Chapter 3
How to Realize Flexible Scripting in CSCL

In CSCL, scripts are designed to structure collaborative processes by distributing activities and roles to group learners and by constraining the mode of interaction among peers or between groups (Fischer et. al. 2013). Designing and implementing scripts, however, requires striking the balance between too little script to be helpful and too much control to allow for natural group interactions (Koedinger and Aleven 2007).

Despite the emphasis on the existence of the risks of over-scripting (Beers et al. 2005; Cohen 1994; Dillenbourg 2002; Karakostas and Demetriadis 2009; Kollar et al. 2007), little attention has been given to defining clearly what accounts for over-scripting. Against this background, the question arises here is how to further conceptualize over-scripting and how to realize flexible scripting in CSCL to avoid the risk of over-scripting.

3.1 Over-Scripting

In scripted CSCL, there is the concern of "over-scripting". Dillenbourg (2002) pointed out that the purpose of scripting is to induce constrains that will shape productive collaborative interactions while avoiding the risk of over-scripting. Some empirical studies provided post hoc evidence for the over-scripting assertion: when negative effects (compared to unscripted collaboration) of scripts on learning outcomes were found, they were explained as 'over-scripting' (Beers et al. 2005; Stegmann et al. 2011). For example, a study by Beers et al. (2005) compared a less coercive scripted version of online discussions to a more coercive one when college students tried to solve the high school drop-out problem using a NegotiationTool. The less coercive version presented students on-screen information about every contribution, and asked students whether it needed to be verified or decided upon (agreeing or disagreeing). The more coercive version presented the same information but divided the problem-solving process into two distinct phases:

X. Wang and J. Mu, *Flexible Scripting to Facilitate Knowledge Construction in Computer-supported Collaborative Learning*, Perspectives on Rethinking and Reforming Education, DOI 10.1007/978-981-10-4020-7_3

negotiation of meaning, and negotiation of opinion. Compared to the less coercive script, the coercive one was found to have negative effects on students' contribution and negotiation of common ground. The negative effect was explained as the coercive script caused disruption of collaboration (over-scripting). This is apparently a post hoc explanation of over-scripting. In order to avoid over-scripting in advance, however, we need to know why over-scripting occurs.

3.1.1 Quantitative Over-Scripting

Over-scripting has often been explained cognitively (Dillenbourg 2002). For experienced students in collaborative learning, for instance, their knowledge on collaboration might be chunked and well-structured, if a script segments collaboration into too many sub-processes, it will increase cognitive load and therefore, hinder learning (Dillenbourg 2002). In addition, segmenting collaboration in too many pieces or distributing the resource to too many group members may make it difficult for group members to adopt shared goals, which have often referred to as an important criterion to define good collaboration (Dillenbourg 1999).

Dillenbourg and Tchounikine (2007) revised their definition of over-scripting by the degree of coercion. Degree of coercion describes how many constraints a script imposes on collaborative learning processes. For example, if a script constraints collaboration in a way that makes it sterile by inhibiting the natural interaction mechanisms, over-scripting occurs. In order to operationalize degree of coercion, Dillenbourg and Tchounikine (2007) made a distinction between intrinsic and extrinsic constraints of scripts.

Intrinsic constraints refer to the script design rationale, which is the core mechanism of how a collaboration script would shape peer interaction. For instance, in the Argue Graph script (Jermann and Dillenbourg 2003) which aims at triggering argumentation among pair members, conflict resolution is the core mechanism.

Extrinsic constraints describe different issues such as technological choices, contextual factors or arbitrary decisions. For instance, in the Concept Grid script (Dillenbourg 2002) which distributes partial knowledge to group learners and aims at triggering mutual explanation, researchers' arbitrary fix of group size is extrinsic constraint.

According to Dillenbourg and Tchounikine (2007), intrinsic constraints set up the limits of flexibility. That means, if intrinsic constraints are removed, the script is off the way of its underlying learning principle. For example, the ArgueGraph script requires students with the most different opinions on a given problems to be paired to trigger argumentation. If this constraint is violated and students are freely paired, the design rationale of the script is destroyed.

