Computer-Supported Collaborative Learning

Scripting Computer-Supported Collaborative Learning

Cognitive, Computational and Educational Perspectives

Edited by
Frank Fischer
Ingo Kollar
Heinz Mandl
Jorg M. Haake

Scripting Computer-Supported Collaborative Learning

COMPUTER-SUPPORTED COLLABORATIVE LEARNING

VOLUME 6

The *Computer-Supported Collaborative Learning Book Series* is for people working in the CSCL field. The scope of the series extends to 'collaborative learning' in its broadest sense; the term is used for situations ranging from two individuals performing a task together, during a short period of time, to groups of 200 students following the same course and interacting via electronic mail. This variety also concerns the computational tools used in learning: elaborated graphical whiteboards support peer interaction, while more rudimentary text-based discussion forums are used for large group interaction. The series will integrate issues related to CSCL such as collaborative problem solving, collaborative learning without computers, negotiation patterns outside collaborative tasks, and many other relevant topics. It will also cover computational issues such as models, algorithms or architectures which support innovative functions relevant to CSCL systems.

The edited volumes and monographs to be published in this series offer authors who have carried out interesting research work the opportunity to integrate various pieces of their recent work into a larger framework.

The titles published in this series are listed at the end of this volume.

F. Fischer
I. Kollar
H. Mandl
J. M. Haake
(Editors)

Scripting Computer-Supported Collaborative Learning

Cognitive, Computational and Educational Perspectives

Springer

Frank Fischer, Knowledge Media Research Center, Germany
Ingo Kollar, Knowledge Media Research Center, Germany
Heinz Mandl, University of Munich, Germany
Jörg M. Haake, University in H agen, Germany

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

ISBN 13: 978-0-387-36947-1 e-ISBN 13: 978-0-387-36949-5 Printed on acid-free paper.
ISBN 10: 0-387-36947-3 e-ISBN 10: 0-387-36949-X
Library of Congress control number: 2006935529

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

springer.com

CONTENTS

PART I
COGNITIVE PERSPECTIVES

CHAPTER 3

Rummel, N., & Spada, H.

CHAPTER 4

Runde, A., Bromme, R., & Jucks, R.

CHAPTER 5

Nückles, M., Ertelt, A., Wittwer, J., & Renkl, A.

PART II
COMPUTATIONAL PERSPECTIVES

CHAPTER 8

Miao, Y., Harrer, A., Hoeksema, K., & Hoppe, H. U.

CHAPTER 9

Lauer, T., & Trahasch, S.

CHAPTER 10

Haake, J. M., & Pfister, H. R.

CHAPTER 11 – Discussion

Suthers, D. D.

PART III
EDUCATIONAL PERSPECTIVES

CHAPTER 12

Weinberger, A., Stegmann, K., Fischer, F., & Mandl, H.

CHAPTER 13

Ertl, B., Kopp, B., & Mandl, H.

CHAPTER 14

Kolodner, J.

CHAPTER 15 – Discussion

Häkkinen, P., & Mäkitalo-Siegl, K.

PART IV
INTERDISCIPLINARY PERSPECTIVES

CHAPTER 16
Dillenbourg, P., & Jermann, P.

CHAPTER 17

Carmien, S., Kollar, I., Fischer, G., & Fischer, F.

CHAPTER 18 – Discussion

Stahl, G.

Contributing Authors

GERARDO AYALA
Universidad de las Américas-Puebla (Mexico)

RAINER BROMME
University of Muenster (Germany)

STEFAN CARMIEN
University of Colorado, Boulder (U.S.A.)

PIERRE DILLENBOURG
Ecole Polytechnique Fédérale de Lausanne (EPFL) (Switzerland)

ANNA ERTELT
University of Freiburg (Germany)

BERNHARD ERTL
Bundeswehr University of Munich (Germany)

FRANK FISCHER
Knowledge Media Research Center, Tuebingen (Germany)

GERHARD FISCHER
University of Colorado, Boulder (U.S.A.)

JÖRG M. HAAKE
University in Hagen (Germany)

PÄIVI HÄKKINEN
University of Jyväskylä (Finland)

ANDREAS HARRER
University of Duisburg-Essen (Germany)

FRIEDRICH W. HESSE
Media Research Center, Tuebingen (Germany)

KAY HOEKSEMA
University of Duisburg-Essen (Germany)

HEINZ ULRICH HOPPE
University of Duisburg-Essen (Germany)

PATRICK JERMANN
Ecole Polytechnique Fédérale de Lausanne (EPFL) (Switzerland)

REGINA JUCKS
University of Muenster (Germany)

ALISON KING
California State University, San Marcos (U.S.A.)

INGO KOLLAR
Knowledge Media Research Center, Tuebingen (Germany)

JANET L. KOLODNER
College of Computing/Georgia Institute of Technology, Atlanta (U.S.A.)

BIRGITTA KOPP
University of Munich (Germany)

TOBIAS LAUER
University of Freiburg (Germany)

KATI MÄKITALO-SIEGL
Knowledge Media Research Center, Tuebingen (Germany)

HEINZ MANDL
University of Munich (Germany)

YONGWU MIAO
University of Duisburg-Essen (Germany)

MATTHIAS NÜCKLES
University of Freiburg (Germany)

HANS-RÜDIGER PFISTER
University of Lueneburg (Germany)

ALEXANDER RENKL
University of Freiburg (Germany)

NIKOL RUMMEL
University of Freiburg (Germany)

ANNE RUNDE
University of Muenster (Germany)

HANS SPADA
University of Freiburg (Germany)

GERRY STAHL
Drexel University, Philadelphia (U.S.A.)

KARSTEN STEGMANN
University of Tuebingen (Germany)

DANIEL D. SUTHERS
University of Hawaii (U.S.A.)

STEPHAN TRAHASCH
University of Freiburg (Germany)

ARMIN WEINBERGER
Knowledge Media Research Center, Tuebingen (Germany)

JÖRG WITTWER
University of Freiburg (Germany)

Preface

Research on Computer-Supported Collaborative Learning (CSCL) is a multidisciplinary field that can be located at the intersection of cognitive psychology, computer science, and education. Yet, the different epistemological and theoretical backgrounds of these disciplines often make fruitful exchange between them difficult. To put it in other words, CSCL urgently needs to develop and use boundary concepts that can bring psychology, computer science, and education closer together to improve cumulative research and development of computer-supported learning environments. This book focuses on one term we believe has the potential to become a real boundary concept in CSCL – "scripting". However, the term script has different connotations and traditions in the different disciplines: For cognitive psychology, scripts are individual memory structures that guide us in understanding and acting in particular situations. In computer science, scripts are used by designers to create and adapt system behaviour and to guide learners through complex work or learning processes. In education, scripts are instructional scaffolds that structure the learning processes within groups of learners. From these different connotations, it becomes clear that efforts have to be taken among researchers in the three disciplines to more precisely describe their specific notions of what scripts are (and what they are not) and to more systematically relate theory and research on scripts between the three disciplines.

It is our belief that this book represents the state of the art of research on scripting computer-supported collaborative learning and that it provides a starting point for the development of a common understanding of scripting in CSCL. As such, we intend it to be a valuable resource for research, development and teaching.

