S-COL: A Copernican turn for the development of flexibly reusable collaboration scripts

Christof Wecker · Karsten Stegmann · Florian Bernstein · Michael J. Huber · Georg Kalus · Ingo Kollar · Sabine Rathmayer · Frank Fischer

Abstract Collaboration scripts are usually implemented as parts of a particular collaborative-learning platform. Therefore, scripts of demonstrated effectiveness are hardly used with learning platforms at other sites, and replication studies are rare. The approach of a platform-independent description language for scripts that allows for easy implementation of the same script on different platforms has not succeeded yet in making the transfer of scripts feasible. We present an alternative solution that treats the problem as a special case of providing support on top of diverse Web pages: In this case, the challenge is to trigger support based on the recognition of a Web page as belonging to a specific type of functionally equivalent pages such as the search query form or the results page of a search engine. The solution suggested has been implemented by means of a tool called S-COL (Scripting for Collaborative Online Learning) and allows for the sustainable development of scripts and scaffolds that can be used with a broad variety of content and platforms. The tool's functions are described. In order to demonstrate the feasibility and ease of script reuse with S-COL, we describe the flexible re-implementation of a collaboration script for argumentation in S-COL and its adaptation to different learning platforms. To demonstrate that a collaboration script implemented in S-COL can actually foster learning, an empirical study about the effects of a specific script for collaborative online search on learning activities is presented. The further potentials and the limitations of the S-COL approach are discussed.

C. Wecker (✉) · K. Stegmann · I. Kollar · F. Fischer
Ludwig-Maximilians-Universität München, Empirische Pädagogik und Pädagogische Psychologie,
Leopoldstraße 13, 80802 Munich, Germany
e-mail: christof.wecker@psy.lmu.de

F. Bernstein · M. J. Huber · G. Kalus · S. Rathmayer
Technische Universität München, Fakultät für Informatik, Boltzmannstr. 3, 85748 Garching, Germany

Changing the approach to the development of flexibly reusable collaboration scripts

Research on technology-based collaboration scripts has been very successful in terms of the development of a broad range of scripts that effectively foster activities and outcomes of computer-supported collaborative learning (e.g., Baker and Lund 1997; De Wever et al. 2009; Kollar et al. 2007; Rummel and Spada 2005; Schellens et al. 2007; Schoonenboom 2008; Slof et al. 2010; Stegmann et al. 2007; Weinberger et al. 2005; Weinberger et al. 2010). The growing importance of this field of research is evidenced by—among other things—the announcement of "Scripting in CSCL" as a "flash theme" in the *International Journal of Computer-Supported Collaborative Learning* (Stahl and Hesse 2007). However, technology-based collaboration scripts are usually developed exclusively for one specific, often experimental, learning platform. Neither the transfer to other experimental platforms nor the transfer into practice has been managed systematically so far. Among the current approaches to overcome these problems, the most prominent one is the attempt to develop a universal formal language (e.g., an extension of IMS-LD) for the specification of scripts to be "read in" and implemented by different collaborative-learning platforms (Weinberger et al. 2007).

In this article, we propose a different and by far simpler solution: Instead of trying to get different platforms to display functionally equivalent but platform-specific versions of the "same" script, we suggest using a pre-implemented script that is embedded in the learner's Web browser. This requires that specific components of the script be invoked whenever the browser recognizes pages displayed by a learning platform as being of the corresponding types of functionally equivalent pages. We call this the S-COL (Scripting for Collaborative Online Learning) approach to the development of flexibly reusable collaboration scripts for diverse Web content. Because of the shift mentioned, the S-COL approach can be regarded as a kind of two-fold "Copernican Turn" in script development: First, instead of the learning platform, the browser of the learner is moved into the centre of script development by making it the source of the support displayed to the learners. Second, the burden of creating flexibility is shifted from the idea of a universal formal description of a collaboration script to be generated by any learning platform to the task of triggering the appropriate components from a pre-implemented script.

We need to clarify right from the start that our claim in this article is not that the S-COL approach leads to superior learning compared to other approaches to the implementation of collaboration scripts or unstructured collaboration. S-COL simply provides a technical frame for the implementation of collaboration scripts. Accordingly, it can be used to implement a broad variety of collaboration scripts, including ineffective and even detrimental ones. What we do claim, however, is that diverse types of collaboration scripts can be implemented in S-COL with no more effort than implementing a script as part of a specific learning platform, yet with the advantage of flexible reusability with different learning platforms. This is not a claim about the psychology or instructional design of computer-supported collaborative learning. It is a claim about the power and generality of a framework for the implementation of support for computer-supported collaborative learning, which we think advances an ongoing discussion in this journal and in the CSCL community (e.g., Dillenbourg, and Tchounikine 2007; Harrer and Malzahn 2006; Kobbe et al. 2007; Miao et al. 2007; Stegmann et al. 2009; Tchounikine 2008; Weinberger et al. 2007). The focus of our claim has consequences for the evidence required to support this claim, which we will elaborate shortly.

The genesis of our approach provides some further insights into a more general problem of which the development of flexibly reusable scripts may be regarded as a special case, and into further applications of the approach: Interestingly, we hit on the solution described above when we were working on an apparently quite unrelated problem. We were looking for a way to provide context-specific support on top of varying Web pages. At some point, we came to view the problem of developing collaboration scripts for flexible reuse with different learning platforms as a special case of developing context-specific support on top of varying Web pages: From this perspective, different Web-based learning platforms are regarded simply as varying Web pages. Therefore, in order to illustrate the general idea, in the following section, we describe both our initial general problem (of providing support on top of varying Web pages) and the more specific problem (of developing scripts for flexible reuse with different learning platforms), including a review of current attempts at solutions. The third section provides a characterization of the general idea behind the comprehensive solution for the problems on both levels. Based on the insights gained, a tool was designed to solve the general problem of support on top of Web pages, and, hence, also the more specific problem of developing scripts for flexible reuse with different learning platforms. In the fourth section, we describe the main features of the S-COL tool, in particular, its graphical user interface, its functions to provide support for learning and its administration features. The fifth section provides a short description of the technical implementation of the S-COL tool directed at a more technically oriented audience. The sixth section uses two cases of collaboration scripts to provide the evidence required for our main claim: The first case shows that collaboration scripts can be implemented in S-COL with no more effort compared to an implementation as part of a specific learning platform. It deals with the re-implementation of a collaboration script for the generation of single arguments, which was originally implemented as an embedded part of a specific learning platform and was effective with respect to activities and outcomes of collaborative learning in prior studies. In order to allow for an evaluation of the claim that implementation in S-COL requires no more effort than in a learning platform, the process of implementing the script as part of the specific learning platform is compared to the process of implementing it in S-COL, and the effort required to reuse it with other learning platforms is described. The second case shows both how S-COL can be used to provide support on top of diverse Web content, and that collaboration scripts implemented in S-COL can foster specific learning activities. A specific script for such an exemplary activity, collaborative online search, is described in detail. Findings from an empirical study about its effects on learning activities during collaborative online search are reported. The final section discusses further potentials as well as limitations of the tool and indicates unresolved problems associated with the approach.

Support for learning on top of diverse Web content and reusable collaboration scripts for different learning platforms: Two unrelated problems?

As indicated in the introduction, our approach to the development of flexibly reusable collaboration scripts was developed from a rather general perspective on the problem which we adopted while looking for a way to provide support for ninth-grade high school students during collaborative online search. In an interdisciplinary collaboration involving educational psychologists and computer scientists, our goal was to develop a tool that enables learners working on different computers to conduct collaborative online searches and provides content- and role-specific support for this collaborative task to each participant. Because S-COL was designed to solve both problems, we first describe this other setting.

Support for learning on top of diverse Web content

With the rapid development of information technology and its role in work and everyday life, online search competence is becoming more and more important as a crucial prerequisite for participation in society (e.g., Bilal 2002; Ikpeze and Boyd 2007). When using a search engine, specific cognitive processes are required to conduct a successful online search (e.g., Pirolli 2005): For example, while a user is on the search query form of a search engine, he or she needs to generate a set of search terms. This set of search terms should, on the one hand, yield results containing the information needed and, on the other hand, preclude results that are irrelevant to his or her demands. While he or she is at the results page, hits need to be selected based on an evaluation of the information provided along with them (link title, text excerpt, and URL). At the pages reached from there, search strategies have to be applied to locate relevant information on the website.

Typically, novices struggle with these cognitive processes: They are less inclined to strive for an overview of information available about a specific topic and to make a plan on how to proceed in an online search (Luconi and Tabatabai 1999; Rogers and Swan 2004). Furthermore, they often choose suboptimal search terms (Tomaiuolo and Packer 1996) and are disoriented in their navigation behaviour (Ikpeze and Boyd 2007). Most importantly, novices need to learn how to evaluate information with respect to its trustworthiness and its relevance for their personal informational needs (Luconi and Tabatabai 1999; Walton and Archer 2004).