Extrinsic constraints describe the spaces for flexibility. That is, when there are too many extrinsic constraints imposed by a script, the script runs the risk of over-scripting because extrinsic constraints increase learners' unnecessary load.

3.1.2 *Qualitative Over-Scripting*

Fischer et. al. (2013) argued that over-scripting occurs not only because of the quantity of constraints, but also because of the quality. In their Script Theory of Guidance, internal collaboration scripts are learners' configuration of knowledge about how to act appropriately in collaborative learning while external collaboration scripts are instructional approaches that provide external constraints to collaboration. Both internal and external scripts have four components arranging from a high to a low level. They are *play*, *scene*, *role* and *scriptlet* component. Their theory emphasizes the importance of an external script to be 'optimal' in regards to the script components it provides.

According to the optimal external scripting level principle in the Script Theory of Guidance, over-scripting occurs when the external script targets at a wrong hierarchical level. For example, when a learner poses already a high level component in his/her internal script such as *scene* or *role* component, an external script targeting at a low level such as *scriptlet* would be less efficient because the learner has to process unnecessary information. In contrast, an external script providing supports at a high level would help the learner's dynamic configuration of his/her subordinate internal script components at a low level.

A study by Mäkitalo and colleagues (2005) provides evidence for the optimal scripting principle. In their study, college students participated in CSCL discussions in triads. An external collaboration scripts providing scaffolds at scriptlet level was found to impede learners' knowledge acquisition. Authors argued that these teacher students from educational science have had adequate scriptlet and thus receiving a low level external script caused extra load.

3.1.3 *Motivational Over-Scripting*

Beyond these two explanations aforementioned, there may also be a motivational explanation of why over-scripting occurs. Following Self-Determination Theory (Ryan and Deci 2000), when a CSCL script is too coercive, it may cause motivational problems since it reduces the necessary feeling of autonomy. Negative effects of collaboration scripts on intrinsic motivation are, therefore, to be expected, particularly in the long term and with adult learners (Bruhn 2000; Hron et al. 1997).

A study by Stegmann et al. (2011) supports this assumption. 81 students of Educational Science participated in online argumentative knowledge construction in their study. Degree of an argumentative script was manipulated (low vs. medium vs. high) and negative effects of the medium and the high degree of scripting on students' intrinsic motivation was found.

3.2 Script with Fading

If over-scripting occurs when an external script imposes too many constraints or when it provides scaffolds at a wrong hierarchical level, a straightforward way to avoid over-scripting is by fading out unnecessary external script components. The philosophy behind the instructional design principle of fading is that once students have acquired the desired competencies the external support becomes redundant and should be faded out (Puntambekar and Huebscher 2005). There are studies investigating different algorithms of fading in various learning contexts (Wecker and Fischer 2007; Renkl et al. 2004).

3.2.1 Forward & Backward Fading

Forward fading is a fading procedure from a starting point with little external collaboration scripts. The degree of external collaboration scripts can be increased by prompts, hints or explanations when learners have no sufficient internal scripts to complete the required tasks appropriately or effectively. However, forward fading is seldom used in practice, which leads to a deficiency of empirical evidence if forward fading outperforms a full and continuous script from a starting point.

Backward fading refers to fading from a starting point with full external collaboration scripts, and then scaffolds are removed gradually. From the constructivist perspective, the backward pattern is much more correspondent to the theoretical definition of fading, which emphasize the process of reducing the amount of external instructions.

Forward and backward fading presented here are distinct from the notions when they were first employed by the studies on worked examples. In a series of studies, Renkl and colleagues (Atkinson et al. 2003; Renkl et al. 2002, 2004) consistently demonstrated that detailed worked examples i.e., all solution steps are shown to the learner, were most appropriate when presented to novices, but they should be gradually faded out with increased levels of learner knowledge. Therefore, two different approaches to fading were investigated in the problem-solving studies, namely, backward fading, in which the last solution steps are faded first and forward fading, in which the first solution steps are faded first. It is clear that the forward/backward fading applied in the worked example study was only different in sequence but not in the degree of providing instructional supports.