The making of the book was facilitated by a variety of international research networks. We would therefore like to thank the European Network of Excellence Kaleidoscope within the 6th Framework Programme of the EU for providing the opportunity for authors of the book to communicate and exchange their ideas and findings. Likewise, the German-American Network in the Field of Technology-Supported Education (collaboratively funded by the German DFG and the U.S. NSF) made it possible that researchers from the U.S. and Germany came together several times to explore potential synergies with respect to research on technology-enhanced learning. Also, the DFG-funded Priority Program "Net-based Knowledge Communication in Groups" was helpful in bringing researchers with different perspectives about scripts together and thus facilitated interdisciplinary discourse on scripting CSCL. Concerning the preparation of the manuscript, Silvia Früh and Thomas Klebes shall be thanked for their precise and engaged work. Also, we would like to thank Marie Sheldon and Mary Panarelli at Springer for supporting the publication process in such a goal-oriented and yet flexible and supportive way. Finally, we would like to thank an anonymous reviewer for providing us with helpful comments concerning the structure and the content of the book.

Tübingen, Munich, Hagen

Frank Fischer, Ingo Kollar, Heinz Mandl, & Jörg M. Haake

Chapter 1 – Introduction

PERSPECTIVES ON COLLABORATION SCRIPTS

Frank Fischer[1], Ingo Kollar[1], Jörg M. Haake[2], and Heinz Mandl[3]
[1]Knowledge Media Research Center, Tübingen; [2]FernUniversität, Hagen; [3]Ludwig-Maximilians-Universität, Munich

Collaborative learning is widely regarded as an effective instructional approach. It has been shown that by having learners collaborate with peers, they may come to externalize their knowledge, monitor each others' learning, and jointly negotiate meaning. These activities may trigger significant individual cognitive processes that ultimately lead to individual knowledge construction (see Webb & Palincsar, 1996). On a theoretical level, the benefits of collaborative learning are often described in Piagetian and Vygotskyan terms: in collaborative learning, it is argued, that "socio-cognitive conflicts" (Doise & Mugny, 1984) may arise. When learners then try to resolve these conflicts, individual learning is stimulated. In addition, researchers claim that collaborators can provide one another with a "zone of proximal development" (Vygotsky, 1978). This is achieved by mutually scaffolding their activity such that they can perform slightly above their current level of competence.

However, a rich body of research has demonstrated that learners often do not collaborate well if left to their own devices. For example, they often do not sufficiently reference each others' contributions (e.g., Hewitt, 2005), do not build well-grounded arguments (e.g., Sandoval & Millwood, 2005), have problems in effectively coordinating their joint efforts (e.g., Gräsel, Fischer, Bruhn, & Mandl, 2001), and engage in quick and superficial consensus-building (e.g., Weinberger, 2003). Some of these problems may even be further augmented when collaboration is mediated by computers. This is because learners may be overstrained by dealing with the computer interface, may have less communication channels available than in natural face-to-face settings, or could adopt a social loafing behavior more easily (see Bromme, Hesse, & Spada, 2005). In other words, to be successful, collaborative learning – be it a face-to-face experience or mediated by a computer – needs to be supported by adequate scaffolds.

In this book, all contributions center around one scaffolding approach that has repeatedly been demonstrated as successful in improving both collaborative learning processes and the individual learning outcomes mediated by these processes: the collaboration script approach. At a fundamental level, collaboration scripts are an instructional means that aim to make collaboration processes more productive (see Dillenbourg & Jermann, this volume). However, different researchers have different notions concerning what specific aspects of collaboration should be subject to scripting. In general, two major focuses of collaboration script research and design can be distinguished. Firstly, collaboration scripts may focus on what may be termed the "macro level" of collaboration (see Ayala, this volume; Haake & Pfister, this volume; Dillenbourg & Jermann, this volume). This involves the organizational issues of collaborative learning concerned with questions such as "Who collaborates with whom?", "What is the group's task?", or "What roles are distributed among the learners?". Secondly, other researchers are more concerned with the micro level of collaboration, designing scripts that provide support for specific activities. In this approach important questions are, for example, "What specific collaboration processes are the learners supposed to engage in?" or "How should learners specifically conduct these activities?".

Especially when looking at collaboration script approaches that provide support at the micro level of collaboration, one critical issue is the question of how coercive the script should be. Indeed, critics of overly coercive collaboration scripts often express their concern that learners are given too little freedom for productive collaboration to take place ("over-scripting"; Dillenbourg, 2002). On the other hand, it often seems necessary to provide learners with scripts that impose some structure to enable them to engage in productive interaction. However, striking a balance between taking the "freedom from" and providing the "freedom to" learners is a delicate issue. This topic is important for the design of collaboration scripts for both face-to-face and computer-mediated script approaches.

Although collaboration scripts were introduced long before the development of computer technologies as ubiquitous educational tools (see King, this volume), they have become a major topic in the research community on computer-supported collaborative learning (CSCL; e.g., Dillenbourg, 2002; Kollar, Fischer, & Slotta, 2005). One main reason for this seems to be that the script concept has a unique potential as a "boundary concept" among the different disciplines that intersect in CSCL: cognitive psychology, computer science, and education. Because the script concept plays a specific role in all three disciplines, it can serve as an anchor in multidisciplinary and interdisciplinary discourse. Furthermore, the script context may contribute to the development of a scientific community with a clear focus on knowledge accumulation. As the collection of chapters in this book illustrates, all three disciplines can make significant contributions towards the design, theoretical rationales, and practical implementations of collaboration scripts. However, the different connotations of the term script in cognitive psychology (e.g., Schank & Abelson, 1977), computer science (Hoppe, Gassner, Mühlenbrock, & Tewissen, 2000), and education (e.g., O'Donnell & Dansereau, 1992) also provide a challenge for discourse between disciplines. Cognitive psychology uses the term script primarily to describe individual memory structures that guide learners in their understanding and behavior in particular event sequences such as a restaurant visit (e.g., Schank & Abelson, 1977). Computer science tries to develop formal language and devices that support designers and practitioners (i.e., teachers) in easily setting up collaboration scripts for computer-mediated learning. Education is interested in the design of collaboration scripts that can be implemented in formal or informal learning settings and effectively guide and improve collaboration processes and subsequent individual learning. Thus, interdisciplinary discourse often faces challenges because the different meanings of the script concept need to be negotiated. This negotiation is necessary so that a joint understanding of what scripts are and what they are not can ultimately emerge. Therefore, this book also provides interdisciplinary approaches to scripting that can lay the groundwork for future interdisciplinary discourse about scripting.

In summary, this book aims to bring these different disciplinary approaches on scripting closer together. We have collected advanced script approaches from (1) cognitive psychology, (2) computer science, and (3) education. Moreover, to demonstrate the opportunities for using synergy to apply the script concept between perspectives, we have included recent (4) interdisciplinary CSCL approaches to scripting. In the following paragraphs, we briefly introduce each of the perspectives and then provide a brief summary of the approaches that are included in the respective sections of this book.