In order to support learners to master these cognitive processes, it is recommended as one among several components that learners receive “just in time” assistance (Van Merrienboer et al. 2002). The only way of delivering just-in-time assistance for online search activities that we found in the literature was by a teacher (Ikpeze and Boyd 2007). However, a teacher cannot provide just-in-time assistance to all learners in a classroom. Instead, a feasible approach would rely on computer-based scaffolds that display just-in-time assistance to each individual learner (Pea 2004; Puntambekar and Hübscher 2005; Quintana et al. 2004). Furthermore, collaborative online search may be an appropriate setting for fostering the acquisition of online search competence: Research has shown beneficial effects of collaborative online search on the strategies employed, although not yet on learning (Lazonder 2005).

So far, there are few solutions for providing support on top of existing Web pages for individuals or groups of learners. One such solution is Greasemonkey, a browser plug-in that allows for changes of the content of a Web page as it is displayed to the user (Greasemonkey 2010). This functionality could be used, in principle, to incorporate scaffolds into existing Web pages, such as reflection prompts on a Google results page. However, first of all, this approach is fragile to any changes in Web pages that may occur at any point in time. In addition, the Web pages traversed during online search are very diverse, both in terms of content and technical structure. Accordingly, it is rather difficult to develop scaffolds that can be integrated in any Web page encountered during an online search.

Therefore, a desideratum for supporting the acquisition of online search competence is a technical solution for implementing scaffolds for individuals or collaboration scripts that can guide collaborating learners strategically during the different stages of an online search depending on where they are in the search process, but apply to any kind of topic as well as to any kind of Web pages encountered.

Reusable technology-based collaboration scripts for different learning platforms

As we have indicated above, collaboration scripts have typically been implemented as embedded parts of specific, often experimental, learning platforms. In some of our own studies (e.g., Stegmann et al. 2007; Weinberger et al. 2005), we used a discussion board developed by ourselves because this allowed for the easy implementation of the collaboration scripts under investigation in the learning platform itself: For example, a script for the generation of arguments can easily be implemented by means of prompts and separate textboxes for the parts of an elaborated argument. These textboxes can be embedded in the form for entering messages and their contents can be composed into one continuous message before posting the contribution (see Fig. 3, part a). Thus, the collaboration scripts were always part and parcel of the learning platform itself.

With the accumulation of findings about beneficial effects of collaboration scripts, the question arose how collaboration scripts that have been developed and tested in the context of a specific learning platform can be transferred and reused in the context of other learning platforms. The problem was framed as the task to integrate the original collaboration script into other learning platforms, that is, to get other learning platforms to display the components of the original collaboration script as part of the new learning platform. This approach led the way to the development of a universal formal language for the description of collaboration scripts (Kobbe et al. 2007; cf. also Kollar et al. 2006). This language is intended to be used for the specification of collaboration scripts that can be "imported" by different learning platforms and used as a basis to display the components of the original script as an embedded part of their interface. This universal scripting language accommodates a small but still comprehensive number of components and mechanisms of computer-supported collaboration scripts: The components are participants, activities, roles, resources, and groups; the mechanisms comprise task distribution, group formation, and sequencing.

On the basis of this universal scripting language, a graphical modelling tool for designing new collaboration scripts has been developed (Harrer and Malzahn 2006). As an output, the modelling tool produces an IMS-LD file, that is, a file that can be read by all learning platforms that support the IMS Global Learning Consortium Standards (cf. Miao et al. 2007). Based on these ideas, a functional framework for accelerating the implementation of scripts represented in IMS-LD for devices such as tabletop displays or mobile phones has been developed (Stegmann et al. 2009). However, we are not aware of any learning platform that is generally available and can import and implement a description of a collaboration script as an IMS-LD file using this IMS-LD extension.

A further promising approach to provide a universal language for the scaffolding of collaborative learning is the "Learning Activity Management System" (LAMS). LAMS provides a graphical modelling tool for sequencing a variety of predefined activities (e.g., a chat tool followed by an individual phase, followed by a plenary discussion). The sequences of activities designed with LAMS can be integrated into several learning platforms such as *Moodle*, *Sakai*, or *Blackboard*. However, the activities that can be sequenced are restricted by the activities available in the graphical authoring tool. Furthermore, the activities cannot be "micro-scripted," that is, learners can be prompted to discuss, but specific activities during discussion, such as the formulation of arguments, cannot be supported.

Another approach is "ManyScripts" (Dillenbourg and Hong 2008): This tool offers teachers an environment to adapt a set of specific scripts according to their own needs, especially their own learning material. At the moment, the Concept Grid, Argue Graph (Dillenbourg and Jermann 2007), and Ice (Dillenbourg and Hong 2008) scripts are available (Manyscripts 2010). For example, the Argue Graph script forms groups of students with divergent opinions with respect to a specific domain (e.g., drug use in sports). To adapt the Argue Graph script, teachers can easily define their own questions that will be used to form these divergent groups. However, the ManyScripts environment is a stand-alone learning platform. A native integration into other learning platforms has not been a goal and is not supported yet.

Consequently, currently neither the universal scripting language and graphical modelling approaches, nor the ManyScripts approach are suitable for developing new scripts and implementing them on a broad range of different learning platforms. A framework that effectively supports the reusability of technology-based collaboration scripts is not available so far. The transfer of a collaboration script from one collaborative-learning platform to another is still hampered by the need to adapt and integrate the script into the new learning platform. Therefore, a solution for using scripts developed and tested on one learning platform on other platforms is also still a desideratum.

The basic idea for a comprehensive solution

At first glance, it might seem that the two problems, that is, the development of scaffolds and collaboration scripts for collaborative learning on top of diverse Web content and the reusability of technology-based collaboration scripts, are unrelated. But at a second glance, these two problems are closely related to each other: The reuse of technology-supported collaboration scripts is hampered by the endless variety of possible learning platforms in which scripts should be implemented. If different learning platforms which typically can be accessed via a Web browser are viewed as but one special case of diverse Web content, the problem of transferring collaboration scripts between platforms becomes a special case of the problem of the development of scaffolds and collaboration scripts applicable to diverse Web content.

The basic idea to solve this problem is to implement scaffolds and collaboration scripts as part of the browser and trigger them based on the recognition of types of functionally equivalent pages on the Internet or within the learning platform. With respect to the example of support for online search, this approach takes advantage of the fact that any search engine such as *Google*, *Yahoo!,* or *Bing* consists of a form for entering a search query that leads to a series of results pages with a common structure. From here, the user can reach Web pages that may contain the information he or she seeks. Accordingly, there are three types of functionally equivalent pages that users have to traverse during online search whatever Web search engine they may be using: (1) the search query form, (2) the results page, and (3) the external Web pages reached from the results page. If a component of the browser manages to recognize these three *types* of page, it can trigger specific kinds of support embedded in the browser. A search query form, for instance, typically contains one (or sometimes several) text field(s) for entering search terms and a button for starting the query. Such page-specific components in combination with the specific URL of the page can be used as a basis for the recognition of the page types. As each of these page types corresponds to a specific stage of an online search, specific support for the cognitive processes associated with each of these stages (e.g., Pirolli 2005) can be provided.

The situation is similar in the case of collaboration scripts for online discussions on collaborative-learning platforms. Many learning platforms such as *Moodle*, *Sakai,* and *Blackboard* contain asynchronous discussion boards. Any discussion board contains functionally equivalent pages such as the form for entering messages. In most learning platforms, the form for entering messages consists of functionally equivalent parts such as separate fields for the message and its title as well as a button for posting the message. Again, if a component of the browser manages to recognize this *type* of page and the *types* of its component objects, the components of a collaboration script pre-implemented in the browser can be triggered. The prompts and textboxes constituting the collaboration script can be displayed in a separate area of the browser window, and the contents of the single textboxes can be composed and sent to the message field when posting the message. The advantage of this approach lies in the fact that it allows for the use of a library of already implemented collaboration scripts contained in the browser that can be used with a broad variety of Web-based collaboration tools.

