Computer-Supported Collaborative Learning
DOI 10.1007/s11412-007-9014-4

Specifying computer-supported collaboration scripts

Lars Kobbe · Armin Weinberger · Pierre Dillenbourg · Andreas Harrer · Raija Hämäläinen · Päivi Häkkinen · Frank Fischer

Received: 4 July 2006 / Accepted: 10 July 2007

Abstract Collaboration scripts facilitate 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. A standardization of collaboration scripts first requires a specification of collaboration scripts that integrates multiple perspectives from computer science, education and psychology. So far, only few and limited attempts at such specifications have been made. This paper aims to consolidate and expand these approaches in light of recent findings and to propose a generic framework for the specification of collaboration scripts. The framework enables a description of collaboration scripts using a small number of components (participants, activities, roles, resources and groups) and mechanisms (task distribution, group formation and sequencing).

Keywords Activities · Collaborative learning · Collaboration scripts · CSCL scripts · Group formation · Roles · Sequencing · Task distribution

L. Kobbe
Q1 Knowledge Media Research Center, Tübingen, Germany

A. Weinberger
Q1 University of Munich, Munich, Germany

P. Dillenbourg
Q1 Swiss Federal Institute of Technology, Lausanne, Switzerland

A. Harrer
Q1 University of Duisburg-Essen, Duisburg, Germany

R. Hämäläinen · P. Häkkinen
Q1 University of Jyväskylä, Jyväskylä, Finland

F. Fischer
Q1 University of Munich, Munich, Germany

L. Kobbe (✉)
Q1 Institut für Wissensmedien, Konrad-Adenauer-Str. 40, 72072 Tübingen, Germany
e-mail: l.kobbe@iwm-kmrc.de

Introduction

Successful collaborative learning depends upon effective interaction amongst learners. However, when learners are left to their own devices, they rarely engage in productive interactions such as asking each other questions, explaining and justifying their opinions, articulating their reasoning, or elaborating and reflecting upon their knowledge (Barron 2003). Collaboration scripts aim to foster collaborative learning by shaping the way in which learners interact with one another. In specifying a sequence of learning activities, together with appropriate roles for the learners, collaboration scripts are designed to trigger engagement in social and cognitive activities that would otherwise occur rarely or not at all.

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 (Fischer et al. 2007; MOSIL 2004). Computer-supported collaboration scripts carry the additional benefit of reducing the coordinative effort both on the teachers' and the students' part. Computers may, for instance, keep track of the students' position within the script sequence, alert or prompt students to engage in specific activities, and provide additional information and resources when needed. Furthermore, once programmed, computer-supported scripts can be reused as many times and with as many students as desired. However, computer-supported scripts are generally so intrinsically interwoven with their particular environment that they suffer from the problem of being neither portable to another learning environment nor reusable in another context. While this may seem to be an issue of merely practical relevance, it points at a more fundamental problem: How can we define a script independent from its particular implementation in a computer-supported learning environment? Without an answer to this question, systematic research on collaboration scripts is difficult because we cannot differentiate between effects from the script and effects from its particular computational implementation. For this, a specification of computer-supported collaboration scripts is needed, addressing concerns from both research and practice.

In this paper, we propose a framework for the specification of collaboration scripts resulting from collaborative research by educational scientists, cognitive psychologists and computer scientists. By this means, we hope to have arrived at a framework that is not biased by any one particular theoretical perspective. We have also tried to take into account a wide range of scripts (following the macro- & micro-script distinction by Dillenbourg and Jermann 2006), some of which will be used as examples in this paper for illustration of the framework and as models of good practice for educators. Script designers may find this paper useful in that we provide a detailed description of script components and mechanisms that can be used as a reference. As the framework serves as a conceptual basis for the computational formalization of computer-supported collaborative scripts, it should also appeal to computer scientists interested in educational modelling languages.

Fostering specific collaborative learning activities through scripts

Collaboration scripts are based on the *scripted cooperation* approach (O'Donnell and Dansereau 1992) which differs from other collaborative learning approaches chiefly in the fact that it focuses on the specific activities that learners are expected to engage in, whereas others leave them unspecified or vague (O'Donnell 1999). In targeting those activities that have emerged from research findings in cognitive and educational psychology as being strongly related to learning (e.g., Cohen 1994; King 2006; Webb and Palincsar 1996), scripts are assumed to lead to higher level cognitive processing and therefore to better learning outcomes. Some of the activities most recurrently referred to are elaboration, explanation, argumentation and question asking.