(1) From a *cognitive psychology perspective*, scripts are culturally shared as well as personal knowledge and memory structures that help people act and understand actions and action sequences in specific every-day situations (Schank & Abelson, 1977). An example is the "restaurant script", which specifies how an individual should act when going for dinner in a restaurant (entering the restaurant, waiting for the waiter, following the waiter to a table, waiting for the menu, choosing a meal, placing an order, etc.). When applied to collaborative learning, the question is, how collaboration scripts can support the acquisition or the activation of appropriate cognitive scripts on how to collaborate. When applied to novel knowledge communication situations in the Web, questions arise such as: What scripts do collaborators apply in novel communication and collaboration contexts? What scripts are

effective in overcoming barriers and biases in novel communication situations? How do new scripts for these novel situations evolve over time?

In the first chapter, *King* emphasizes that collaboration is not effective as such for learning but is mediated by specific cognitive and metacognitive activities of the individual. These activities, however, can be triggered by specific collaborative activities (e.g., explaining, argumentation). Since research has shown that these activities rarely occur spontaneously, King identifies scripts that have proven to be effective in structuring interaction to improve the individual's learning in a group. In analyzing four script examples for collaborative learning, King describes how these beneficial collaborative activities can be guided, clustered to roles, and sequenced to optimally activate and guide cognitive and metacognitive processes.

The chapter by *Rummel and Spada* addresses the question of whether collaboration scripts can be internalized. Do learners really learn to collaborate when supported by a script? In an interdisciplinary problem-solving scenario in a videoconferencing environment, learners first collaborated using an external script and afterwards were also able to demonstrate important aspects of the collaborative behavior without the script. In their contribution, Rummel and Spada point to important conditions that must be met in order for script internalization to take place (e.g., guiding reflection, fading out, motivation).

The chapter by *Runde, Bromme, and Jucks* also emphasizes scripting as a way to support communicating individuals with largely differing knowledge structures, namely medical experts and patients in an online counseling scenario. In contrast to many of the other chapters in the book that use explicit collaboration scripts, Runde et al. focus on the effects of implicit scripting that was realized by external representations shared by the doctor and the patient. In their study, they found evidence for representational guidance by implicit scripting. This evidence was indicated by positive effects of the external representation on the content of the expert-layperson communication.

Nückles, Ertelt, Wittwer, and Renkl draw our attention to a highly promising function of collaboration scripts: Supporting the communication between individuals with large differences in prior knowledge. Nückles et al. investigated the effects of a collaboration script that supported the online communication between laypersons and experts in a computer helpdesk scenario. Findings of their experimental study show that providing the laypersons with successive prompts to better formulate their query substantially improved the effectiveness of communication by yielding the best expert reconstruction of the problem.

In his discussion, *Hesse* points to possible drawbacks of the collaboration script approach. One disadvantage may be that the designer of a collaboration script forces learners who may already possess effective collaboration strategies to adopt a strategy that interferes with their personal, possibly highly functional collaboration approach. As an alternative to the scripting approach, Hesse therefore introduces what he calls the "awareness approach". The aim of this is to provide collaborators with information about the group members, the history of the group, or the group's situation instead of instructions (of varying detail) concerning how to structure their collaboration.

(2) From a *computer science perspective*, the prescription of activities and their sequences is an important issue. In the research areas of computer-supported collaborative work, collaborative learning and Artificial Intelligence (AI), scripts have been used to support developers in defining, configuring and adapting system behavior (such as in Hypercard, 1987). Scripts have also been used to guide users through complex work or learning processes (cf. Haake & Schümmer, 2003; Hoppe, et al., 2000; Hron, Hesse, Cress, & Giovis, 2000; Wessner & Pfister, in press). While approaches such as Workflow Management Systems (http://www.wfmc.org; http://www.e-workflow.org) focus on organizational processes (macro level), process modeling and execution languages (Dowson & Fernström, 1994) focus on supporting detailed work processes (micro level). Important issues involve the representation and computational semantics of scripts. For example, these include how scripts can be efficiently constructed and executed, and how their presentation and interaction mechanisms at the user interface should be designed to facilitate process execution and learning. Connecting macro and micro level approaches is still an open issue.

Ayala addresses the question of how software agents can contribute to scripting collaboration. He identifies a potential for agent-based procedural collaboration support both on the macro-level of collaborative learning (e.g., supporting the formation of appropriate groups) as well as on the micro-level (e.g., by supporting coordination). Illustrating his approach with two examples of agent-based environments, Ayala suggests different approaches for agent-supported collaboration scripts used with domains, which are pedagogically structured than for those, which are not. For both cases, Ayala examines the types of support possible on the macro and micro levels.

Miao, Harrer, Hoeksema, and Hoppe analyze the extent to which IMS LD appropriately addresses crucial aspects of scripting collaborative learning. They identify five major shortcomings of current approaches, e.g., the problem of modeling groups and the complexity of modeling dynamically changing artifacts that are produced and modified by collaborators in the learning groups during runtime. Miao et al. propose a CSCL scripting language aimed at overcoming these issues, e.g., by explicitly introducing the group and the artifact as entities and by extending the space of actions and expressions. By analyzing a typical CSCL script, they exemplify their approach together with the modeling environment they developed.

The chapter by *Lauer and Trahasch* is devoted to facilitating learning from multimedia lecture recordings through annotations and scripted discussions. They introduce the concept of scripted anchored discussions that combine the approaches of scripted and annotation-based discussion. They define scripted anchored discussion as an activity in which several learners exchange structured comments. These comments are connected to certain spatial and temporal positions in digital documents. Lauer and Trahasch propose a formal model using a finite state machine formalism and provide examples to illustrate their approach. Using this model, they propose different strategies for increasing the script's adaptivity by fading components out.

Haake and Pfister suggest that the main function of CSCL scripts is providing support for coordinating learners by constraining their potential activities. They identify the inflexibility of collaboration scripts as a shortcoming of current approaches. These scripts are mostly built-in components of the CSCL environments and cannot be adapted quickly to specific contexts of use. They propose a formal model of CSCL scripts as extended finite state automatons as an important step in the direction of more flexible scripts, which can easily be changed by designers and teachers. For learners in specific roles and states, the script defines what they are allowed to do and what user interface they see. Moreover, the chapter presents a tool based on this model that supports the editing of scripts on varying levels of granularity.

In his comments, *Suthers* distinguishes two roles of computational scripts. Scripts can be a means of decreasing the cognitive load of learners and may help create effective learning situations. Scripts can also be a means of making the design of learning situations more explicit and accessible for discussions among educators and learners. Suthers discusses the ambivalences of computational scripts: they may provide guidance but may also remove "out of context" interaction. Scripts may support successful collaboration episodes, but may also serve as a potential resource for learners.

(3) From an *educational perspective*, scripts are primarily interesting for their potential to improve collaboration processes and individual learning outcomes in formal and informal educational settings like schools, university courses, or museums. Educational approaches are typically based on the constructivist assumption of active learners in a zone of proximal development (Vygotsky, 1978). In this zone of proximal development, learners collaboratively use technological tools and/or participate in a knowledge community. A collaboration script then provides such a zone of proximal development. The script should increasingly be replaced by the individual's self-regulation. Important research questions are: What kinds of activities and roles and which kind of sequencing are beneficial for collaborative learning and should therefore be used in the design of collaboration scripts? How do collaboration scripts compare to other forms of facilitating collaborative learning? How can collaboration scripts be effectively integrated into different computer-supported collaboration scenarios?