3.2.2 Horizontal & Vertical Fading

As stated in the Script Theory of Guidance (Fischer et. al. 2013), collaboration scripts indeed embody four important elements based on a hierarchical model:

namely Scriptlet, Role, Scene and Play. One of the basic principles in this theory is that the effective external collaboration script is to direct learners at the highest possible hierarchical level of internal collaboration script components for which subordinate components are already available to the learner. From this point of view, collaboration scripts not only need to be faded horizontally within bottom layer of Scriptlet, but also crossing the layers to be faded vertically and hence enables learner to reach beyond the current competencies, handle the entire task in question and explore new understandings from a broader and generalizes layer (e.g., Scene & Play).

So far, only a couple of studies on the fading effects have been conducted from the hierarchical perspective of collaboration scripts. Wecker and Fischer (2010, 2011) proposed a systematic schedule to operationalize the fading process, in which three types of scaffolds (external script components) that differ in their cognitive target levels including exemplifying and explanatory application supports, the sequential information prompts and the structural argument schemata were gradually faded out (in a backward fashion) to foster the internalization of the required argumentation skills. The fading occurred after the accomplishment of certain required learning tasks. Results of these pioneering work, however, indicated that in some cases at least, merely reducing the support does not seem to be enough.

3.2.3 *Fading Effects*

It's expected that fading the external collaboration script forces students to practice their knowledge and develop the necessary skills to apply it in similar situations. This argument is theoretically in line with the 'internalization' idea stated by Vygotsky (1978). In practice, however, research results on the effects of fading are sparse and inconclusive.

A study by McNeill et al. (2006) found students who received fading of the written scaffolds gave stronger explanation in terms of their reasoning compared to those with continuous support. Wecker and Fischer (2010, 2011) found that fading fosters learning only in combination with additional support such as distributed monitoring among learners. Another study (Bouyias and Demetriadis 2012) also demonstrated that enriching argumentation scripts with the peer-monitoring technique can substantially improve learning outcomes, while simply fading out the script did not seem to improve student learning in any aspect. There seems an interaction between peer monitoring and faded collaboration scripts.

Peer monitoring is often cited together with peer feedback and peer assessment (Topping 1998; Topping and Ehly 1998), in which specific components of control may be distributed among collaborating learners. Hence, one of the appealing features of the notion is that peer-monitoring provides an opportunity for students to take over more control of their own learning (King 1998). Researchers argued, therefore, fading should not be based on a fixed procedure, but on ongoing diagnosis of learners' performance; fading algorithm should not be designer controlled but learner controlled. That is an adaptive or adaptable fading, a topic we will turn into in the next two sections.

Another important limitation of the previous studies is that effects of fading were measured only by means of declarative test (target at subject matter knowledge), but not in a separate performance in transferable situations in which learners' collaborative learning was guided by their internalized script components (Wecker and Fischer 2011).

From an instructional point of view, the central point of CSCL is not only to facilitate domain-specific knowledge acquisition, but also to foster learners' mastery of desirable collaboration skills through the internalization mechanism. These skills are to some extent domain-general and might be applied in similar situations in further learning practices (Kolodner 2007). In scripted CSCL, the fading effect on learners' internalization of external scripts is still a topic not yet well investigated.

To summarize, the research on fading is still at an early stage. In order to address the effects of the 'optimal' scripts to scaffold collaborative learning, researchers and pioneers in the field should be more careful with building a best-fit procedure of fading by drawing on precise theoretical models and practical evidence. Particularly, attention should be paid to the manipulation of fading pattern (forward/backward; horizontal/vertical), fading procedure (based on ongoing diagnosis) and learner control (instead of designer control) of fading. Additionally, educators need to pay careful attention to the effects of fading collaboration scripts on script internalization.

3.3 Adaptive Scripting

3.3.1 Adaptivity

In recent years, there has been a new trend in research concerned primarily with the adaptive use of instructional supports or scaffolds (Deiglmayr and Spada 2010; Fischer et al. 2013; Walker et al. 2009). In contrast to the classic approaches to support learning with static, one-size-fits-all instructions, the promising direction of research on implementing collaboration scripts is to provide students with instructional supports "adaptively tailored to the particular needs of particular learners in a particular moment of the learning process" (Rummel and Krämer 2010, p. 3). Thus scaffolding is a flexible process in which instructional supports must be adjusted based on ongoing diagnosis of learners' performance rather than remaining constant.