Elaboration is characterized by relating new ideas and concepts to that which is already known, making it personally more meaningful and expanding it in multiple ways, such as adding details, giving examples, making analogies, creating visualizations and predicting outcomes. The benefits of elaboration to learning are well documented (e.g., van Boxtel et al. 2000). High quality elaboration involves a reorganization of the learner's knowledge and can, for instance, be promoted by contrasting concepts rather than by simply creating personal examples or expanding on the effects of concepts (Hamilton 1997).

Good explanations go beyond simply giving a step-by-step solution insofar as they also articulate the explainer's reasoning and elaborate on the concept (Bargh and Schul 1980). Giving explanations fosters learning in that it prompts learners to check for inconsistencies and knowledge gaps as well as to reorganize and clarify the material in order to meet the target audience's knowledge level (e.g., Webb 1989). In order to benefit from explanations that they receive, learners should try to apply the help given until they reach the point where they are able to solve a problem on their own (Webb et al. 1995). Q3

Argumentation involves the generation of claims or assertions and their justification with evidence. Although arguments are primarily intended to convince someone else of one's own position, they are also effective in promoting self-explanations and deeper understanding of a subject (Baker 2003). In constructing arguments, learners must actively search for knowledge with which they can support their claim. In order to arrive at strong arguments that cannot easily be refuted, learners need to reflect upon their own understanding and check for possible ambiguity and inconsistencies in their statements. Furthermore, collaborative learners construct and exchange arguments, counter-arguments and replies in discourse, in which learners challenge and refine their knowledge (Leitão Q3 2000).

Question asking can foster comprehension by elaborating on content and, furthermore, help to monitor comprehension by checking for errors in understanding (Palincsar and Brown 1994; King 1994). Q3 It can also be used for eliciting information that is needed in order to fill in gaps in knowledge (as in 'help seeking'). Question asking is frequently assigned as an activity in collaborative learning. However, when learners are simply asked to engage in question asking, they typically ask rather simple, factual questions that require little cognitive effort. Research has identified a number of strategies (e.g., providing generic example questions or making use of question starters) to help learners generate more effective, thought-provoking questions (see Rosenshine et al. 1996, Q3 for an overview).

To conclude, the role of activities in collaborative learning is of central importance, since the type of activities induced has a strong impact on the cognitive processes evoked. Nevertheless, it is not simply what the learners do, but also how they do it, that is predictive of the learning outcomes. While all collaboration scripts aim at promoting these desired activities in the learners' interaction, there appear to be general differences that lead researchers to distinguish macro- from micro-scripts (e.g., Dillenbourg and Jermann 2006). Macro-scripts typically differ from micro-scripts in the level of granularity (i.e., activities typically describe longer time segments compared to micro-scripts). Micro-scripts, on the other hand, tend to provide a higher degree of scaffolding (additional instructional support for facilitating tasks that exceed the learner's current level of competence) with sentence starters, question prompts or detailed descriptions that may gradually be faded out as learners become more competent (Kollar et al. 2006; Pea 2004, on scaffolding and fading). Generally speaking, macro-scripts take a rather top–down and pedagogical approach to structuring interaction (emphasizing the orchestration of activities within the classroom and thus supporting the teacher in implementing collaborative learning scenarios), whereas micro-scripts take a more bottom–up and psychological approach (emphasizing the activities of individual learners and thus facilitating learners to engage in specific socio-cognitive processes when learning together).

Approaches to the specification of collaboration scripts

Q2

Some attempts at specifying collaboration scripts have been made and contributed to the development of our framework (namely those by Dillenbourg 2002; Dillenbourg and Jermann 2006; Kollar et al. 2006), each of which we will briefly describe:

In a pioneering attempt to analyze collaboration scripts, Dillenbourg (2002) identified a number of aspects that served as a preliminary framework for script comparison and design. Dillenbourg (2002) described scripts as a sequence of phases, each characterized by the following five attributes: *type of task* to be accomplished, *group formation* (and composition), *distribution of the task* within and among groups, *type and mode of interaction* (e.g., co-located vs. remote, synchronous vs. asynchronous, text-based vs. voice-based, etc.), and *timing* of the phase. From phase to phase, each of these attributes can change. The allocation and re-allocation of roles and activities, as well as physical or virtual resources, are considered to be part of the task distribution.