Main features of S-COL

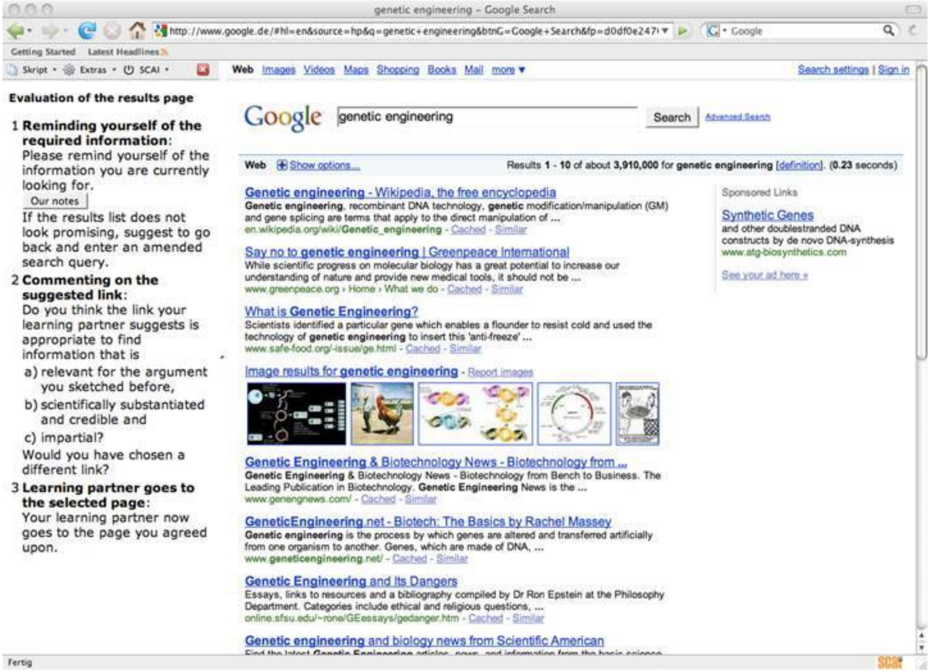
We now turn to the implementation of these ideas as part of a tool we developed in order to demonstrate that the two interconnected problems described above can be solved in this way and to create a technical frame for providing support for computer-supported collaborative (and individual) learning on the Web.

The graphical user interface

The tool was implemented as a browser plug-in. Accordingly, the main part of its graphical user interface is the browser itself. The area of the browser used for displaying Web pages is broken up in two parts (see Fig. 1). The area on the right-hand side is called the “browsing area.” It exhibits exactly the same behaviour as a standard Web browser: It can present any kind of Web page, and the user can navigate by using links and menu elements of the browser such as the home, forward, and backward buttons or entering a URL. The part on the left-hand side is called the “scaffolding area.” Its size is flexibly adaptable both by the user dragging its border as well as by programmed functions (in JavaScript). Furthermore, it can be invoked and hidden by a function key. Its content can be flexibly designed using HTML. The content of the scaffolding area (textboxes, buttons, etc.) can “interact” with objects in the browsing area. For instance, information from the browsing area such as the content of tables and textboxes or the URL of the actually displayed Web page can be read out. Furthermore, the browsing area can be controlled and manipulated by the tool by means of automatically posting text into forms, activating buttons, or even by navigating to an arbitrary URL. The scaffolding area moreover contains a menu bar (right above “Evaluation of the results page” in Fig. 1) providing functionalities such as loading collaboration scripts or scaffolds and configuring the navigation behaviour of the tool (see below).

Tool functions

The tool provides two main functions: It can display support in the scaffolding area depending on the type of content displayed in the browsing area and on the role that a learner has been assigned before, and it allows for collaborative navigation on the Web using several interconnected browsers.


genetic engineering – Google Search
http://www.google.de/#hl=en&source=hp&q=genetic+engineering&btnG=Google+Search&fp=d0df0e247
Google
Getting Started
Latest Headlines
Skript
Extras
SCAI
Web
Images
Videos
Maps
Shopping
Books
Mail
more ▾
Search settings | Sign in
Google
genetic engineering
Search
Advanced Search
Fertig

Evaluation of the results page

1 **Reminding yourself of the required information**:
Please remind yourself of the information you are currently looking for.
Our notes
If the results list does not look promising, suggest to go back and enter an amended search query.

2 **Commenting on the suggested link**:
Do you think the link your learning partner suggests is appropriate to find information that is
a) relevant for the argument you sketched before,
b) scientifically substantiated and credible and
c) impartial?
Would you have chosen a different link?

3 **Learning partner goes to the selected page**:
Your learning partner now goes to the page you agreed upon.

Web Show options... Results **1** - **10** of about **3,910,000** for **genetic engineering** [definition]. (**0.23** seconds)

Genetic engineering - Wikipedia, the free encyclopedia
Genetic engineering, recombinant DNA technology, **genetic** modification/manipulation (GM) and gene splicing are terms that apply to the direct manipulation of ...
en.wikipedia.org/wiki/**Genetic_engineering** - Cached - Similar

Say no to **genetic engineering** | Greenpeace International
While scientific progress on molecular biology has a great potential to increase our understanding of nature and provide new medical tools, it should not be ...
www.greenpeace.org › Home › What we do - Cached - Similar

What is **Genetic Engineering?**
Scientists identified a particular gene which enables a flounder to resist cold and used the technology of **genetic engineering** to insert this 'anti-freeze' ...
www.safe-food.org/-issue/ge.html - Cached - Similar

Image results for **genetic engineering** - Report images

Genetic Engineering & Biotechnology News - Biotechnology from ...
Genetic Engineering & Biotechnology News - Biotechnology from Bench to Business. The Leading Publication in Biotechnology. **Genetic Engineering** News is the ...
www.genengnews.com/ - Cached - Similar

GeneticEngineering.net - Biotech: The Basics by Rachel Massey
Genetic engineering is the process by which genes are altered and transferred artificially from one organism to another. Genes, which are made of DNA, ...
www.**geneticengineering**.net/ - Cached - Similar

Genetic Engineering and Its Dangers
Essays, links to resources and a bibliography compiled by Dr Ron Epstein at the Philosophy Department. Categories include ethical and religious questions, ...
online.sfsu.edu/~rone/GEessays/gedanger.htm - Cached - Similar

Genetic engineering and biology news from Scientific American

Sponsored Links

Synthetic Genes
and other doublestranded DNA constructs by de novo DNA-synthesis
www.atg-biosynthetics.com

See your ad here »

Fig. 1 The S-COL graphical user interface with the implementation of a collaboration script for collaborative online search

Content- and role-specific support The display of content- and role-specific support for learners collaborating with this tool requires the recognition of types of Web pages and their component objects. For example, to scaffold the writing of arguments, a script needs the information whether the current page is a page for the composition of a new message or not. Based on this, scaffolds and components of collaboration scripts are displayed in the scaffolding area of the tool. The recognition of the page types is achieved by means of a template file that contains a description of each variant of every page type as well as its components. To identify a page type, both the URL and the content of the page (including elements such as textboxes or buttons) can be used. For example, the template file may refer to the URL of the Google variant of the search query form to identify this page as the page type "search query form." The template file can contain the same information for other search engines as well as similar information for the other page types traversed during an online search. Based on this, the contents of the scaffolding area are adapted in a content-specific way according to the page type recognized, and in a role-specific way according to the role that a person may have been assigned before. This adapting includes the possibility to configure the scaffolding area to disappear if no scaffolds or components of collaboration scripts should be provided.

Collaborative Web browsing The tool, furthermore, allows for collaborative Web browsing. This is to say that all learners belonging to the same group can automatically view the same Web pages in their browsers. The assignment to groups is done via a dialog window for group formation (described in more detail below). Each member of a group has the opportunity to "lead the whole group" to a different Web page: By simply using his or her browser the usual way, that is, by clicking on links, menu elements, or entering a new URL, one member brings a new page onto the screens of all members of the group. If a learner opens a new tab, new tabs will be opened in all connected browsers.

The collaborative Web-browsing function can be adapted in several ways. In principle, each user can dissociate him- or herself from collaborative Web browsing. This comprises an active and a passive component: On the one hand, a user may switch off the function that "sends" his or her navigation actions to the other group members. This has the effect that his or her navigation actions no longer influence what is displayed on the computer screens of the other members of the group, so he or she can no longer "lead" the group to other pages. On the other hand, he or she may switch off the function that "receives" the navigation actions of the other group members. This has the effect that navigation actions of other group members no longer influence what is displayed on the respective group member's computer screen, so he or she no longer "follows" other group members to other pages. S-COL also offers a JavaScript function that can be used by script developers to switch these communication functions on and off, for example, depending on the page type currently displayed in the browsing area: Learners might be dissociated from collaborative navigation whenever one group member logs into a learning platform for individual study of learning materials, and reconnected as soon as all members are outside of this platform again. Furthermore, the rights to manually switch on and off the "sending" and "receiving" of navigation actions can be configured globally to allow teachers to control their students' options during collaborative-learning tasks on the Internet.

Group and script administration

The tool provides administration functionalities for the use of teachers or experimenters. These include group formation and the selection of collaboration scripts to be displayed in the learners' browsers.

Group settings The tool contains a dialog window for the formation of groups (cf. Kobbe et al. 2007) that also allows roles to be assigned to individual members of the groups and to select scripts and scaffolds from the library (see below) to be displayed in the scaffolding area for individual users. Group size is unlimited in principle. This window can be used to change the group-related settings of any of the users in the same network from any browser with an activated S-COL plug-in. However, it is password protected in order to restrict access to specific persons (e.g., a teacher or experimenter).