The highlighted adaptive nature of scaffolding (Wood et al. 1976) is based on the intrinsic component of the scaffolding system called 'internalization', as introduced by Vygotsky (1978). In the context of scripted CSCL, this implies that once learners have internalized the collaboration script for performing a particular learning task, they will be able to perform this task independently. The specificity of the external script therefore, needs to "gradually be reduced to ensure that learners are not given instruction they actually do not need" (Collins et al. 1989, p. 456). The process of reducing the amount of external instruction (Pea 2004) is known as fading, which was introduced in the previous section. It is important to note that adaptive instruction requires dynamic fading in and fading out of instrumental support, rather than fading based on a fixed procedure designed by the researcher. Adaptive instruction, in this sense, is a temporary entity that is used to reach one's potential and then is removed when learners demonstrate their learning.

3.3.2 Intelligent Tutoring Systems

Adaptive educational systems have received some attentions in the context of Intelligent Tutoring Systems (ITS), which are computer-based instructional systems that specify what to teach and how to teach (Wenger 1987). Specifically, the intelligent tutoring system uses the intelligent tutor to assess what a student knows and needs to know and fades in/out scaffolds accordingly (Lajoie 2005); carefully monitor the problem-solving process to ensure that it stays on track and to help direct students back toward a productive solution path when needed (Merrill et al. 1995); and allow the intelligent tutor to more closely approach the benefits of individualized instructions (Murray 1999).

Synthetically, intelligent tutors typically accomplish two of the principal task characteristics: (i) monitoring the student's performance and providing context-specific instruction just when the individual student needs it, and (ii) monitoring the student's learning and selecting problem-solving activities involving knowledge goals just within the individual student's reach (Koedinger and Corbett 2006).

A number of adaptive support systems have been evaluated with beneficial effects compared to fixed support (Diziol et al. 2010; Kumar et al. 2007) in a variety of domains. Diziol and colleagues (Diziol et al. 2010; Walker et al. 2008) worked with the CTA (Cognitive Tutor Algebra), which is a tutoring system providing just-in-time feedback once an incorrect problem solving action was automatically detected. In addition, students can actively request help from the CTA, receive hints tailored to their current focus of attention.

COLER (Collaborative Learning environment for Entity-Relationship modeling) is another intelligent collaborative learning system which aims to promote group interactions and maintain balanced participation (Constantino-Gonzalez et al. 2003).When relevant opportunities for learning are found by monitoring students' participation and recognizing differences between students' individual and group solutions, the coach tries to guide students to practice collaborative skills, providing advice such as encouraging students to participate and to discuss their differences in real time.

Baghaei and colleagues (2007) developed a constraint-based intelligent tutoring system COLLECT-UML to support the acquisition of both problem-solving skills and collaboration skills, by providing feedback at different levels of detail if a difference between the group solution and individual solutions is detected. The initial level of feedback is deliberately low, which indicates whether or not a problem solution is correct, a feedback without too much detailed information is given. And the level of feedback can be increased incrementally by providing hints which provide a more detailed message.

3.3.3 Adaptive Collaboration Scripts

It must be emphasized that Intelligent Tutoring Systems (typically single user, with system feedback) on the one hand, and Computer-Supported Collaborative Learning systems (typically multiple users, usually with no system feedback) on the other hand were two more-or-less separated research fields (Scheuer et al. 2010). But the trend toward the promising integration of the advanced educational systems ITS and CSCL grows stronger because of the emerging demand for providing adaptive instructional support for learners in computer-based learning settings.

Thus far, the Intelligent Tutoring Systems being shortly reviewed above often apply Artificial Intelligence (AI) techniques to provide 'intelligent' supports for learners. In the field of CSCL, however, researchers prefer the broader term 'adaptive scripting' rather than 'intelligent tutor' because the former is in line with other pedagogical terms widely used in this field such as collaboration scripts. Additionally, by using efficient but very simple techniques, some adaptive collaboration scripts are not intelligent, even though both of these notions are certainly of interest to providing 'optimal' scaffolds to meet the personalized needs of learners.