Weinberger, Stegmann, Fischer, and Mandl introduce collaboration scripts as an instructional approach for facilitating argumentation in a problem-oriented and distributed learning environment. They analyze the written discourse of distributed groups of students who were supported by different script components. The script components address different dimensions of argumentative knowledge construction (e.g., the epistemic and the social dimension). Their findings show that the script components improved argumentation with respect to the dimension they focused on. Moreover, they identify script components, which – in addition to improving collaboration – facilitate individual transfer from collaboration.

Ertl, Kopp, and Mandl explore the effects of collaboration scripts in videoconference-based tutoring environments. Their scripts specifically aim to support the interaction of learners separated by distance. They report on two experimental studies that consider the effects of such scripts on collaboration and learning outcomes. Their results show that scripts can have rather different effects on collaboration processes and individual outcomes. Their findings further point to the importance of analyzing the effects of scripts in the broader instructional context. Their collaboration scripts that aim to improve interaction proved to be effective for individual learning outcomes only when additional conceptual support was provided in the form of a content scheme.

Kolodner's chapter introduces the Learning by Design approach as a way to help learners acquire and refine (cognitive) scripts for successful participation in science-related discourse practices. Kolodner couples the script concept as presented by Schank and Abelson (1977) and Schank (1999) with stances taken in Lave and Wenger's (1991) conception of communities of practice. The Learning by Design approach focuses on the activity structures (or instructional scripts) that require learners to present the results of their work to the class (poster session, pin-up session, gallery walk). Kolodner explores how these can help to form stable cognitive scripts for participation in scientific practices.

In their discussion of educational approaches to scripting, *Häkkinen and Mäkitalo-Siegl* suggest considering scripts as contextual and situated resources in collaborative learning environments. An educational challenge connected to such a perspective is the integration of CSCL into the classroom. More specifically, they identify a current deficit with respect to theory-based instructional strategies for teachers to better integrate CSCL scripts into the overall classroom activity. They identify a further challenge for future research as exploring how external scripts can be gradually replaced by individual self-regulation. Methodologically, the authors conclude that these challenges can be addressed more appropriately when longer-term follow-up studies are included in research programs.

(4) Interdisciplinary perspectives. Although scripts can be regarded as a boundary concept for CSCL, cognitive psychology, computer science and

educational approaches have just started to collaborate in designing better representations and user interfaces for collaboration scripts. This has been achieved by exploring how the cognitive scripts of collaborators interact with the externally represented scripts provided by different instructional and computational approaches. In this context, two chapters aim to describe perspectives for promising interdisciplinary research on collaboration scripts.

Dillenbourg and Jermann introduce their SWISH model for the design of integrative scripts. They basically suggest splitting the task so that collaborators have to interact in a way that makes learning processes more likely to happen. From this cognitive design rationale, they then head in two directions. First, they describe collaboration scripts as part of a larger didactic activity in the classroom. They provide a systematization of script families that can be specified for different contexts and learning goals, thus connecting their approach to educational theory building. Second, they take a step towards computational approaches in generalizing their scripts and in presenting a generic modeling scheme.

Building on Perkins' (1993) Person-Plus-Surround approach, *Carmien, Kollar, G. Fischer, and F. Fischer* propose a conceptual framework. In their framework, three main components are proposed to describe the complex interplay between internal (cognitive) and external scripts in accomplishing collaboration tasks. These components are the activity, the underlying knowledge, and the executive function. Two script approaches from computer science and from educational psychology are analyzed and compared with the conceptual framework. One represents a script for living (supporting people with cognitive disabilities) and the other represents a script for learning (facilitating argumentation in biology classes in high school).

Stahl's comment highlights an aspect not prominently addressed in both of the approaches of this interdisciplinary section, but one which is of great importance for research on CSCL: the aspect of scripting group cognition. To align theory building on collaboration scripts with current socio-cultural thought, he argues for re-conceptualizing scripts as situated resources rather than prescriptions for acting in collaborative situations.

This volume should be seen as a reference on collaborative learning that brings together scripting approaches from cognitive psychology, computer science, and education. We believe that research on collaboration scripts has an extraordinary potential for advancing the multidisciplinary endeavor of CSCL research. It is our hope that this book can provide a rich basis for further exploring and realizing this potential.

REFERENCES

Bromme, R., Hesse, F. W., & Spada, H. (Eds.) (2005). *Barriers and biases in computer-mediated knowledge communication – and how they may be overcome.* New York, NY: Springer.

Chinn, C. A., & Brewer, W. F. (1993). The role of anomalous data in knowledge acquisition: A theoretical framework and implications for science instruction. *Review of Educational Research, 63*, 1-49.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen: Open Universiteit Nederland.

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

Dowson M., & Fernström, C. (1994). Towards Requirements for Enactment Mechanisms. In B. Warboys (Ed.), *Software Process Technology. Third European Workshop, EWSPT '94, Villard de Lans, France, February 1994. LNCS 772.* Heidelberg: Springer.

Gräsel, C., Fischer, F., Bruhn, J., & Mandl, H. (2001). Let me tell you something you do know. A pilot study on discourse in cooperative learning with computer networks. In H. Jonassen, S. Dijkstra, & D. Sembill (Eds.), *Learning with multimedia – results and perspectives* (pp. 111-137). Frankfurt a. M.: Lang.

Haake, J. M., & Schümmer, T. (2003). Supporting collaborative exercises for distance learning. In B. Wasson, S. Ludvigsen, & U. Hoppe (Eds.), *Designing for Change in Networked Learning Environments* (pp. 125-134). Dordrecht: Kluwer.

Hewitt, J. (2005). Toward an understanding of how threads die in asynchronous computer conferences. *The Journal of the Learning Sciences, 14*(4), 567-589.

Hoppe, H. U., Gassner, K., Mühlenbrock, M., & Tewissen, F. (2000). Distributed visual language environments for cooperation and learning - applications and intelligent support. *Group Decision & Negotiation, 9*, 205-220.

Hron, A., Hesse, F. W., Cress, U., & Giovis, C. (2000). Implicit and explicit dialogue structuring in virtual learning groups. *British Journal of Educational Psychology, 70*, 53-64.

King, A. (1998). Transactive peer tutoring: distributing cognition and metacognition. *Educational Psychology Review, 10*, 57-74.

Kollar, I., Fischer, F., & Hesse, F. W. (in press). Collaboration scripts – a conceptual analysis. *Educational Psychology Review.*

Kollar, I., Fischer, F., & Slotta, J. D. (2005). Internal and external collaboration scripts in web-based science learning at schools. In T. Koschman, D. Suthers, & T.W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years!* (pp. 331-340). Mahwah, NJ: Lawrence Erlbaum Associates.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambrigde, UK: Cambrige University Press.