In a recent approach, Dillenbourg and Jermann (2006) expanded the scope of collaboration scripts to encompass more than just small group interaction. While collaborative activities are still regarded as being essential to the learning process and constitute the core learning mechanism, *integrative scripts* also include individual activities and activities encompassing a whole class. Thus, collaborative *core scripts* are positioned within a *didactic envelope*; that is, pre- and post-structuring activities that enable scripts to be optimally integrated into the lesson plan (e.g., introducing the topic, reflecting on what was discussed, etc.) and which contribute to their effectiveness and consistency. The integrative aspect of scripts becomes even more evident in the case of computer-supported collaboration scripts, in which scripts integrate virtual with physical activities and manage the data flow between them. For example, in the ArgueGraph Script (Table 1), discussion across a whole class is based on a graph that was computed from individual responses priorly collected in the form of a questionnaire. According to Dillenbourg and Jermann (2006), the didactic envelope features two salient dimensions: The *time structure* and the *social structure* of scripts. As for the time structure, each activity within a script can be mapped onto a certain phase or point of time in the overall sequence and is also limited in its duration. Regarding the social structure, activities can be mapped onto specific social planes (individual, group, class, community or world) on which an activity is to be carried out (i.e., activities on the class plane will involve all students in the class). The core script within the didactic envelope typically involves a *task distribution* among all group members. Dillenbourg and Jermann (2006) viewed interaction as the means of overcoming these task divisions and proposed a model, according to which the nature of the task distribution also determines the nature of the members' interaction (e.g., explanation, argumentation, mutual regulation, etc.). Following Schwartz' (1995) definition of

Table 1 Summary of collaboration script examples

Name (Source)	Summary
MURDER Script (Dansereau et al. 1979)	MURDER is short for Mood, Understanding, Recall, Detection, Elaboration, and Review. Students learn in pairs from a textbook, one being the summarizer and the other the listener. After setting the mood for studying, both read a text passage for understanding. The Summarizer recalls what has been read while the listener detects errors or omissions and gives feedback. Then both elaborate on the read passage and repeat everything with switched roles for the next passage of text. Finally, both review the read passages and reflect on what they have learned.
Universanté Script (Dillenbourg and Jermann 2006)	Students from different nations solve problem cases in mixed and changing teams. Each case is first read and discussed in teams of mixed nations. National teams then inform each other about the cases read and create a national fact sheet. These national fact sheets are then compared and compiled by teams of students with thematically similar cases (mixed nations). These same teams present their compiled fact sheets to other teams of their same nationality and receive feedback. Finally, students return to their initial case group and work out a case solution.
ArgueGraph Script (Dillenbourg and Jermann 2006)	Students first individually argue for or against items on a questionnaire. Their opinion is plotted onto a two-dimensional graph. Students with highly conflicting opinions (point distance in the graph) are grouped together in pairs and receive another copy of the questionnaire to fill out. Students discuss what arguments to write for each item. The teacher collects the questionnaires and helps each small group in turn to elaborate on and revise their arguments. The teacher then groups all arguments by item. Finally, each student is assigned one item for which to write a synthesis of all arguments.
Social Script (Weinberger et al. 2005)	Three cases studies are analyzed and reviewed by groups of three students in parallel. Each student writes a case analysis, then critiques the other two written case analyses and finally revises his/her own case analysis based on the critiques received by the other students. Both roles of case analyst and constructive critic are additionally supported with text prompts that learners are supposed to act out, such as “These aspects are not clear to me yet”; “We have not reached consensus concerning these aspects”; and “My proposal for an adjustment of the analysis is” for the critic’s role.

collaborative learning as the effort necessary to build a shared understanding, learning results from the interactions that students engage in to build a shared understanding despite the fact that the task is distributed (Dillenbourg and Jermann 2006). According to the design principle ‘Split Where Interaction Should Happen’, the model was named SWISH.

In another recent approach, Kollar et al. (2006) compared scripts from two research traditions: Computer-supported collaborative learning (CSCL) and instructional psychology. Five conceptual components were identified according to which scripts could be differentiated: *Objectives*, *activities*, *sequencing*, *roles*, and *types of representation*. Scripts from both research traditions had similar objectives regarding the group task. However, as all of the CSCL scripts made use of computer-mediated communication, a major part of these scripts was devoted to the support of smooth and coherent communication and coordination with respect to the inherent weaknesses of the medium (e.g., Herring 1999). Although scripts did not differ much in the kind of activities that they promoted (e.g., arguing, explaining, question asking), scripts from instructional psychology gave much more support when it came to how to carry out these activities (through training or detailed instruction). CSCL scripts typically lacked a prescribed sequential structuring of activities, although the interface sometimes implicitly induced a certain sequence of activities. Also, CSCL scripts often lacked clear role distribution in contrast to scripts from instructional psychology. Scripts from both traditions were rather inconsistent in the way which they were represented during the learning phase (e.g., represented internally in the learners' mind or represented externally by buttons on the screen, prompt cards, etc.). These findings highlight the diversity of scripts, both in realization as well as in function and purpose.

