005; 2006). Furthermore, it is the learners who achieve this accomplishment while attempting to make sense of a situation and of their relationships with each other. Work in scripting to date has not adequately addressed an interactionalist epistemology (Suthers, 2005; 2006) of how learning happens through group interaction (Stahl, 2006) in addition to within individual minds. At the macro level, it makes sense to have an experienced instructor set up situations that generate the productive tensions that drive learning through interaction. At the micro level, guidance is also appropriate, but we should suggest rather than constrain interaction. Over-scripting micro-interactions leaves no place for contingent achievement that comes out of the interaction of the individuals involved.

Pre-authored scripts are a proxy by which an instructional expert can participate in the accomplishment of a successful collaborative learning episode. This participation is a partnership with the participants, who though their own understandings and negotiations can also contribute to the success of the episode. Explicit and participant-editable representations of scripts (e.g., Carell et al., 2005) can serve as a resource in this process if designed in a manner that minimizes the costs of coordinating representations (van Bruggen, Kirschner, & Jochems, 2002). Scripts are not only for the computer and for the educator who would control learners through the computer: they are also a potential resource for the learners themselves.

REFERENCES

Andriessen, J., Baker, M., & Suthers, D. (Eds.) (2003). *Arguing to learn: Confronting Cognitions in Computer-Supported Collaborative Learning Environments*. Kluwer book series on Computer Supported Collaborative Learning, Pierre Dillenbourg (Series Editor). Dordrecht: Kluwer.

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

Carell, A., Herrmann, T., Kienle, A., & Menold, N. (2005). Improving the coordination of collaborative learning with process models. In T. Koschmann, D. Suthers, & T.-W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years! (pp. 18-27).* Mahwah, NJ: Lawrence Erlbaum Associates.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds.), *Perspectives on Socially Shared Cognition.* (pp. 127-149). Hyattsville, MD: American Psychological Association.

Constantino-Gonzales, M. A., Suthers, D., & Escamilla, J. (2003). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education* 13(2-4), 263-299.

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.

Flores, F., Graves, M., Hartfield, B., & Winograd, T. (1988) Computer systems and the design of organizational interaction. *ACM Transactions on Office Information Systems, 6*(2), 153-172.

Hollan, J., & Stornetta, S., (1992). Beyond being there. In *Proceedings of CHI'92* (pp. 119-125). ACM Press.

Ikeda, M., Go, S., & Mizoguchi, R. (1997). Opportunistic group formation – a theory for intelligent support in collaborative learning. In *Proceedings of AIED-97* (pp. 167-174). Kobe, Japan.

Jermann, P., & Dillenbourg, P. (2003). Elaborating new arguments through a CSCL script. In J. Andriessen, M. Baker, & D. Suthers (Eds.), *Arguing to learn: Confronting Cognitions in Computer-Supported Collaborative Learning Environments. Kluwer book series on Computer Supported Collaborative Learning, Pierre Dillenbourg* (Series Editor). (pp. 205-226). Dordrecht: Kluwer.

Koschmann, T., Zemel, A., Conlee-Stevens, M. Young, N., Robbs, J., & Barnhart, A. (2005). How do people learn? In R. Bromme, F. Hesse, & H. Spada (Eds.), *Barriers and Biases in Computer-mediated Knowledge Communication (and how they may be overcome)* (pp. 265-294). Amsterdam: Kluwer Academic Press.

Murray, T. (1999). Authoring intelligent tutoring systems: Analysis of the state of the art. *International. Journal of Artificial Intelligence and Education, 10*(1), 98-129.

Robertson, J., Good, J., & Pain, H. (1998). BetterBlether: The design and evaluation of discussion tool for education. *International Journal of Artificial Intelligence in Education, 9,* 219-236.

Slavin, R. E. (1995). *Cooperative Learning: Theory, Research and Practice. (2nd ed.)* Boston: Allyn & Bacon.

Stahl, G. (2006). *Group Cognition: Computer Support for Collaborative Knowledge Building.* Cambridge, MA: MIT Press.

Steiner, I. D. (1972). *Group Process and Productivity.* New York: Academic Press.

Suthers, D. (2001). Collaborative representations: Supporting face-to-face and online knowledge-building discourse. *Proceedings of the 34th Hawai'i International Conference on the System Sciences (HICSS-34)[CD-ROM].* Maui, Hawai'i: Institute of Electrical and Electronics Engineers, Inc. (IEEE).

Suthers, D. (2005). Technology affordances for intersubjective learning: A thematic agenda for CSCL. In T. Koschmann, D. Suthers, & T.-W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years! (pp. 662-671).* Mahwah, NJ: Lawrence Erlbaum Associates.

Suthers, D. (2006). Technology affordances for intersubjective meaning-making: A research agenda for CSCL. *International Journal of Computers Supported Collaborative Learning, 1*(3).

Sweller, J., van Merriënboer, J. J. G., & Paas, F. G. W. C. (1998). Cognitive architecture and instructional design. *Educational Psychology Review, 10*(3), 251-296.

Toth, E., Suthers, D., & Lesgold, A. (2002). Mapping to know: The effects of evidence maps and reflective assessment on scientific inquiry skills. *Science Education, 86*(2), 264-286.

Van Bruggen, J., Kirschner, P., & Jochems, W. (2002). External representation of argumentation in CSCL and the management of cognitive load. Learning and Instruction, 12(1), 121-138.

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes.* Cambridge, MA: Harvard University Press.

Weinberger, A., Reiserer, M., Ertl, B., Fischer, F., & Mandl, H. (2005). Facilitating computer-supported collaborative learning with cooperation scripts. In R. Bromme, F. W. Hesse, &

H. Spada (Eds.), *Barriers and Biases in Computer-mediated Knowledge Communication (and how they may be overcome)* (pp. 15-37). New York: Springer.

Whitworth, B., Gallupe, B., McQueen, R. (2000). A cognitive three-process model of computer-mediated group interaction. *Group Decision and Negotiation, 9,* 431-456.