A script is adaptive when a computer system within which collaboration takes place is able to reliably assess learners' internal scripts on the fly and adjust the external script based on the level at which the group is currently collaborating. This is achieved by building a model of optimal collaboration and comparing it to a constantly updated user model based on collaboration quality (Diziol et. al. 2010). First studies of adaptive scripting provide evidence for its effectiveness. For instance, Walker et. al. (2011) found that learners who collaborated in a peer-tutoring scenario with the support of Adaptive Peer Tutoring Assistant (APTA), an adaptive script designed to improve help-giving, showed higher levels of collaboration during the experiment than those whose collaboration was supported by a fixed peer-tutoring script; however, no effect on individual learning outcomes (measured by domain-specific subject knowledge) was found.

Despite this promising evidence, building adaptive systems (e.g., computing a user model) is quite demanding. An model of optimal collaboration needs to be defined (if possible), the on-the-fly assessment of learners' current collaboration needs to be reliable and valid, and the fit between the measured performance level and the automatically adapted instance of the external collaboration script needs to be calibrated. To achieve this, extensive intelligent tutoring technology (Walker et. al. 2011) or automated language analysis (Mu et. al. 2012) has to be developed, tested and constantly refined. Furthermore, from a theoretical perspective, adaptive scripting is not unproblematic; because adaptive scripts do not require students to make conscious decisions about the amount of support they need, which may hinder learners from regulating their collaboration unaided. The effect of adaptive scripting on collaboration processes and learning outcomes measured by domain-specific knowledge as well as domain-general skills is therefore still inconclusive and needs to be investigated in the future.

3.4 Adaptable Scripting

3.4.1 Adaptability

Apart from adaptivity, adaptability is another way to realize customized instructional support. A learning environment is called 'adaptable' when users (i.e., teachers or students) can adjust external support to meet their perceived needs (Leutner 2009). When learning with an adaptable system, it is the learners who have control over the learning processes rather than the system or the designers. By adapting the environment to suit individual needs, learners get the opportunity to self-regulate their learning, which has been regarded as an influential factor in academic achievement, especially in domain-general skill acquisition (Anderson 1987; Zimmerman and Kitsantas 1997). Skills that get acquired or practiced through adaptation experiences can further be transferred to other similar situations in further learning (Scheiter and Gerjets 2007).

Although both adaptive scripting and adaptable scripting aim to realize flexibility of CSCL scripts, they vary in the degree of 'learner control' (Merrill 1975; Scheiter and Gerjets 2007). Proponents of learner control argued that in order to improve the fit between learners and learning environments, it should not be the system or the instructor who decides what treatment is best for a single learner, it should be the learners who make decisions on how to adapt the environment to themselves (Merrill 1975; Scheiter and Gerjets 2007). Following the learner control perspectives, learning with an adaptive system which always has the optimal tactic provided makes students depend on a perfect system and unable to cope with the real world which is not so accommodating. Adaptable system, on the contrary, might enable students to select the particular tactic that is optimal for their unique configurations of aptitudes and expertises at a specific moment in time (Merrill 1975; Scheiter and Gerjets 2007).

By making decisions on how to adapt the system and to select tactics learners are expected to be self-regulated in learning, such as to monitor the process or to evaluate their performance. Self-regulated learning and the feeling of control over their own learning are also expected to help cultivate intrinsically motivated learners, according to Self-Determination Theory (Ryan and Deci 2000).

3.4.2 Learner Control with Hypermedia

As the adaptability idea originates in learner control perspectives, the accumulating research on leaner control in web-based instruction and learning with hypermedia (e.g., control over the navigation) will be taken to discuss the operationalization of adaptable scripting. When learning with web-based instructions, three factors are often subject to learner control, namely the control of learning activities and contents, control of sequence, and control of pacing (McLoughlin et al. 2006; Scheiter and Gerjets 2007).

The form of control over learning activities provides learners with the opportunity to choose the content they wish to explore. Within a specific content, learners are given the option to access different activities, such as viewing examples, practicing problems, and taking tests. Control of sequence refers to a learner's opportunity to control the order in which he/she would like to receive instruction. The learner may also skip and revisit topics. For example the forward or backward button offers users the opportunity to return to previously visited content. Leaner control of pacing allows learners to adapt instructional pace to their comfort level. Control of pacing allows learners to cover the learning content slower or faster, and allows them to take a break from instruction.