Specifying collaboration scripts—a framework Q2

Based on the works of Dillenbourg (2002), Dillenbourg and Jermann (2006) and Kollar et al. (2006), we aim to propose a framework that consolidates recent approaches with current ideas and conceptualizations of our research. The framework will be illustrated with four collaboration scripts that cover a wide range of possible script designs (summarized in Table 1).

Our framework distinguishes between different levels of abstraction for the specification of scripts. At the highest level of abstraction are the so called script schemata (Dillenbourg and Jermann 2006). Schemata describe the core design principle through which the script is expected to trigger specific interactions. For instance, many scripts use a variation of the Jigsaw method (Aronson et al. 1978) and form pairs with complementary information. Other scripts pair learners with contradictory knowledge or opinions in order to create socio-cognitive conflict, or assign and alternate roles that foster reciprocal activities, such as questioning or tutoring. Accordingly, these core design principles are referred to as the Jigsaw schema, the conflict schema or the reciprocal schema (Dillenbourg and Jermann 2006). Besides these schemata, it is expected that further schemata exist, which we aim to identify in the future. However, for the purpose of specification and subsequent formalization, we target a level of abstraction that is below that of script schemata and can still be generalized to allow for adaptation to the learning context, for example, the number of participants and the kind and content of resources. Dillenbourg and Jermann (2006) refer to this level as the level of script classes, which cover a range of scripts representing acceptable variations of a prototype. For example, the Universanté Script is a prototype of the Jigsaw schema, whereas the ArgueGraph Script is a prototype of the conflict schema (see Table 1). This level of abstraction also allows us to describe the structural composition of both macro- and micro-scripts.

Regarding the structural composition of scripts, we distinguish between mechanisms on the one hand and components on the other. This distinction has not been made in earlier approaches. For example, Kollar et al. (2006) regarded *role distribution* as a script component, whereas we refer to roles as a component and distribution as a mechanism. Furthermore, we do not regard earlier components such as *type of representation* or *mode of interaction* as either mechanism or component, because these aspects are merely design decisions that can be used to build variations of one and the same script class. The script components and mechanisms will be presented separately in the following sections and will be illustrated with script examples in Tables 2 to 5.

Table 2 Components of collaboration script examples

Name	MURDER Script	Universanté Script	ArgueGraph Script	Social Script
Participants	An even number of participants	Participants from at least two nations with at least as many participants per nation as there are case descriptions	An even number of at least four participants (works best with 20–30 participants) and a tutor	An amount of participants that must be divisible by three

Activities[a]	a) relaxing, focusing;
	b) reading, monitoring comprehension;
	c) summarizing, explaining;
	d) monitoring, giving feedback;
	e) elaborating;
	f) reviewing, reflecting
		a) analyzing and elaborating the case;
		b) summarizing and explaining;
		c) analyzing, comparing and relating new information to personal prior knowledge;
		d) giving feedback and critiquing;
		e) problem solving
			a) justifying opinions and constructing arguments;
			b) comparing, evaluating, and elaborating;
			c) negotiating and constructing arguments;
			d) explaining and justifying opinions;
			e) summarizing and making connections
				a) applying theoretical concepts to cases and constructing arguments;
				b) critiquing (initially scaffolded with prompts for eliciting clarification, identifying conflicting views and constructing counter-arguments)

Roles	A summarizer and a listener	None	None	An analyst and two critics
Resources	Learning material with a small number of text passages	Case descriptions from at least two themes, with at least two case descriptions per theme.	One questionnaire for each participant and another copy for each small group. One argument sheet per questionnaire item.	Three case descriptions
Groups	Pairs	Theme groups, case groups and national groups	Class group and pairs	Case groups

[a] Letters are cross-references to activities listed in Table 5

Script Components Q2

Our framework aims to be economic in enabling scripts to be described with just the small number of components shown in Table 2: the individuals that participate in a script, the activities that they engage in, the roles they assume, the resources that they make use of, and the groups they form.

Participants Q2

The term *participants* is used synonymously with users, persons, or people; in other words, as a general abstraction of concrete individuals. Scripts usually have certain requirements regarding the total number of participants that they can handle, sometimes given as a variable range (e.g., from three to eight participants) or as a multiple of another script component (e.g., two participants per text book). Scripts may also take into account specific participant characteristics, such as individual opinion or knowledge in a domain. For

Table 3 Task distribution mechanism in collaboration script examples

Name	Task distribution
MURDER Script	Each pair receives the learning material. Within each pair, one participant assumes the role of summarizer and the other the role of listener.
Universanté Script	Each case group receives one case description.
ArgueGraph Script	In the survey phase, all participants receive a copy of the questionnaire. In the conflict and elaboration phase, each small group receives another copy of the questionnaire. In the reflection phase, each participant receives one argument sheet.
Social Script	Each case group receives one case description. Roles are distributed in such a way that each participant assumes the role of analyst in one case group and the role of critic in all other case groups.

example, the ArgueGraph Script makes use of participants with different opinions, whereas the Universanté Script utilizes participants with different background knowledge.