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

Perkins, D. N. (1993). Person-plus: a distributed view of thinking and learning. In G. Salomon (Ed.), *Distributed cognitions: psychological and educational considerations* (pp. 88-110). Cambridge, UK: Cambridge University Press.

Sandoval, W. A., & Millwood, K. A. (2005). The quality of students' use of evidence in written scientific explanations. *Cognition and Instruction, 23*(1), 23-55.

Schank, R. C. (1999). *Dynamic Memory Revisited.* New York: Cambridge University Press.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding.* Hillsdale, NJ: Erlbaum.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.* Cambridge, MA: Harvard University Press.

Webb, N. M., & Palincsar, A. S. (1996). Group processes in the classroom. In D. C. Berliner & R. C. Calfee (Eds.), *Handbook of Educational Psychology* (pp.841-873). New York, NY: Simon and Schuster MacMillan.

Weinberger, A. (2003). *Scripts for computer-supported collaborative learning. Effects of social and epistemic cooperation scripts on collaborative construction.* Doctoral dissertation, Ludwig-Maximilians-University, Munich, Germany. Available at: http://edoc. ub.uni-muenchen.de/archive/00001120/01/Weinberger_Armin.pdf.

Wessner, M., & Pfister, H.-R. (in press). Points of cooperation: Integrating cooperative learning into web-based courses. In H. U. Hoppe, M. Ikeda, & H. Ogata (Eds.), *New technologies for collaborative learning.* Dordrecht: Kluwer.

PART I

COGNITIVE PERSPECTIVES

Chapter 2

SCRIPTING COLLABORATIVE LEARNING PROCESSES: A COGNITIVE PERSPECTIVE

Alison King
California State University San Marcos

Abstract: Scripting collaborative learning is an effective approach to promoting learning in both face-to-face and on-line computer learning contexts. Although the term *script* originated in cognitive psychology, it is used in educational contexts to describe ways of structuring interaction and scaffolding collaborative learning through the use of roles, activities, and sequencing of activities. There are several specific types of learning activities that numerous lines of research have shown enhance learning during interaction, however, these activities rarely occur spontaneously during naturally-occurring group collaboration. Also, it is not always clear what individuals learn during collaboration, how they learn it, and the underlying cognitive mechanisms that account for learning collaboratively. Four illustrative approaches to scripting face-to-face collaboration are presented. Each approach is examined to reveal how roles, activities, and sequence of activities, are used to structure collaborative learning and what particular cognitive, metacognitive, and socio-cognitive processes their scripts are intended to induce in learners. The expectation for some scripts is that over time learners will internalize the roles, activities, and sequence; and, once learners can play all of the roles of a script on their own, they will self-regulate their learning without the aid of an external script. However, the wide range of differences in both the complexity and goals of scripts affects their potential for internalization, and some external scripts are not intended to be discontinued even if roles are internalized.

A large body of research has shown that collaborative approaches to learning can be effective in producing achievement gains, promoting critical thinking, and enhancing problem solving in both face-to-face learning contexts (e.g., Cobb, 1988; King, 1989; Webb, 1989; Webb & Palincsar, 1996) and more recently in computer-supported learning environments (e.g., Weinberger, Fischer, & Mandl, 2002).

From a cognitive perspective, learning is defined as cognitive change or conceptual change; that is, some form of reorganization and reconstruction of the learner's own knowledge. This change occurs as connections are made between the new material and prior knowledge and are integrated into the learner's existing knowledge base. From a *socio*-cognitive perspective (e.g., Mugny & Doise, 1978, Vygotsky, 1978), these cognitive changes are strongly influenced by interaction and activity with others.

Any interaction with another provides opportunities for learning to occur; however, some forms of interaction and activity have been found to be more effective in facilitating learning than others. For example, giving explanations is more effective than receiving them (Webb, 1989). And helping behavior that supports others' problem solving by offering cues and hints that guide them to achieve a solution on their own is more effective in promoting learning than helping by simply providing the right solution. Moreover, it appears that different levels of *verbal* interaction promote different kinds of learning (e.g., Chan, Burtis, Scardamalia, & Bereiter, 1992; King, 1994; Webb & Palincsar, 1996) and are therefore conducive to different kinds of learning tasks. For example, factual questioning and responding tend to be effective for knowledge retelling tasks because fact questions tend to elicit facts. However, fact questions are less effective for complex learning tasks which involve analyzing and integrating ideas, constructing new knowledge, and solving ill-structured problems, as they seldom elicit the required thoughtful responses (Cohen, 1994; King, 1994).

Unfortunately it is rare for collaborating learners to spontaneously use effective interaction procedures and match them to the task at hand without some form of explicit prompting or other guidance (Bell, 2004; Britton, Van Dusen, Glynn, & Hemphill, 1990; Cohen, 1994; King, 1994; King & Rosenshine, 1993; Kuhn, 1991). Indeed, even when given instructions to work collaboratively on a task, learners generally tend to interact with each other at a very basic level (Vedder, 1985; Webb, Ender, & Lewis, 1986) and do not even consistently activate and use their relevant prior knowledge (see Pressley, McDaniel, Turnure, Wood, & Ahmad, 1987). For this reason, numerous attempts have been made to promote learning by structuring and regulating the interaction within collaborating groups so that learners are required to interact in ways that induce cognitive processes appropriate to their learning task. Such structures compel learners to assume designated roles, follow a prescribed sequence of activities, and sometimes even engage in a particular pattern of dialogue (e.g., Dansereau, 1988; King, 1997; Palincsar & Brown, 1984; Pressley, Symons, McDaniel, Snyder, & Turnure, 1988; Webb & Farivar, 1994).

These methods of structuring interaction have generally been referred to variously as "scaffolding learning", "prompting thinking", "using problem solving supports", "guiding cognitive performance", and "strategy instruction". However, recently the term *scripting collaboration* has appeared in the literature to describe these and other ways of structuring and regulating interaction during collaborative learning. Even more recently, *scripting collaboration* has been used to describe computer-supported collaborative learning (CSCL) environments (e.g., Weinberger, et al., 2002) where collaboration is partly or totally mediated by computer (see also, Lauer & Trahasch, this volume; and Ertl, Kopp, & Mandl, this volume). Thus, *scripted collaboration* is a term currently used to refer to externally structured collaborative learning in both on-line and face-to-face learning contexts.

This chapter presents a cognitive perspective on scripting collaborative learning. The following section presents different meanings of the term *script* as it is used in cognitive psychology and in collaborative learning contexts. The section after that deals with specific cognitively-oriented activities that several lines of research have shown enhance learning during interaction. Next, four illustrative scripted collaboration approaches that use some or all of these activities are examined to reveal the cognitive, metacognitive, and socio-cognitive processes their scripts are designed to induce in learners; the use each script makes of roles, activities, and sequence of activities in structuring collaborative learning is also analyzed. The issue of what individuals learn during collaboration and how they learn it is a thread that runs throughout the chapter. A final discussion of the potential for learners to self-regulate their collaborative learning revolves around differences in approaches to scripting collaboration and the related question of when (or even if) use of a script can be discontinued once roles and scripts are internalized.