Scaffold and collaboration script library Furthermore, S-COL has a scaffold and collaboration script library that contains the different scaffolds and scripts that can be invoked in the scaffolding area. Currently, this is implemented as a folder that contains all the files with the contents of the scaffolding area from which a teacher or experimenter can select. For the future, we plan to either develop or integrate a scaffold and script editor. This will allow for easy configuration of the template file used for the recognition of page types as well as the organization of hierarchical structure of page types, subtypes, and their component objects. It will simplify the linking of scaffolds and components of collaboration scripts to page types and subtypes. It will also permit roles and states of counters to cause the fading of scaffolds or scripts when students have already practiced certain skills a specified number of times.

Technical implementation of S-COL

General architecture S-COL's current implementation is a plug-in for the Firefox browser that is part of a client-server architecture necessary for the collaborative Web-browsing functionality. In this setup, each browser in a network of S-COL-endowed computers can directly access the Internet (see Fig. 2; the further components of the architecture are explained shortly).

Implementation of the plug-in and the server The graphical user interface of the browser plug-in (that integrates S-COL's scaffolding area and configuration dialogs into the graphical user interface of the Firefox) was programmed using the XUL language for the Firefox browser. The communication between the clients is currently implemented in C++ and Java; the server component is a stand-alone Java program. S-COL offers specific JavaScript functions that can be used in HTML files loaded into the scaffolding area and provide access to the different features of S-COL, that is, controlling the content and appearance of the scaffolding and browsing areas, manipulation of Web pages, group and role changes, communication, logging, and handling of variables.

Content- and role-specific support Providing support in the scaffolding area that is sensitive to both the content displayed as well as the role assigned to the learner in the group settings requires (a) an implementation of a collaboration script to be displayed in the scaffolding area and (b) the recognition of the type of the page displayed in the browsing window. The content of the scaffolding area consists of HTML files, typically also including JavaScript code. The recognition of the type of the page displayed in the browsing window is based on the template file mentioned above which contains a description of each variant of every page type and its components in the Resource Description Framework (RDF). Both the URL and the Document Object Model (DOM)—along with XPath expressions or the ID of control elements such as the textbox for the search terms on the Google search query form—can be used for the identification of the page type. Based on the information contained in the RDF file, a JavaScript function yields the type of the page currently displayed. Depending on the values returned by this function (and potentially also the role that the person using this computer is assigned), the contents of the scaffolding area are selected by JavaScript code contained in the HTML file loaded into the scaffolding area.

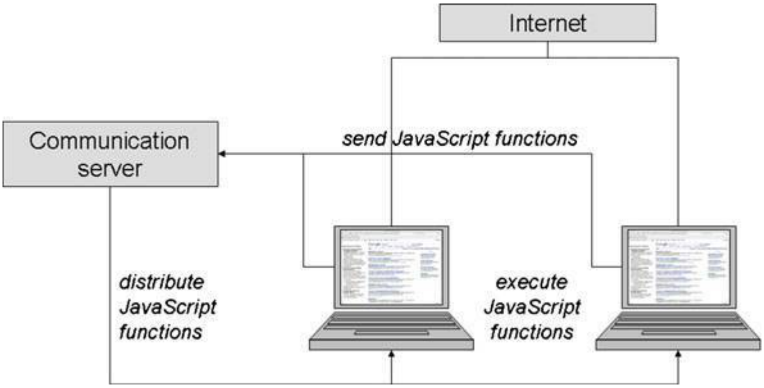

Internet
Communication server
send JavaScript functions
distribute JavaScript functions
execute JavaScript functions

Fig. 2 The S-COL environment for the collaborative Web-browsing function

Collaborative Web browsing The collaborative Web-browsing function is based on JavaScript functions that send messages to all connected Web browsers with activated S-COL plug-in. To coordinate the behaviour of browsers connected via the Web-browsing functionality, the S-COL plug-in in each browser sends messages including, for example, JavaScript functions to be executed to all other S-COL plug-ins in the network via a communication server (see Fig. 2). These messages contain all the information required to make other browsers assigned to the same group “follow” the navigation in the browser from which they were sent. Upon receipt, they are evaluated by all S-COL plug-ins in the same group as the sending browser, thereby directing the “following” browsers to the corresponding pages. This messaging system for the implementation of the collaborative Web-browsing functionality can also be used for synchronizing the scaffolds, that is, in case that a scaffold should be faded after a specific number of activities of a certain type, or for implementing a chat system between the connected browsers. In general, they can be used for distributing any information necessary for manipulating the scaffolding and browsing areas of all connected browsers.

Application: Two cases

So far, S-COL has not been used by practitioners, but it has constituted the framework for the implementation of support in several studies about Web-based inquiry learning, design-based learning, and case-based learning in CSCL environments at the university and high school levels that took place in Egypt (El-Refai et al. 2010) and Germany (e.g., Wecker et al. 2010). In order to provide evidence that S-COL solves the problems outlined above, we selected two cases of its application: First, to show that the reuse of scripts with different learning platforms is actually feasible and comparably easy with S-COL, in the first case we describe the process of implementing a technology-based collaboration script for the generation of single arguments as an embedded part of a specific learning platform without S-COL, and report on the process of re-implementing this collaboration script in S-COL. In order to allow for an evaluation of our claim, we also describe the possibility and the effort required to reuse both implementation variants of the script with other learning platforms.

The second case serves two purposes: First, it provides empirical evidence that collaboration scripts implemented in S-COL can foster specific learning activities. Second, the specific script used in this study illustrates how S-COL can be used to provide content- and role-specific support on top of diverse Web content.

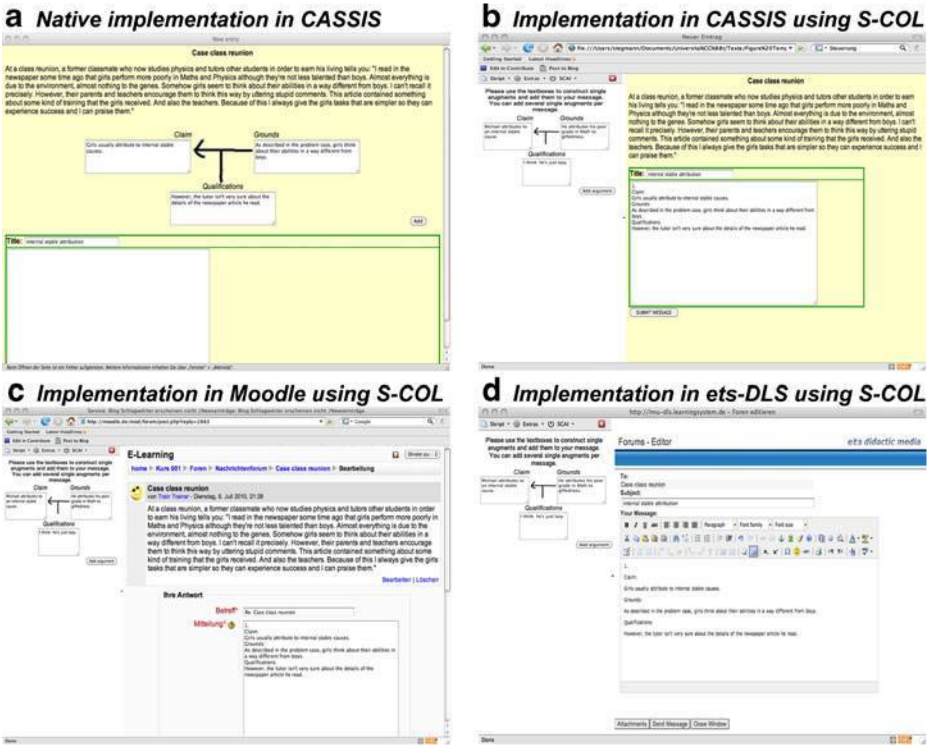
Collaboration scripts for argumentation in different online learning platforms

The collaboration script considered in this first example is one from a rather large number of collaboration scripts investigated in a series of studies (e.g., Stegmann et al. 2007; Weinberger et al. 2005): So far, more than 35 different collaboration scripts, combinations of collaboration scripts or translations have been developed, and experimental studies with about 1,000 students have been conducted. All these collaboration scripts were originally implemented as an embedded part of the experimental *CASSIS* (Computer-supported Argumentation Supported by Scripts—experimental Implementation System) learning platform (described in Clark et al. 2008; Clark et al. 2010). On this platform, three students per group discussed problem cases in a customized asynchronous text-based discussion board while sitting in different laboratory rooms. The interface allowed for the exchange of text messages that resemble emails (for details on the methodology and the results of these experiments, see Stegmann et al. 2007; Weinberger et al. 2005).

In the following, the focus will be on one specific script: the script for the generation of arguments. In its original, embedded implementation as part and parcel of the learning platform, this script structured a student's formulation of an argument by means of the learning platform's interface for the composition of new messages. It provided input textboxes for a claim, grounds, and qualifications (see Fig. 3, part a). Each textbox of the interface had to be filled in by the learners. By clicking on the "add" button, the contents of the three input textboxes were combined into a pre-specified textual structure of the argument. Learners were not limited to using the three input textboxes for constructing arguments. They could also write questions, comments, or expressions of emotions directly into the main input textbox.