Activities Q2

The kinds of activities specified are highly relevant to the type and degree of learning taking place (King 2006). Activities form a hierarchical structure in which any greater activity can be decomposed into lesser, more fine-grained activities, and any lesser activity can be subsumed by greater, more coarse-grained activities. For instance, discussing can be decomposed into explaining, constructing counter-arguments, question asking, etc., and asking somebody to check a report for mistakes can be generalized as help-seeking. Moreover, activities are provided with different degrees of scaffolding. When learners are assumed to be proficient with a specified activity, a much lower degree of scaffolding (down to none) is needed than with activities they are not proficient with. In the Social Script, for example, the activity of critiquing is initially scaffolded with prompts (sentence starters) for eliciting clarification, identifying conflicting views and constructing counter-arguments, because learners typically lack the knowledge of how to effectively engage in critiquing (see Table 2). These prompts may be faded out when the learners become more competent. Finally, it needs to be kept in mind that activities can be described from

Table 4 Group formation mechanism in collaboration script examples

Name	Group formation
MURDER Script	All participants are grouped into pairs.
Universanté Script	For each case description, one case group is formed, composed of at least one participant per nation, balanced amongst the groups. For each theme, one theme group is formed, composed of all case groups related to the theme.
ArgueGraph Script	In the survey phase, all participants together form the class group. In the conflict and elaboration phase, participants form pairs composed of participants with maximal difference in their questionnaire responses.
Social Script	All participants are grouped by the number of case descriptions. For each case description, one case group is formed that consists of all participants. Thus, each participant becomes member of all case groups although with different roles in each: Each participant is the responsible analyst for one case and critic for all other cases. The number of participants equals the number of case descriptions.

Table 5 Sequencing mechanism in collaboration script examples

Name	Sequencing[a]
MURDER Script	Within each pair, both participants set the *M*ood for studying. (a)
	Within each pair and for each passage of text:
	... both participants read the text for *U*nderstanding. (b)
	... the summarizer *R*ecalls what has been read. (c)
	... the listener *D*etects errors/omissions and gives feedback. (d)
	... both participants *E*laborate on the read material. (e)
	... after each cycle, participants rotate roles.
	Within each pair, both participants *R*eview the learning material and reflect on what they have learned. (f)
Universanté Script	Within each case group, all participants read and discuss the case. (a)
	Within each national group, the members of each theme group in turn present a synthesis of their case experience. (b)
	Within each theme group, the members of each national group create a fact sheet concerning the theme's status within their nation. (b)
	Within each theme group, all participants discuss the similarities and differences between the fact sheets of different nations. (c)
	Within each nation group, and for each theme group in turn: the members of the theme group present their fact sheet. (d)
	everybody else provides comments on the presented fact sheets. (e)
	the members of the theme group modify their fact sheet according to the comments. (b)
	Within each case group, all participants propose a solution for the case problem. (f)
ArgueGraph Script	Survey phase:
	Within the class group, all participants individually fill out the first copy of the questionnaire. (a)
	The tutor displays the aggregated results of the questionnaire (the participants' choices are plotted as points in the graph) to the participants.
	Within the class group, all participants jointly discuss the displayed results of the questionnaire. (b)
	Conflict phase:
	Small groups are formed based on each participant's responses to the questionnaire
	Within each small group, all participants jointly fill out the second copy of the questionnaire, i.e., they negotiate on a single choice and generate a shared argument for this choice. (c)
	Elaboration phase:
	The tutor collects all questionnaires
	For each small group in turn, the tutor asks participants to comment on their arguments and gives advice on how to relate their arguments to theories and concepts. (d)
	For each item of the questionnaire, the tutor compiles all choices and arguments from each small group on an argument sheet
	Reflection phase:
	The tutor distributes one argument sheet to each participant
	Within the class group, each participant individually writes a synthesis of all arguments on the argument sheet. (e)
Social Script	Within each case group:
	the analyst writes a case analysis. (a)
	...wait for all case group analysts to be done each critic in turn writes a first critique of the case analysis. (b)

Table 5 (continued)

Name	Sequencing[a]
	...wait for all case group critics to be done the analyst considers each critique and writes a reply to each. (a)
	...wait for all case group analysts to be done each critic in turn reads the reply and writes a second critique. (b)
	...wait for all case group critics to be done... the analyst considers all critiques and revises the case analysis. (a)
	Note: The roles of each participant rotate twice over the course of the script. Each participant begins with the role of analyst and changes his/her role to the first critic, second critic, etc. This cycle is repeated twice and ends with each participant resuming the role of analyst.