1. SCRIPTS AND SCRIPTING

1.1 Scripts in cognitive psychology

According to Schank and Abelson's (1977) seminal work on the topic, a script is an internal memory structure of a "sequence of actions that define a well-known situation" (p. 41) where there is a socially shared understanding of the roles and procedures to be followed (for example, in the frequently cited "going to a restaurant" script: getting seated, looking at the menu, ordering food, eating, and then paying). Thus, a script is a guide to the roles and steps people follow for what to do and how to do it in a specific social situation. An individual develops a particular script from repeated participation in several specific instances of a social situation and by abstracting common features from those instances (Schank & Abelson, 1977). Simply put, a script involves a sequence of actions where each actor has a specific part to play and pre-specified actions to take, somewhat like the script of a play where action and stage directions are prescribed by the playwright. Once stored in memory, a script can be activated when cued by a similar situation and can guide the individual in how to act in that situation.

There are several ways in which scripts facilitate information processing. Because a script involves expectations about the order as well as the occurrence of events, having a script for a situation can help an individual to understand that particular situation, remember procedures to be followed, and predict roles and actions of those involved (Schank & Abelson, 1977). Furthermore, scripts play a useful role in reducing cognitive load for individuals so they can focus their attention on what is important in an interaction and its context (Dansereau, 1988); with *procedure* scripted, attention can be focused on *content* of an interaction.

1.2 Scripts and scripting in educational contexts

Although in cognitive psychology the term "script" refers to the Schank and Abelson definition, the term *scripting* has also begun to be used in educational settings (particularly in computer-supported learning), where the meaning it has taken on is somewhat different. In contrast to the Schank and Abelson (1977) view of script as a fairly static internal memory structure with a narrowly constrained set of actions and roles, researchers in educational psychology talk about *scripting* the interaction of learning groups (Dansereau, 1988). In this context, scripting is used more broadly to describe how collaborative learning can be externally structured or scaffolded for the purpose of prompting group interaction that promotes learning. Scripting of the interaction during collaboration is designed so that the roles of participants, actions engaged in, and the sequence of events, prompt specific cognitive, socio-cognitive, and metacognitive processes, thus ensuring that the intended learning takes place.

Whereas in the Schank and Abelson cognitive psychology view, a *script* is an internal memory structure with a narrow application, in the educational view scripts are externally imposed, are more flexible, and have broader application. They also differ in terms of their location, their point of origin, and their purpose.

The purpose of a script in cognitive psychology is to guide the individual in the social roles and actions expected in a specific social situation; whereas in educational settings a script's purpose is to prompt collaborating learners to focus on, remain engaged in, and regulate specific roles and actions which are expected to promote learning. While both kinds of scripts emphasize roles and actions to be taken, those roles and actions originate from different sources and are created by different agents. In cognitive psychology a script is seen as a memory structure, residing internally to the individual but created by that individual by means of abstracting the essence of a social situation from repeated external experiences. In contrast, in educational settings a script is designed externally by *others* and explicitly imposed on learners (by a teacher or other learning facilitator) as a guiding structure to prompt them in how to act. Initially the script is external to the individual but the expectation is that, over time, it will become internalized through practice (e.g., in the Vygotskian, 1978, sense) and the timely fading of external prompts.

Thus, the term *internal* collaboration script often refers to an internalized version of an external script; of course, at the same time, it may also refer to prior socially/culturally-derived rules for cooperating as in the Schank and Abelson "script" (see also Carmien, et al., this volume). For example, every learner by the age of three has already developed an internal Schank and Abelson kind of cooperation script (perhaps only a rudimentary one that specifies roles such as turn-taking and rules such as sharing).

2. COGNITIVE, METACOGNITIVE, AND SOCIO-COGNITIVE ASPECTS OF LEARNING THROUGH INTERACTION

Cognitive processes of thinking and learning take place within the individual, as do metcognitive processes (monitoring, regulating and evaluating one's own thinking and learning). In contrast, according to theories of the social construction of knowledge (e.g., Bearison, 1982; Damon, 1983; Mugny & Doise, 1978; Perret-Clermont, 1980; Vygotsky, 1978), socio-cognitive processes are induced by joint activity where learners scaffold their collaborative thinking and learning in a shared construction of knowledge. As such, partners' actions are interdependent, each triggering the other's cognitive and metacognitive processes; in such mutual cognition learners contribute jointly to development of the learning outcomes. By necessity then, socio-cognitive processes always arise in some kind of social context, real or virtual. In this view, cognitive and metacognitive processes are individual cognitions (occurring internally "in the head" of the individual) while socio-cognitive processes are social (occurring outside the individuals – in the interaction per se). Cognitive, metacognitive, and socio-cognitive processes involve thinking; whereas cognitively-oriented *activities* are experiences, behaviors, and interactions, that often (but not always) induce cognitive, metacognitive, and socio-cognitive processes in the learners engaged in those activities.

Socio-cognitive, cognitive, and metacognitive processes come together during collaborative learning. Ideally, during interaction and activity individual learners are continually using each other's ideas, reasoning, explanations, and argumentation to modify their own thinking and restructure their own knowledge (individual cognitive processes). At the same time they are jointly constructing knowledge and negotiating meaning with each other (socio-cognitive processes). The products of that socio-cognitive process (the jointly-constructed knowledge and meanings) are (to a greater or lesser extent) internalized by both learners individually; and the procedures, skills, and strategies used are also internalized by both (Rogoff, 1990). For example, when summarizing occurs during collaboration it usually is in response to another's question or in the context of engaging in a pre-specified role (e.g., the summarizer role) of a collaboration script; in such a case, where the summary is created jointly by learners mutually building on each other's contributions in a coordinated interdependent effort, summarization is a socio-cognitive process. The learning product is the jointly-constructed summary; and when it is internalized, that summary, being far more coherent and complete than it would be if developed by each learner alone, can result in a richer knowledge base for both learners. Ideally, each learner's summarizing skills (identifying main idea, selecting and sequencing details, etc.) are enhanced because of the other's contributions; and these summarizing skills are also internalized to be applied in similar learning situations in the future. Thus both the new knowledge constructed during a collaboration and the cognitive skills which individuals learn, refine, and use during that collaboration are what is retained by the individual learner after the collaboration.

Metacognitive processes can play a major role in collaborative learning as learners mutually regulate their joint learning. Activities of monitoring and regulating learning during collaboration can induce corresponding metacognitive processes in individual learners. Again, cognitive and metacognitive processes are always individual, while socio-cognitive processes are induced in interaction with others.

Generally the term collaborative learning means that learners are engaged in activities that are intended to induce socio-cognitive processes. This meaning implies an important distinction between collaborative and cooperative learning. Cooperative learning often involves separate activities by individuals through the distribution of labor or task components, with little of the joint activity that induces socio-cognitive processes so characteristic of true collaborative learning.