For the original implementation, new templates for the discussion board used in *CASSIS* had to be created. The script for the generation of arguments was directly embedded in the template for new messages. Although this script only consists of three textboxes and the function for merging the content of the three textboxes and pasting this argument into the main textbox, changes of templates such as these typically take about 2 h.

For these adaptations, the software developer needs some specific knowledge about the *CASSIS* platform, including where the relevant templates can be found. Because, in this case, the software developer of the script for the generation of arguments was the developer



Case class reunion

At a class reunion, a former classmate who now studies physics and tutors other students in order to earn his living tells you: "I read in the newspaper some time ago that girls perform more poorly in Maths and Physics although they're not less talented than boys. Almost everything is due to the environment, almost nothing to the genes. Somehow girls seem to think about their abilities in a way different from boys. I can't recall it precisely. However, their parents and teachers encourage them to think this way by uttering stupid comments. This article contained something about some kind of training that the girls received. And also the teachers. Because of this I always give the girls tasks that are simpler so they can experience success and I can praise them."

Claim: Girls usually attribute to internal stable causes.

Grounds: As described in the problem case, girls think about their abilities in a way different from boys.

Qualifications: However, the tutor isn't very sure about the details of the newspaper article he read.

Title: internal stable attribution

Please use the textboxes to construct single arugments and add them to your message. You can add several single arugments per message.

Claim: Michael attributes to an internal stable cause.

Grounds: He attributes his poor grade in Math to giftedness.

Qualifications: I think he's just lazy.

Title: internal stable attribution

1.
Claim
Girls usually attribute to internal stable causes.
Grounds
As described in the problem case, girls think about their abilities in a way different from boys.
Qualifications
However, the tutor isn't very sure about the details of the newspaper article he read.

E-Learning

Home ▸ Kurs 001 ▸ Foren ▸ Nachrichtenforum ▸ Case class reunion ▸ Bearbeitung

Case class reunion
von Train Trainer - Dienstag, 6. Juli 2010, 21:28

Bearbeiten | Löschen

Ihre Antwort

Betreff*: Re: Case class reunion

Mitteilung*:
1.
Claim
Girls usually attribute to internal stable causes.
Grounds
As described in the problem case, girls think about their abilities in a way different from boys.
Qualifications
However, the tutor isn't very sure about the details of the newspaper article he read.

Forums - Editor

To: Case class reunion

Subject: internal stable attribution

Your Message:
1.
Claim
Girls usually attribute to internal stable causes.
Grounds
As described in the problem case, girls think about their abilities in a way different from boys.
Qualifications
However, the tutor isn't very sure about the details of the newspaper article he read.

Fig. 3 Implementations of the script for the generation of arguments: **a** Native implementation in *CASSIS*, **b** implementation with S-COL in *CASSIS*, **c** in *Moodle*, and **d** in *ets-dls*

of *CASSIS*, he did not have to acquire this knowledge first. However, in all other cases, developers will need some time (at least 4 h for a system like *Moodle*) to become acquainted with technical documentations. Overall, the implementation of the rather simple script for the generation of arguments took one day. More complicated scripts (i.e., ones that involve role distribution, rotation of activities, etc.; cf. Weinberger et al. 2005) require a much longer development phase of typically several weeks.

The data gathered in experiments with this original implementation demonstrated that the script for the generation of arguments had a positive effect on learning activities and outcomes: Learners supported by this script constructed more formally complete arguments and acquired more knowledge about the generation of arguments than learners who were not supported by the script (Stegmann et al. 2007).

However, exactly the same functionality can be provided within S-COL. In order to demonstrate this, the script for the generation of arguments was re-implemented as a HTML page loaded into the scaffolding area of S-COL (see Fig. 3, part b). Provided that the information required for identifying the elements of a certain collaborative-learning platform's interface is contained in the template file of S-COL for page type recognition, this implementation can be used with *CASSIS, Moodle*, or many other learning platforms (see Fig. 3 for examples using two further learning platforms, i.e., *Moodle*–part c–and *ets-dls*–part d): The content of the textboxes in the scaffolding area can be incrementally added to the message textbox of the collaborative-learning platform in the browsing area by clicking on the "add" button in the scaffolding area. The scaffolding area is invoked on the basis of a page definition of the message composition form of the particular platform in the template file, which also makes the control elements of the form accessible to the components of the script.

This re-implementation took about 1 h for all three learning platforms (*CASSIS, Moodle*, and *ets-dls*). The original implementation was used as the starting point, because the main functionality was not changed. Most of the time for the re-implantation was spent on the adaptation of the S-COL template file for the identification of Web pages and their elements. Developers who work with S-COL for the first time may estimate about 4 h for getting familiar with the S-COL-API and the template file. Overall, the implementation of the script for the generation of arguments for three learning platforms took less than one day. In general, once a collaboration script is developed for S-COL, the reuse with other learning platforms is quite easy and takes a minimal amount of time.

While the script cannot be reused with other learning platforms if implemented as an embedded part of the learning platform as in the original case, this new implementation of the script allows for comparably easy reuse. This reuse requires nothing more than the adaptation of the template file and comprises the following three steps:

(1) Inserting unique features of the message composition form of the online learning platform (e.g., URL, control elements such as input textboxes or buttons) into the template file.
(2) Identifying the control elements of the online learning platform's message composition form in the template file.
(3) Distributing an S-COL version with the new template file and the script file for installation on all computers used for collaboration.

Thus, the adaptation of collaboration scripts that has been the focus of intensive efforts in prior research has not only become feasible but is now reduced to a couple of simple steps. What is important for future use of scripts implemented in S-COL by practitioners is the fact that their adaptation to a specific learning platform is even possible without any administrative access to the learning platform in use. All that is needed is information about unique features of specific pages of the learning platform such as the URL or the ID of control elements like input textboxes or buttons, which can be identified using Firefox Add-Ons such as DOM Inspector (Hewitt and Aillon 2003) or XPather (Zigo 2009). Therefore, nearly all HTML-based learning platforms, including those from commercial providers, can be supported. Compared to other approaches for the transfer of collaboration scripts that place rather high demands on the online learning platform to be used (such as the description of collaboration scripts in IMS-LD), this makes the reuse of collaboration scripts a real possibility that is even in the reach of practitioners without any programming or modelling expertise.

Collaboration scripts for collaborative online search

The purpose of the second case is to provide empirical evidence that a script that is implemented in S-COL actually fosters learning. To this purpose, a field study in real-world classrooms was conducted. Furthermore, it serves as an example in which most of the functionalities of S-COL are used, that is, providing content- and role-specific support on top of varying Web pages and collaborative Web browsing.

Research question This study was designed to investigate the effects of a collaboration script for online search on students' collaborative strategy use during online search in the context of an extended inquiry-learning curriculum (Wecker et al. 2010). Here we focus on the specific question whether this collaboration script, which was implemented in S-COL, can support the learners in focusing on activities that are important during a specific stage of online search and can prevent them from engaging in other activities that are less functional during this stage. The initial stage of collaborative online search was chosen as an exemplary focus because the success of the online search as a whole is considered to be strongly influenced by the search goals that learners set themselves during this early stage.

Method: Participants and design The sample consisted of 93 students (46 girls and 45 boys, 2 students did not provide this information) from four 9th-grade classes from three urban high schools in Germany. Their mean age was 14.72 years ($SD=0.64$). Two intact classes were assigned to each of two conditions differing in instructional support in a quasi-experimental design: In two classes ($N_1=42$), the students worked without a collaboration script, whereas in the other two classes ($N_2=51$) they were supported by means of a collaboration script.

Curriculum unit and instructional setting The effects of the instructional support by the collaboration script were studied in the context of a Web-based inquiry-oriented curriculum unit that stretched over 5 weeks and contained ten consecutive biology lessons. The topic covered was "Genetics and Genetic Engineering." The unit was centred around a continued discussion about potential benefits and dangers of so-called "green" Genetic Engineering, that is, the genetic modification of plants for agricultural purposes. Each student was equipped with a laptop computer that was the same in each session.

The unit started with an introductory lesson in which students acquired background knowledge about inheritance from an online "library" comprising fundamental information about the topics of Genetics and Genetic Engineering. This online library was implemented as a module on the Web-based Inquiry Science Environment (WISE; Slotta 2004; Slotta and Linn 2009). After this lesson, there were three inquiry cycles that lasted two lessons each. Each cycle covered a different aspect of the whole issue—economic, ecological, and health-related aspects of green Genetic Engineering—and consisted of three phases: First, the students always had to gather scientific background knowledge relevant to the current aspect of the topic from the online library. This task was performed in dyads. Then, the dyads collaboratively conducted online searches to support their own stance toward the current aspect with arguments based on information about recent research on the possibilities and consequences of green Genetic Engineering. During this phase, the manipulation of instructional support by means of the collaboration script was applied. The final phase of each cycle was a classroom discussion in which the students presented and discussed their arguments and eventually experienced the need to further substantiate their views in subsequent cycles. In both conditions, the teacher provided an introduction to important steps and aspects to consider during online search before the online search phase in the first cycle. In both conditions, this introduction covered all the information contained in the collaboration script used in the experimental condition.