[a] Letters are cross-references to activities listed in Table 2

different angles, such as ‘writing a case analysis’ compared to ‘applying theoretical concepts to cases.’ While the former wording reflects the mode and outcome of an activity (which is especially useful from a computational point of view), the latter is more helpful in terms of determining the cognitive processes underlying the activity. This is particularly evident in the case of the ArgueGraph Script where the activity of ‘filling out a questionnaire’ could mean anything from reporting personal habits to solving text problems, but in this case implies the learners’ engagement in cognitive activities such as constructing arguments and justifying opinions. Therefore, script examples in Table 2 and Table 5 provide a description of activities from both angles (cross-referenced with letters).

Roles Q2

A simple function of roles in collaboration scripts is to refer to specific participants when assigning activities or allocating resources. However, roles also provide a kind of legitimacy and are associated with privileges, obligations and expectancies. For example, a person in the role of the critic has the legitimacy to criticize another student’s work, the privilege to speak out frankly, the obligation to justify the critique, and is expected to point out shortcomings as well as ways of improvement. As roles are closely tied to learning activities, their title is usually predictive of the activities that participants are allowed, obliged or expected to engage in (e.g., the role of “summarizer” and “listener” in the MURDER Script). Thus, specific roles may foster particular learning activities, such as a ‘scientist’ promoting planning, observing, and drawing conclusions. Participants may assume one or more roles at any given time and can exchange these roles with other participants. Even if no explicit role assignment takes place, collaboration scripts may implicitly assume a specific ‘non-role condition’ in order to prevent counterproductive self-appointment of roles. For instance, it is often not desirable for small groups to allow one participant assume the self-appointed role of group leader.

Resources Q2

Resources in general comprise virtual or physical objects that can be allocated to script participants (e.g., text books, links to websites, etc.). Some resources may be predefined objects offering important information (information sources like websites, articles, etc.) or functionality (tools like calculators, online dictionaries, etc.) whereas others may be created or significantly modified during the script (editable products, such as fill-out questionnaires and written notes). Within this framework, resources are not restricted to tangible objects, but may in effect encompass everything the learning activities operate on, such as verbal and non-verbal productions. Although resources have not been identified as script components in earlier approaches, they are important within the context of scripts because they often constitute a common object of or reason for interaction. Resources often foster interaction in that they are distributed amongst learners in a way that induces social interdependence (i.e., each learner needs access to the others' resources), a method commonly employed in collaborative learning scenarios such as Jigsaw (Aronson et al. 1978; see the Universanté Script, which is based on the Jigsaw method) and conceptualized in the SWISH-model by Dillenbourg and Jermann (2006).

Groups Q2

Groups generally form a hierarchical structure with larger groups being composed of one or more smaller groups. Participants can be grouped in terms of existing common features such as gender, age or national groups (ala the Universanté Script) or can be distributed into new groups based on certain criteria such as desired group size, amount, or composition (see the group formation mechanism). Participants can also become members of multiple and overlapping groups. In the Social Script, for instance, each participant is a member of all case groups, but with different roles in each group.

Script Mechanisms Q2

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*). Each of these mechanisms features particular principles that are important for issues of scalability (e.g., for applying the same script to a varying number of participants) and later formalization.

Task Distribution Q2

A key feature of scripts is the distribution of the task, such as providing participants with only one piece of the information that they need in order to foster knowledge exchange with one another. Decomposable activities can also be distributed in such a way that one learner engages in a cognitive activity while the other learner engages in a supportive metacognitive activity. Distribution typically makes use of a function; in other words, the elements of one set (e.g., roles) are mapped through a specific function (e.g., one-to-one) onto the elements of another set (e.g., participants). In some cases, this function may become quite complex, such as arranging roles, participants and groups in such a way that each participant assumes a different role in each group (see the Universanté Script in Table 3).