3.4.3 Adaptable Scripting

Although originated in the field of learning with hypermedia, the three forms of learner control cannot be directly translated to adaptable scripting as scripts are interaction-related guidance other than web-based instruction provided to individual learners. Nevertheless, like the control of learning activities and content in learning with hypermedia, the basic components of an external collaboration script (task specification, sequencing, and role distribution) could be subject to learner control to realize adaptability. For example, learners could be provided with the opportunity to have control over the component of role distribution in a way that learners decide rather than being assigned the roles that they would like to play in a CSCL practice.

Furthermore, other components of an external collaboration script, such as the use of prompts in the aforementioned peer-review script by Weinberger et al. (2005) could also be subject to learner control. Learners could make decision on whether they would like to use the prompts or not (by switching them off).

It is noteworthy that an external collaboration script provides both intrinsic and extrinsic constraints to collaborative learning processes, as proposed by Dillenbourg and Tchounikine (2007; see Sect. 3.1). According to the authors, extrinsic constraints of a script should be adaptable to teachers and/or students, while intrinsic constraints should be fixed and constant, because intrinsic constraints ensure the core mechanism of how a collaboration script would shape peer interaction.

By adapting an externally provided collaboration script to suit individual needs, learners are expected to benefit on learning as an adaptable script has the potential to provide the optimal external scripting level. 'Optimal' means the degree of external scripting should be based on learners internal collaboration scripts (Fischer et al. 2013). Learners vary from each other on their internal collaboration scripts (Kollar et al. 2007) within a specific CSCL environment. An external collaboration script is likely to hinder rather than foster knowledge acquisition when it provides narrow structuring that inhibits learners' application of appropriate internal collaboration script components (Fischer et al. 2013). An adaptable script, on the other hand, allows learners to turn off the unnecessary parts of the external script and to apply their appropriate internal collaboration script components to guide learning.

Furthermore, external collaboration scripts should provide the optimal scripting level on the basis of learners' domain-specific prior knowledge (Dillenbourg 2002; Reisslein et al. 2006). Learners differ in their domain-specific prior knowledge. Low-knowledge learners need in principle very much support to achieve the learning goal, whereas high-knowledge learners need little or no support at all to achieve the same goal (Kalyuga et al. 2003). For high-knowledge learners, too many scaffolds may contrariwise impede learning as they cause unnecessary workload ("redundancy effect" or "expertise reversal effect"; Kalyuga 2007). Although it was content scaffolds that have been argued to be redundant for high-knowledge learners to acquire domain-specific knowledge, scaffolds that aim at structuring interactive processes (e.g., external collaboration scripts) might also be redundant for high-knowledge learners as their processing of external collaboration scripts or even the interactive processes might be unnecessary for them to gain domain-specific knowledge (Johnson and Johnson 1999). An adaptable script might compensate the redundancy effect for high-knowledge learners to acquire domain-specific knowledge as they can switch off the redundant scaffolds to avoid over-scripting (Leutner 2009).

Thirdly, it has been argued that scaffolding should be a dynamic process based on learners' performance at any moment in time, so is fading. Therefore, fading on the fly, that is adaptive or adaptable fading, has been regarded as promising approaches, as the former provides feedbacks or supports based on dynamic assessed level of performance (Diziol et al. 2010) while the latter gives learner opportunities to fade in/out external support based on self-perceived needs (Leutner 2009). As there is a lack of empirical studies on adaptable scripting in CSCL, it is still an open question whether adaptable scripting is an encouraging approach to realizing flexibility.

Evidence from outside CSCL supports the argument that adaptable scaffolding enhanced learning compared to the inflexible counterpart. However, adaptable scripting may also be problematic as it requires learners to use high-level metacognitive skills, such as monitoring and reflection on learning; research from learning with hypermedia has shown that by no means all learners are competent in this respect (Azevedo 2005; Scheiter and Gerjets 2007). Against this background, we will report a study on adaptable scripting in Chap. 5 to provide some empirical evidence in the field of CSCL.