In both experimental conditions, during the collaborative online search phase, the two students in each group sat next to each other, so they could talk to each other face-to-face. They both had their own laptop computer, which was connected to the learning partner's computer via the collaborative Web-browsing functionality of S-COL. The students had no opportunity to deactivate this configuration, so whoever of the two students in a group clicked on a link in his or her browser navigated both of them on the Web. Accordingly, they had to coordinate their collaborative online search activities, that is, talk and decide about the next page to navigate to before each navigation activity.

Independent variable In order to provide an illustrative example of S-COL's functionality for content- and role-specific support, we now describe the implementation of the collaboration script for collaborative online search during the second phase of each inquiry cycle in some detail. This script was based on problems typically occurring during online search that were collected based on an exploratory think-aloud study with persons who were more versus less experienced with respect to online search (Kollar et al. 2009) and results from empirical studies of online search competence (e.g., Luconi and Tabatabai, 1999). It was implemented as complementary text prompts in the scaffolding area of S-COL in the browsers of both group members (see Fig. 1). The students in this condition had no opportunity to permanently deactivate S-COL's scaffolding area. In each dyad, there were two roles (A and B) that the learners were asked to switch after returning to the search engine from any other Web page encountered during the search activities.

The students were asked to start the collaborative online search phase in each of the three cycles by moving to a specific Web page, which triggered a set of prompts in the scaffolding area for reflection about the argument they wanted to pursue ("*sketch of the initial argument*" stage): Learners A and B were both required to come up with an initial idea for an argument and select one of them as a starting point for their online search. Then learner A was requested to describe this initial argument in a note field, while B was asked to knock together a sketch of the information required to support it (for instance, a study showing that an alleged effect of Genetic Engineering has been observed, or a genetic explanation how a hypothesized effect could be possible). To complete this stage, first A was prompted to present the initial argument formulation to B and improve it according to B's suggestions, then B was asked to present the sketch of the information required to support this argument and amend it according to A's comments.

When the learners jointly moved to the search query form of Google ("*selection of search terms*" stage), A was prompted to come up with a set of search terms and present them to and discuss them with B. Meanwhile, B had the task to first recall the information that they had decided to look for, and then comment on A's suggestions for the search terms with respect to their likelihood of yielding both suitable and unsuitable hits. At the results page ("*evaluation of the results page*" stage), the scaffolding area asked learner A to scan through the list of results and evaluate them with respect to relevance, credibility, scientific support, and impartiality on the basis of the title, the text excerpt, and the URL provided by the search engine, and to suggest the page to visit. Learner B was prompted again to recall the information that they were looking for and to comment on the pages A suggested with respect to the criteria mentioned. When the group navigated to one of the websites found by means of the search engine ("*localization of the information*" stage), learner A received prompts in the scaffolding area about how to localize the required information on the website (e.g., by using search functions on the page), to present the information in his or her own words to learner B, and to discuss with B how to proceed. Learner B had the task to suggest to A to return to earlier steps of the search if he or she had the impression that the current page was not promising and to comment on the information presented by A with respect to the criteria mentioned. B's task at this stage also comprised the documentation of the information retrieved (including the URL as a reference) and the discussion of the next step to take.

When the dyad agreed that their online search had been successful, any of the two members could click on a button that invoked support for the formulation of an argumentation to be used in the subsequent plenary discussion ("*refinement of the argument*" stage). The prompts asked B to summarize all the information collected during the previous online search in his or her own words and compose a written summary of this argumentation based on A's comments. Learner A received prompts to comment on B's spoken summary with respect to its persuasiveness and possible counterarguments and to provide suggestions for improvement for the written version of their argumentation. If they had the impression that they needed further information, they could return to the search engine for further research.

In the condition without the collaboration script, the scaffolding area of S-COL was permanently deactivated, whereas the collaborative Web-browsing functionality was used in the same way as in the condition with the collaboration script. It should be kept in mind that the learners in this condition received the same introduction to online search by the teacher as the learners in the control group, covering all the information contained in the collaboration script described above.

Data sources and dependent variables The learners' verbal interactions and activities on their computers were recorded using screen-and-audio capturing software. As the collaboration script applied to the second phase (the online-search phase) of each of the three inquiry cycles (which lasted 45, 30, and 30 min, respectively), a time sample of 10 min per cycle was analyzed for the occurrence of the activities suggested by the collaboration script. In accordance with our exemplary focus on the early stage of online search dealing with the development of an initial search goal, these time samples were taken from the beginning of each of the three online-search phases.

A coding scheme, which included all of the 31 activities suggested by the collaboration script as mutually exclusive codes, was used to code segments of 10 s of length separately for both members of each dyad. That is, for each time segment, the coding identified which activities the learner predominantly performed during that segment. The material was analyzed by three coders who were evenly distributed over the two conditions. On an 11% subsample of the data that was coded by all three of them their agreement was acceptable (Coders 1 and 2: percentage agreement 70%; Cohen's κ=.67; Coders 1 and 3: percentage agreement 68%;

Cohen's κ=.64; Coders 2 and 3: percentage agreement 91%; Cohen's κ=.89). For each of the five stages of the script—the *sketch of the initial argument* (Cronbach's α=.82), the *selection of search terms* (Cronbach's α=.65), the *evaluation of the results page* (Cronbach's α=.61), the *localization of the information* required on the Web pages retrieved (Cronbach's α=.66), and the *refinement of the argument* (Cronbach's α=.57) –, the proportion of time spent on the corresponding activities was determined, averaged over the two members of each group.

For the purpose of qualitative analyses to illustrate the collaborative online searches of dyads in the two conditions, interactions during online search phases of typical dyads were transcribed. From the control condition a dyad with a comparably low number of activities that were functional during the *sketch of the initial argument* stage was selected for presentation in this article; in the experimental condition a dyad with a comparably high number of activities that were functional during the *sketch of the initial argument* stage was selected. These transcriptions include both verbal interactions and navigation behaviour.

Results and discussion We first present excerpts from the transcriptions from two mixed-gender dyads, one from the control condition and one from the experimental condition. Both are taken from the collaborative online search phase of the second topical cycle concerned with ecological aspects of green Genetic Engineering and have been translated from German. The learners' utterances appear in normal print, their actions on the computer are printed in italics. The first excerpt shows how the dyad from the control condition started the first online search in this cycle:

> B: Do we go directly to Google?
>
> ...
>
> B: Just do "green Genetic Engineering"!
>
> ...
>
> A: So, where do we go? Genetic Engineering?
>
> B: Just do – err, text editor! I have a – Just go to – Wait, look, I'll show you something!
>
> *On the Google search query form: enters "transgen" (the name of a website encountered before).*
>
> *Clicks on first hit on Google results page ("transgen").*
>
> A: What's that?
>
> B: There we can search.

Quite different from what was suggested by the collaboration script in the experimental condition, these learners from the control condition did not spend any time reflecting about their positions on the issue, initial ideas for an argument to elaborate, and the kind of information that might be helpful for elaboration. Furthermore, there is no discussion about the very broad and unspecific search term suggested by learner B. Instead the dyad started uncritically browsing a website provided by a pressure group one of the two learners suggested. This uncritical use of information and even avoidance of processing on part of at least one member of the dyad is exemplified by a further excerpt from this group:

> A: Come on, now you just copy this, this here, look: this here!
>
> B: Adam, please!
>
> A: Now, this is all copied now.
>
> *Marks text with his mouse.*
>
> B: That's something only you do, I don't do that.
>
> A: No, hey, you write for us.

Apart from the apparently low level of cognitive processing and discussion of the information retrieved, the dyad also experienced severe problems in their collaborative navigation due to the lack of coordination of their activities:

After being moved to a different Web page because of a click of his learning partner:
A: What have you done now? I see, you (went) somewhere else.
B: Hey, man, please, now stop this!
A: Yes.
B: Man, I have just been reading!
A: I see!
B: Man, let me finish reading quickly!
A: Yes, okay. You are ( ) – Look: If we say, we are currently reading, //
B: // Okay.
A: then the other one can go nowhere else. But if we, if we don't say anything, then the other one can change, okay?
B: Okay.
A: So, I am reading now.
B: Me too.
A: What are you reading?
B: I'm reading "Precious [...] " whatever.
A: "Resources Water".