Group Formation Q2

While some groups just happen to exist by definition (e.g., gender groups such as men and women or hair color groups such as blondes, brunettes and redheads), other groups must be formed by a particular procedure or principle. In most cases, group formation is very simple, such as forming groups by amount (e.g., dividing a class into four groups), by size (e.g., dividing a class into groups of four) or by a combination thereof (e.g., forming four groups of four). Some scripts make use of more complicated principles that do not only take into account group numbers and size, but also the composition of each group and the overall balance among the groups (see Table 4).

Sequencing

Q2

Collaboration scripts provide a temporal structure for interaction; that is, they specify the order in which events and activities are to take place. The sequence is typically designed according to a theoretical model (e.g., the inquiry cycle model by White and Frederiksen 1998). Script sequences are often divided into distinct segments, referred to as *phases* (see the ArgueGraph Script in Table 5), that provide a helpful point of reference for students and teachers and are frequently used for managing time. Scripts are not limited to a strictly linear sequence of activities, but may also feature complex sequential arrangements with loops and branches. Many collaboration scripts make use of repetitions with minor variations: In the case of the MURDER Script, for instance, the text passage and role allocations change with each cycle. Further common sequencing patterns are traversion, rotation and fading. Traversion describes a sequence in which all elements of one set are looped through, with only one element being in use at any given time (the text passage in the MURDER Script). This is often used to give students the chance to practice the same activities on different sets of data. Rotation permutates the order of elements in a given set (the roles in the MURDER Script) and is frequently used to give each student the opportunity to engage in each activity. Fading refers to features that are gradually added (faded in) or removed (faded out) from a script and is often used to gradually increase or decrease the degree of scaffolding in activities.

Conclusions

Beginning with a review of the original conception of scripts for collaborative learning, we presented recent approaches toward a specification of collaboration scripts. We then presented and elaborated on a number of essential components and mechanisms that constitute a script, namely participants, activities, roles, resources and groups as components and task distribution, group formation and sequencing as mechanisms.

The proposed framework serves a variety of different purposes. First of all, it provides a common terminology as the cornerstone for knowledge exchange and accumulation in scientific research on scripting collaborative learning. It is aimed at facilitating the systematic exploration of script mechanisms as well as the comparison and integration of research results. Nevertheless, the framework does not impose a specific theoretical perspective on the interplay of script components and mechanisms, but is rather intended to be useful within a wide range of theories. For practitioners, the components and mechanisms elaborated here may serve as a checklist for the design of scripted CSCL environments, as they are of high relevance to collaborative learning. We also expect the prototypical examples to be useful as models of good practice.

In the near future, we aim to go beyond the semi-formal description demonstrated in the script examples to a formal modelling language that can be used in computer-supported learning environments. Because the linear structure of text-only representations is ill-suited to model concepts such as rotation, fading, or traversion, at least part of the formal language will use a graphical notation. The use of UML activity or state diagrams for the modelling of collaboration scripts has been discussed recently (Haake and Pfister 2007; Harrer and Malzahn 2006) and can be used as a reference point for graphical notations. We plan to design a visual language that relies on the formal semantics of these notations, yet uses graphical elements for the concepts of the framework we presented here.

Nevertheless, we believe that a semi-formal script description is more than just a transitional step in the formalization process and is in itself valuable: Many researchers and practitioners (even in the field of technology-enhanced learning) may not feel at ease with reading formal, computational languages, let alone applying this formalism in transcribing scripts that they have designed themselves. Thus, a semi-formal description may serve as an intermediate representation that bridges the gap between differing demands.

Acknowledgments This research was conducted by the European Research Team on 'Computer-Supported Scripting of Interaction in Collaborative Learning Environments' (ERT CoSSICLE) and was funded by the Network of Excellence 'Kaleidoscope.'

References

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

Baker, M. (2003). Computer-mediated argumentative interactions for the co-elaboration of scientific notions. In: J. Andriessen, M. Baker, & D. Suthers (eds.), *Arguing to learn: confronting cognitions in computer-supported collaborative learning environments.* (Vol. 1) (pp. 1–25). Dordrecht: Kluwer.

Bandura, A. (1997). *Social learning theory.* Englewood Cliffs, NJ: Prentice-Hall. Q5

Bargh, J. A., & Schul, Y. (1980). On the cognitive benefits of teaching. *Journal of Educational Psychology, 72*, 593–604.

Barron, B. (2003). When smart groups fail. *Journal of the Learning Sciences, 12*(3), 307–359.

Chi, M. T. H., deLeeuw, N., Chiu, M., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*, 439–477. Q5

Cohen, E. G. (1994). Restructuring the classroom: Conditions for productive small groups. *Review of Educational Research, 64*(1), 1–35.