2.1 Effective learning activities

Effective learning activities are ones that induce relevant cognitive, metacognitive and socio-cognitive processes in participants. Although repetition, rehearsal and retelling are effective activities for memorizing factual material, and summarizing and paraphrasing are effective for promoting understanding and demonstrating comprehension, research has revealed that when complex learning occurs during interaction it can be attributed primarily to activities that go beyond memorization and comprehension. Effective learning interactions induce complex cognitive processes such as analytical thinking, integration of ideas and reasoning. Activities that have been found to promote such higher-level cognitive processes include: elaborating on content (e.g., Webb, 1989); explaining ideas and concepts (e.g., Chi, deLeeuw, Chiu, & LaVancher, 1994); asking thought-provoking questions (e.g., King, 1994); argumentation (e.g., Kuhn, 1991); resolving conceptual discrepancies (e.g., Piaget, 1985) and modeling of cognition. Although these activities are learned and refined during interaction with others (Leont'ev, 1932; Luria, 1928; Vygotsky, 1978), they can be accomplished by an individual learner alone (see, for example, Chi's & VanLehn's, 1991, "self-explanation" and King's, 1989, "self-questioning") as well as in a social context such as a collaborative learning group.

Recently researchers have designed various collaborative learning approaches that structure or "script" group interaction so as to elicit and regulate these specific learning activities in the expectation that they will, in turn, induce high-level cognitive, metacognitive, and socio-cognitive processes in learners. The phenomenon of making thinking explicit through "thinking aloud" during interaction sets the stage for such higher-level learning to occur.

Thinking Aloud. Talking or writing about the task at hand is known as thinking aloud. The advantage of thinking aloud during collaboration is that it makes thinking explicit and available to the individual doing the thinking and also exposes that same thinking to the rest of the group. Such verbalization during collaboration promotes learning in and of itself because it forces those who are "thinking aloud" to clarify their own ideas, elaborate on them, evaluate their existing knowledge for accuracy and gaps, or in some other manner re-conceptualize the material (Bargh & Schul, 1980; Brown & Campione, 1986). As importantly, making thinking explicit allows others access to that thinking too; they may then respond by challenging, disagreeing, asking for proof, offering examples and other elaboration, justifying and so on. Making thinking explicit by thinking aloud in a group is a general phenomenon that can give rise to the powerful learning activities of explanation, questioning, elaboration, argumentation, resolution of conceptual discrepancies, and modeling of cognition.

Explaining. An effective explanation goes far beyond description; it tells the "why" and "how" about whatever is being explained (King, 1997), rather than just describing it (telling the "what" of it). Explaining must be in the learner's own words rather than simply repetition of already-memorized material (King, 1997) because accurate paraphrasing is an indication that the explainer understands. In addition to demonstrating true understanding, a useful explanation requires analytical thinking, as the explainer must make connections between the phenomenon being explained and prior knowledge.

Explaining something to someone else often requires the explainer to think about and present the material in new ways such as: relating it to the other's knowledge or experience, translating it into terms familiar to the other, or generating new examples. Thus, explaining expands understanding for the individual doing the explaining because it forces the explainer to clarify concepts and generally reorganize knowledge structures (Chi & VanLehn, 1991; Chi, et al., 1994; Vygotsky, 1978; Webb, 1989; Webb & Farivar, 1994). Receiving explanations often enhances learning also (Webb, 1989).

Two separate lines of research by Chi and Webb and their colleagues have shown the power of self-explanation in promoting learning for individuals (see the "self-explanation" effect, Chi & VanLehn, 1991). However, during collaborative learning, when one partner's explanation is enlarged upon or clarified by others and a fuller explanation is jointly constructed, explaining becomes a collaborative activity that induces socio-cognitive processes.

Asking thought-provoking questions. Factual questions and comprehension questions are important in learning contexts as their responses help determine whether certain information has been acquired (for factual questions) and the extent of understanding achieved (for comprehension questions). Both of these types of question are memory-based and require little cognitive effort; both ask for the recall of information from memory and the reproduction of that information, either verbatim retelling of it (for fact questions) or a reconstructed version that is paraphrased to show understanding (for comprehension questions). However, for inducing higher-level cognitive processes, asking questions that are thought-provoking is much more effective.

Thought-provoking questions require thinking. They ask learners to go beyond exact reproduction of material or reconstruction of it, to actually thinking *with* that material and *about* that material, making connections between elements of the material and between that material and what is already known. Thought-provoking questions call for higher-level cognitive processes such as integrating ideas into newly constructed knowledge to make inferences, generalizations, speculations, justifications, applications, alternative perspectives, problem solutions, and the like. In a collaborative learning context, thought-provoking questioning and the comparably thoughtful responses those questions elicit can be a valuable learning activity, and results of several programs of research have confirmed that asking and answering thought-provoking questions promotes high-level learning (e.g., Graesser, 1992; King, 1989, 1994; Lepper, Aspinwall, Mumme, & Chabey, 1990; Pressley, et al., 1992).

Simply posing thought-provoking questions on one's own is an activity that triggers higher-level cognitive processes in individuals (see King's, 1989, "self-questioning"). In generating such questions learners must identify the main ideas and think about how those ideas relate to each other and to the learners' own prior knowledge and experience. According to theories of information-processing, thinking about material in these ways establishes complex cognitive networks connecting the new ideas together and linking them to what the learner already knows. Such extensive cognitive representations of the material are more memorable.

Elaborating. Elaborating on an issue, topic, or idea involves adding details, giving examples, generating images, and in general relating the new material to what is already known. These elaborations are incorporated into learners' existing knowledge; and, as a consequence, their mental representations are reorganized and increased in complexity, thus improving understanding and recall (Dansereau, 1988; Webb & Farivar, 1994). A number of research programs have demonstrated the effectiveness of elaboration as a method for learning new material (e.g., O'Donnell & Dansereau, 1992; Pressley et al., 1987; Webb, 1989).

Explaining, questioning, and elaboration are activities that benefit an individual learner even without another's involvement. However, in a group-learning context these activities (and the cognitive processes they induce) are more likely to occur because they are triggered by others during interaction.

Argumentation. Reasoned argument involves giving adequate and convincing evidence or reasoning to support one's claims, statements and other assertions (Kuhn, Shaw, & Felton, 1997). Although one primary purpose of argument is to convince others of a belief or claim, argumentation can also be used to explore an issue and arrive at a deeper understanding of that issue (Wright, 1995). During collaboration when a learner makes an assertion, such as a conclusion arrived at, a statement of cause and effect, a hypothesis to account for some phenomenon, an explanation, a theory of how things are, that assertion elicits evidence-based thinking in others; that is, they ask for evidence or reasoning that supports the assertion (Kuhn, et al., 1997). Any collaborative activity provides a context for learners to develop and practice argumentation skills because it offers opportunities for them to generate, compare and evaluate multiple conclusions, theories, counter theories, counter arguments, and rebuttals along with any supporting evidence provided. In effect, during this verbal interaction, learners are not just exchanging theories and rebuttals, they are often negotiating meaning and arriving at re-conceptualized and deeper understanding about the topic or issue being argued. These jointly constructed meanings can be internalized by individuals as their own revised mental representations of the topic or issue. Engaging in such constructive argumentation usually promotes learning (Kuhn, et al., 1997). Like explanation, elaboration, and questioning, argumentation is a learning activity that can occur independently of others. In fact, individual deliberation about an issue often takes the form of an internal argument where the individual considers all sides of the issue – all possible challenges, counterarguments, justifications, and refutations. Thus, whether used by an individual or in interaction with others, argumentation can aid in clarifying thinking and promoting understanding.