As evidenced by this excerpt, the lack of structuring for collaborative online search may not only result in superficial processing of information, but also in considerable friction with respect to the coordination of activities: It took the learners quite a while to negotiate how to coordinate themselves, before they could align their activities to achieve a shared focus. In sum, this dyad exhibited hardly any activities that could be considered functional for successful online search.

In contrast, a prototypical dyad in the experimental condition supported by the collaboration script started the online search with a decision about a position to pursue and some reflection about what would be needed to support it:

C: Are you for or against?
D: "For" there's probably more, right?
C: Dunno. But most people are against, but I think that it's more for it.
...
Or is it all the same to you?
D: To me it – To me it's actually all the same.
C: So I'm for it, and the argument would be: Hm, for ecology there's no argument. Or I say that it's harmless.
D: Yes.
C: Good. Hm. How could you prove something that isn't? And now here we're supposed to – Okay, you ske– make the argument, and I am supposed to write down the information needed. Err.
D: So "for".
C: Yes.
...
D: And why? Because it's harmless?
C: Yes. So –
...

> *C writes: "scientific studies that prove that the effects of modified plants on."*
> ...
> So, I thought we need – Or do you already have an argument?
> D: Yes, because it's harmless.
> C: Okay. Good, I think it's good. "Information required." My information required: We need studies that show that on a normal scale this is harmless, I mean, every plant changes, and then there are effects, but that, of course it does effect, but that it is no problem.
> D: Okay.

In this dyad, learner C took responsibility for performing the activities suggested by the script with the effect that the group agreed about a position to pursue and a—still quite unspecific— argument. Furthermore, learner C contributed a reasonably narrow characterization of information that would help them substantiate and specify their initial general argument. This characterization helped learner C refine the search terms suggested by learner D and make them more likely to yield scientific studies at the subsequent stage of their online search:

> *They go to the Google search query form.*
> D: Okay, what do we think: What should we look for?
> D: Err, "Effects of Genetic Engineering" – "on nature"?
> C: We could just look whether we simply just – Then, yes, mhm – Either we look in a way that we search "counter", that is that we simply enter "arguments against", then there is certainly also (some) from – argument with ecology, and if that's a good one, we just can change opinion, and if it's a bad one, we say there is nothing against? Or just somehow that one looks – "scientific studies pro" – dunno, maybe simply "scientific study green Genetic Engineering"?
> D: Yes.

Here, learner C, in accordance with his role assigned by the script, reactivated their initial search goal to find scientific studies. Also, at the stage of trying to localize and select relevant information on a specific Web page, this awareness of the information required helped the group judging the appropriateness of information encountered:

> C: Well, that is quite nice – So they *are* against, but somehow they only write there are uninvestigated risks that could happen. They don't really have something against, but they say, here: "The significance of unintended secondary effects".
> D: Yes.
> C: Here: "unintended", but they have not investigated it. So, actually they say the risk is that – one can't investigate everything in this way, the risks, that is, they have, there are risks that we really haven't investigated.
> D: So, there could happen some, but it needn't be.

In sum, this dyad from the experimental group performed many of the activities suggested by the script, such as negotiating their position, selecting an initial argument to elaborate, and writing down a characterization of the information needed. During later steps of their online search, they clearly benefited from the specificity of their search goal when they were selecting search terms and evaluating information.

In order to test whether the differences between the patterns of activities exhibited by these two groups hold more generally, the quantitative measures derived from the codings of the learners' activities were statistically analyzed using the whole sample. The level of significance was set at 5% for all tests. The descriptive results for the time spent on the activities suggested by each of the five stages of the collaboration script are displayed in Table 1. As can be seen, the collaboration script slightly decreased the overall amount of activities suggested by the script. Specifically, it increased the amount of time spent on the sketch of the initial argument and decreased the amount of time spent on the evaluation of the results page, the localization of information, and the refinement of the argument, whereas the time spent on the selection of search terms appears unaffected.

A multivariate analysis of variance with the five indicators for the proportion of time spent on the activities suggested by each of the five stages of the collaboration script as dependent variables and classrooms nested within the two levels of instructional support (collaboration script vs. no collaboration script) as well as instructional support as independent factors yielded a significant effect of the collaboration script, Wilks Λ=.63; $F(5;\ 85)=9.91$; $p<.001$; partial $\eta^2=.37$. Separate univariate analyses of variance were conducted in order to clarify this result. The effects on the amount of time spent on the *sketch of the initial argument*, $F(1;\ 89)=16.01$; $p<.001$; partial $\eta^2=.15$, and the *refinement of the argument* were significant, $F(1;\ 89)=17.70$; $p=.001$; partial $\eta^2=.17$, whereas there was no significant effect on the *selection of search terms*, $F(1;\ 89)<1$; $p=0.99$; partial $\eta^2<.001$, the *evaluation of the results page*, $F(1;\ 89)=2.68$; $p=.11$; partial $\eta^2=.03$, and the *localization of information*, $F(1;\ 89)=1.88$; $p=.17$; partial $\eta^2=.02$.

These findings indicate that in an early stage of collaborative online search analyzed here (the first 10 min of each online search phase), the collaboration script was effective in increasing the amount of time spent on activities that are functional at this early stage. Such activities included sketching an initial argument to start with and reflecting on information that might be helpful to refine it. The script was also effective in decreasing the amount of time spent on activities that are functional at later stages of online search, that is, evaluating the results page of the search engine and refining the initial argument by adding information.

More generally, this study provides evidence that a collaboration script implemented in S-COL can actually foster specific collaborative-learning activities in real-world classrooms. The two main functionalities of S-COL—providing content- and role-specific support on top of varying Web pages and collaborative Web browsing—were used in this case to create a scenario for meaningful collaborative learning of important skills on the Web.

Table 1 Means and standard deviations of the proportions of time spent on the activities suggested by each of the five stages of the collaboration script

	No collaboration script		Collaboration script	
	M	*SD*	*M*	*SD*
Activities suggested by the script (overall; %)	61.0	13.0	57.3	11.1
Sketch of the initial argument (%)	3.0	3.4	7.0	6.0
Selection of search terms (%)	7.0	3.1	7.0	2.4
Evaluation of the results page (%)	9.1	6.0	7.4	3.6
Localization of information (%)	38.3	11.1	35.4	13.4
Refinement of the argument (%)	3.7	5.1	0.5	1.5

Further potentials, limitations, and unsolved problems

In this contribution, we suggested an approach to the development of flexible and reusable scaffolds and collaboration scripts that draws on earlier conceptual work and empirical research on scaffolds and collaboration scripts, but makes a big step forward by changing perspectives on the problem. While we think that S-COL is already a very helpful tool for research on technology-supported collaboration scripts, we are also quite sure that it has still more potential. As described above, important steps have been taken toward a universal scripting language (cf. Harrer and Malzahn 2006). An implementation of an interpreter of this IMS-LD based language in S-COL would allow the graphical modelling of new collaboration scripts *and* their broad application in many Web-based learning platforms. Subsequently, the transfer of successful scripts as well as systematic research on scripts would be much easier to conduct. Also, approaches like "ManyScripts" (Dillenbourg and Hong 2008) could be integrated into the scaffolding area of S-COL. Thereby, the tool may bridge the gap between the development of new technology-based collaboration scripts and their systematic application in research and practice. In our interdisciplinary collaboration between educational psychologists and computer scientists, furthermore, we quickly learned that this approach is not necessarily restricted to collaboration scripts but might be applied to other situations in which collaborative learning involving varying Web content should be supported by just-in-time scaffolding.

In the wake of the features of S-COL that allow for an easy implementation of collaboration scripts, several additional functions to support research were developed. For example, if it is necessary to analyze the online search activities of learners, usually screen recordings have to be coded. S-COL can log the browsing behaviour including all clicks, mouse movements, and the content (i.e., the DOM) of all visited Web pages. S-COL may also be used to transfer identification data from pre-test to post-test in field studies, even without awareness of the participants, thereby reducing the likelihood of mistakes and data loss. Furthermore, the tool can be used to administer process measurements (e.g., for measures of cognitive load or flow experiences) in the context of an application of the Experience Sampling Method (ESM) during learning activities: By means of a function using the messaging system of S-COL that implements collaborative Web browsing, each individual browser in a network can be triggered to open a short questionnaire in a pop-up window.

Some limitations and open issues also remain to be discussed. The main limitation of S-COL is the restriction of its full range of features to HTML-based learning platforms: Currently, S-COL can only "talk to" elements such as textboxes and buttons on HTML pages. However, a growing share of learning platforms now implements Java- or Flash-based communication tools. Accordingly, the use of these tools can hardly be scaffolded with S-COL. Besides, the possibility of the logging of all user events including the DOM constitutes a hazard with respect to the protection of user data: S-COL could easily be configured to trace all Web activities of a user, even on the level of keystrokes, and send these data to a server anywhere on the Web. However, an S-COL version without unsafe tracing functions could easily be derived from the current version.