Dansereau, D. F., Collins, K. W., McDonald, B. A., Holley, C. D., Garland, J. C., Diekhoff, G., et al. (1979). Development and evaluation of a learning strategy program. *Journal of Educational Psychology, 71*, 64–73.

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. (2006). Designing integrative scripts. In: F. Fischer, I. Kollar, H. Mandl, & J. Haake (Eds.), *Scripting computer-supported collaborative learning: Cognitive, computational and educational perspectives.* New York: Springer.

Dillenbourg, P., & Tchounikine, P. (2007). Flexibility in macro-scripts for CSCL. *Journal of Computer Assisted Learning, 23*, 1–13. Q5

Doise, W., & Mugny, G. (1984). *The social development of the intellect.* Oxford: Pergamon Press. Q5

Fischer, F., Kollar, I., Mandl, H., & Haake, J. (eds.). (2007). *Scripting computer-supported collaborative learning: Cognitive, computational and educational perspectives*. New York: Springer.

Haake, J. & Pfister, H. R. (2007). Flexible scripting in net-based learning groups. In: F. Fischer, I. Kollar, H. Mandl, & J. Haake (eds.), *Scripting computer-supported collaborative learning: Cognitive, computational and educational perspectives.* New York: Springer.

Hamilton, R. J. (1997). Effects of three types of elaboration on learning concepts from text. *Contemporary Educational Psychology, 22*(3), 299–318.

Harrer, A., & Malzahn, N. (2006). Bridging the gap—towards a graphical modelling language for learning designs and collaboration scripts of various granularities. In: Kinshuk, R. Koper, P. Kommers, P.

Kirschner, D. Sampson, & W. Didderen (eds.), *Advanced learning technologies ICALT 2006* (pp. 296–300). Los Alamitos, CA: IEEE Computer Society.

Herring, S. C. (1999). Interactional coherence in CMC. *Journal of Computer-Mediated Communication, 4*(4).

King, A. (1989). Effects of self-questioning training on college students' comprehension of lectures. *Contemporary Educational Psychology, 14*, 366–381. Q5

King, A. (1994). Guiding knowledge construction in the classroom: Effects of teaching children how to question and how to explain. *American Educational Research Journal, 30*, 338–368.

King, A. (2006). Scripting collaborative learning processes: A cognitive perspective. In: F. Fischer, I. Kollar, H. Mandl, & J. Haake (eds.), *Scripting computer-supported collaborative learning: Cognitive, computational and educational perspectives.* New York: Springer.

Kollar, I., Fischer, F., & Hesse, F. (2006). Computer-supported cooperation scripts—A conceptual analysis. *Educational Psychology Review, 18*(2), 159–185.

Kuhn, D. (1991). *The skills of argument.* Cambridge: Cambridge University Press. Q5

MOSIL (2004). *Framework for integrated learning*. Retrieved January 13th, 2006, from http://www.iwm-kmrc.de/cossicle/resources/D23-05-01-F.pdf

O'Donnell, A. M. (1999). Structuring dyadic interaction through scripted cooperation. In: A. M. O'Donnell, & A. King (eds.), *Cognitive perspectives on peer learning.* (pp. 179–196). Mahwah, NJ: Lawrence Erlbaum Associates.

O'Donnell, A. M., & Dansereau, D. F. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance. In: R. Hertz-Lazarowitz, & N. Miller (eds.), *Interaction in cooperative groups: The theoretical anatomy of group learning.* (pp. 120–141). New York: Cambridge University Press.

Pea, R. D. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *Journal of the Learning Sciences, 13*(3), 423–451.

Schwartz, D. L. (1995). The emergence of abstract representations in dyad problem solving. *The Journal of the Learning Sciences, 4*(3), 321–354. Q5

Slavin, R. E. (1983). When does cooperative learning increase student achievement? *Psychological Bulletin, 94*(3), 429–445. Q5

Van Boxtel, C., van der Linden, J., & Kanselaar, G. (2000). Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning and Instruction, 10*, 311–330.

Webb, N. (1989). Peer interaction and learning in small groups. *International Journal of Educational Research, 13*(1), 21–39.

Webb, N. M., & Palincsar, A. S. (1996). Group processes in the classroom. In: D. C. Berliner, & R. C. Calfee (eds.), *Handbook of educational psychology* (pp. 841–873). New York: Macmillan.

Weinberger, A., Ertl, B., Fischer, F., & Mandl, H. (2005). Epistemic and social scripts in computer-supported collaborative learning. *Instructional Science, 33*(1), 1–30.

White, B. Y., & Frederiksen, J. R. (1998). Inquiry, modelling, and metacognition: Making science accessible to all students. *Cognition and Instruction, 16*(1), 3–118.