While these issues still need to be addressed and there is still further potential to be actualized by connecting S-COL to previous progresses in scaffold and collaboration script development (such as the description of collaboration scripts in IMS-LD and the graphical modelling tool), the idea behind S-COL might be a big step in the development of flexibly reusable scaffolds and collaboration scripts for diverse Web content. Among other things, it is much easier now for both researchers and practitioners to exchange scaffolds and scripts for purposes of replication and practical use.

References

Baker, M., & Lund, K. (1997). Promoting reflective interactions in a CSCL environment. *Journal of Computer Assisted Learning, 13*(3), 175–193.

Bilal, D. (2002). Children’s use of the Yahooligans! Web search engine: III. Cognitive and physical behaviors on fully self-generated search tasks. *Journal of the American Society for Information Science and Technology, 53*(13), 1170–1183.

Clark, D., Sampson, V., Stegmann, K., Marttunen, M., Kollar, I., Janssen, H., et al. (2010). Online learning environments, scientific argumentation, and 21st century skills. In B. Ertl (Ed.), *E-collaborative knowledge construction—learning from computer-supported and virtual environments* (pp. 1–39). Hershey: IGI Global.

Clark, D. B., Stegmann, K., Weinberger, A., Menekse, M., & Erkens, G. (2008). Technology-enhanced learning environments to support students’ argumentation. In S. Erduran & M. P. Jiménez-Aleixandre (Eds.), *Argumentation in science education* (pp. 217–243). Dordrecht: Springer.

De Wever, B., Van Keer, H., Schellens, T., & Valcke, M. (2009). Structuring asynchronous discussion groups: The impact of role assignment and self-assessment on students’ levels of knowledge construction through social negotiation. *Journal of Computer Assisted Learning, 25*, 177–188.

Dillenbourg, P., & Jermann, P. (2007). Designing integrative scripts. In F. Fischer, I. Kollar, H. Mandl, & J. M. Haake (Eds.), *Scripting computer supported communication of knowledge: Cognitive, computational and educational perspectives* (pp. 275–301). New York: Springer.

Dillenbourg, P., & Hong, F. (2008). The mechanics of CSCL macro scripts. *International Journal of Computer-Supported Collaborative Learning, 3*(1), 5–23.

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

El-Refai, W., Kollar, I., & Fischer, F. (2010, August/September). *Supporting Online Design-Based Learning with Collaboration Scripts and Incomplete Concept Maps*. Paper to be presented at the meeting of EARLI SIG 6 (“Instructional Design for motivated and competent learning in a digital world”) and EARLI SIG 7 (“Learning and Instruction with Computers)”. Ulm, August, 30 – September, 1, 2010.

Greasemonkey (2010). Retrieved 22 June 2010, http://en.wikipedia.org/wiki/Greasemonkey.

Harrer, A., & Malzahn, N. (2006). Bridging the gap—towards a graphical modeling language for learning designs and collaboration scripts of various granularities. *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)* (pp. 296–300). Kerkrade, the Netherlands: IEEE Computer Society Press.

Hewitt, J., & Aillon, C. (2003). DOM Inspector. Retrieved 22 June 2010, from https://developer.mozilla.org/En/DOM_Inspector.

Ikpeze, C. H., & Boyd, F. B. (2007). Web-based inquiry learning: Facilitating thoughtful literacy with WebQuests. *The Reading Teacher, 60*(7), 644–654.

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

Kollar, I., Fischer, F., & Hesse, F. W. (2006). Collaboration scripts—A conceptual analysis. *Educational Psychology Review, 18*, 159–185.

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

Kollar, I., Wecker, C., & Fischer, F. (2009, August). *Supporting high-school students’ scientific literacy through scripted collaborative inquiry—a use-inspired basic research program*. Paper presented at the 13th Biennial Conference for Learning and Instruction of the European Association for Research on Learning and Instruction (Earli) “Fostering Communities of Learners", Amsterdam, August, 25–29, 2009.

Lazonder, A. W. (2005). Do two heads search better than one? Effects of student collaboration on web search behaviour and search outcomes. *British Journal of Educational Technology, 36*(3), 465–475.

Luconi, F., & Tabatabai, D. (1999). Searching the Web: Expert-Novice Differences in a Problem-Solving Context. Retrieved 22 June 2010, http://www.eric.ed.gov/PDFS/ED429619.pdf.

Manyscripts (2010). Retrieved 22 June 2010, http://manyscripts.epfl.ch/.

Miao, Y., Harrer, A., Hoeksema, K., & Hoppe, H. U. (2007). Modeling CSCL scripts—a reflection on learning design approaches. In F. Fischer, I. Kollar, H. Mandl, & J. M. Haake (Eds.), *Scripting computer-supported collaborative learning: Cognitive, computational and educational perspectives* (pp. 117–135). New York: Springer.

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

Pirolli, P. (2005). Rational analyses of information foraging on the Web. *Cognitive Science, 29*, 343–373.

Puntambekar, S., & Hübscher, R. (2005). Tools for scaffolding students in a complex learning environment: What have we gained and what have we missed? *Educational Psychologist, 40*(1), 1–12.

Quintana, C., Reiser, B. J., Davis, E. A., Krajcik, J., Fretz, E., Duncan, R. G., et al. (2004). A scaffolding design framework for software to support science inquiry. *The Journal of the Learning Sciences, 13*(3), 337–386.

Rogers, D., & Swan, K. (2004). Self-regulated learning and Internet searching. *Teachers College Record, 106* (9), 1804–1824.

Rummel, N., & Spada, H. (2005). Learning to collaborate: An instructional approach to promoting collaborative problem solving in computer-mediated settings. *The Journal of the Learning Sciences, 14* (2), 201–241.

Schellens, T., Van Keer, H., De Wever, B., & Valcke, M. (2007). Scripting by assigning roles: Does it improve knowledge construction in asynchronous discussion groups? *International Journal of Computer-Supported Collaborative Learning, 2*(2–3), 225–246.

Schoonenboom, J. (2008). The effect of a script and an interface in grounding discussions. *International Journal of Computer-Supported Collaborative Learning, 3*(3), 327–341.

Slof, B., Erkens, G., Kirschner, P. A., Jaspers, J. G. M., & Janssen, J. (2010). Guiding students' online complex learning-task behavior through representational scripting. *Computers in Human Behavior, 26* (5), 927–939.

Slotta, J. D. (2004). The Web-Based Inquiry Science Environment (WISE): Scaffolding knowledge integration in the science classroom. In M. C. Linn, E. A. Davis, & P. Bell (Eds.), *Internet environments for science education* (pp. 203–231). Mahwah: Erlbaum.

Slotta, J. D., & Linn, M. C. (2009). *WISE science: Web-based inquiry in the classroom*. New York: Teachers College Press.

Stahl, G., & Hesse, F. (2007). Welcome to the future: ijCSCL volume 2. *International Journal of Computer-Supported Collaborative Learning, 2*(1), 1–5.

Stegmann, K., Streng, S., Halbinger, M., Koch, J., Fischer, F., & Hussmann, H. (2009). eXtremely Simple Scripting (XSS): A framework to speed up the development of computer-supported collaboration scripts. In A. Dimitracopoulou, C. O'Malley, D. Suthers, & P. Reimann (Eds.), *Computer Supported Collaborative Learning Practices: CSCL2009 Community Events Proceedings* (pp. 195–197). ISLS.

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

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

Tomaiuolo, N. G., & Packer, J. G. (1996). Web search engines: Key to locating information for all users or only the cognoscienti. In *Online Information 96. Proceedings of the International Online Information Meeting*, Retrieved 22 June 2010, http://www.eric.ed.gov/PDFS/ED411811.pdf.

Van Merrienboer, J. J. G., Clark, R. E., & de Crock, M. B. M. (2002). Blueprints for complex learning: The 4C/ID model. *Educational Technology Research and Development, 50*(2), 39–64.

Walton, M., & Archer, A. (2004). The Web and information literacy: Scaffolding the use of web sources in a project-based curriculum. *British Journal of Educational Technology, 35*(2), 173–186.

Wecker, C., Kollar, I, Fischer, F., & Prechtl, H. (2010, June/July). Fostering online search competence and domain-specific knowledge in inquiry classrooms: Effects of continuous and fading collaboration scripts. Paper presented at the 9th International Conference of the Learning Sciences (ICLS), Chicago, June, 29 – July, 2, 2010.

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

Weinberger, A., Ronen, M., Tchounikine, P., Harrer, A., Dillenbourg, P., Haake, J., et al. (2007). Languages and platforms for CSCL Scripts. In C. Chinn, G. Erkens, & S. Puntambekar (Eds.), *[Proceedings of the] CSCL 2007. Rutgers, The State University of New Jersey, New Brunswick, New Jersey, USA, July 16th - July 21st, 2007* (pp. 831–832). International Society of the Learning Sciences.

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

Zigo, V. (2009). XPather Documentation. Retrieved 22 June 2010, from http://xpath.alephzarro.com/documentation.