Unfortunately, without specific prompting and scaffolding, even adults rarely engage in reasoned argumentation; for example, most adults have been found to make assertions and, even when prompted, are unable to support them with evidence or logical reasoning (Kuhn, 1991). Interaction during collaborative learning can be structured to guide and support learners' reasoned argumentation during complex learning tasks. Research (e.g., Hogan, Nastasi, & Pressley, 1999) shows that such activity can promote learners' skills of developing sound arguments and detecting faulty ones.

Reconciling cognitive discrepancies. During group interaction, differences between individuals' opinions and understandings about a topic are exposed. Individuals discover that their own understanding of an aspect of the content, their opinions about an issue, or even their basic background information about the material may not be shared by others in the group and may even differ to a great extent from others. When individuals are confronted with these conceptual discrepancies, they experience cognitive conflict within themselves (see Piaget, 1985; De Lisi & Golbeck, 1999) and they might feel the need to resolve it through further interaction with others.

Reconciling cognitive discrepancies can give rise to a number of other cognitive, metacognitive, and socio-cognitive processes. Individual group members may find they must clearly articulate their own position, explain their ideas, defend their views, verbalize their confusions, acknowledge gaps in knowledge, recognize any misconceptions, and generally present their thoughts in a reasoned manner. Other group members may do the same as the group experiences *socio-cognitive* conflict (Mugny & Doise, 1978). In attempting to understand each other's ideas and views and reconcile them with their own in this way, group members arrive at shared meaning (Roschelle, 1992). Thus, ideally, in the process of resolving those cognitive discrepancies, knowledge is jointly constructed, and the product of group interaction, the new jointly constructed knowledge, is individually internalized.

Modeling of cognition. A general phenomenon of learning through interaction is social modeling of cognition and metacognition. In collaborative learning contexts, when skilled peers demonstrate accurate use of questioning, explaining, and elaborating, they become ideal models for others to observe and imitate. When individual learners observe and imitate their peers' use of cognitive skills, they modify and refine their own use of those skills. They can even learn new cognitive strategies by modeling their own reasoning, argumentation style, questioning and problem-solving strategies on those of other group members. Similarly, sharing questions and responses with each other can help the group develop ideal standards of expert questioning and responding.

Although peer modeling of cognition is generally not intentional, it is a very powerful way of learning during interaction. However, individual cognition can be modeled during interaction only if it is exposed; and this is where thinking aloud contributes to modeling by making thinking explicit and available to all. Of course, before any of these higher-level cognitions can be modeled during interaction, they first have to occur, either spontaneously or through some form of prompting.

2.2 The need for structuring interaction

There may be several explanations for why learners generally do not interact in cognitively effective ways without some structured guidance. Learners may not know what it actually means to explain and argue and analyze ideas, they may not have been taught how to do so; or they may not be well practiced in the skills of explanation, argumentation, analysis and other aspects of high-level discourse in a collaborative setting. Perhaps for some learners, their internal scripts for collaborative learning (the script they have built up from their experiences in groups) may be limited to such cooperative action as taking turns, dividing labor, and getting the task completed. For learners with such a naïve conception of what constitutes group collaboration, their most frequently occurring verbal interaction may be no more than simply sharing information and checking for consensus.

Because giving explanations, asking thoughtful questions, elaborating on content, argumentation, and engaging in exposing and reconciling cognitive discrepancies are known to be effective in collaborative learning but do not generally occur spontaneously, scripted collaboration approaches focus on structuring group interaction so as to elicit these and other kinds of effective activity. Many of these scripting approaches also prompt the metacognitive processes needed to monitor and regulate those activities.

In the following section several examples of face-to-face scripted collaboration approaches are presented. Each script is analyzed to reveal the cognitively effective activities their scripts support and the cognitive, metacognitive and socio-cognitive processes those activities are intended to induce in participating learners.

3. EXAMPLES OF FACE-TO-FACE SCRIPTED COLLABORATION

Collaboration scripts run along a continuum from very basic to very sophisticated; and scripts can be designed for many kinds of learning tasks and objectives from ordinary factual learning, to text-based comprehension, to higher-level learning that involves knowledge building and problem solving. The scripted collaboration approaches presented here represent a variety of approaches to scripting and show how scripting can be used for a range of tasks from simple knowledge retelling to complex problem solving.

All of the scripts presented here are domain independent, and therefore all can be used with a variety of subject areas for learning from a range of materials. Because each script is designed to match a particular kind of learning task, some scripts focus more on inducing socio-cognitive processes than others; also, some scripts structure the task and its sequence, while others also explicitly scaffold group communication.

3.1 Scripted Cooperation

One of the earliest and simplest approaches to scripting collaboration in educational contexts is Pair Summarizing (Vaughan & Estes, 1986) in which one partner summarizes material read and the other checks for errors and omissions. Pair Summarizing is commonly used to promote recall and understanding of definitions, procedures, and similar conceptual material (see also U. S. Department of Education, 1986). A somewhat more sophisticated version of Pair Summarizing, and the first use of the term *scripting* in an educational context is Dansereau's Scripted Cooperation (Dansereau, 1988; Larson, Dansereau, Goetz, & Young, 1985; Spurlin, Dansereau, Larson, & Brooks, 1984). In this approach to collaborative learning, partners each have specific roles and activities to carry out and the script is used to direct the performance and sequence of those roles and activities. The script consists of the roles of *recaller* and *listener* (cf. the listener and explainer roles in Johnson & Johnson's, 1993, Academic Controversy) and a specific sequence of activity (summarizing, feedback, and joint elaboration, usually followed by exchanging roles for the next portion of content). First, both partners read material (or listen to a lecture) and take notes; then the one designated as recaller summarizes the main ideas of the material orally while the partner listens and checks for errors and omissions (using the notes if needed). When the recaller has finished summarizing, the listener provides feedback on errors, distortions, and material omitted. Then both partners together elaborate on the material read by adding details, generating examples, developing images, and in general relating the new material to what they already know.

The sequence of summarizing, error detection and feedback, then elaboration, is repeated on the next section of text with partners alternating the listener and recaller roles.

Cognitive, metacognitive and socio-cognitive processes underlying Scripted Cooperation. Based on an extensive program of research, Dansereau and his colleagues (Dansereau, 1988; Larson, et al., 1985; Spurlin, et al. 1984) have found this strategy to be effective in enhancing learning for both the recaller and listener. Learning in Scripted Cooperation can be accounted for by the cognitive and socio-cognitive processes induced by the script. First, summarizing helps the summarizer to reformulate and consolidate material into the memory structure already developed during the initial reading of the material. This makes the information more stable in memory and therefore more readily recalled. Summarizing also involves metacognitive processes; during summarizing the recaller is monitoring or self-checking on how well the material is understood. For the recaller, inability to summarize signals a lack of understanding, and errors or omissions suggest inaccurate or incomplete comprehension. During error detection the listener also engages in metacognitive processes to constantly compare what is being orally recalled to the actual content of the material read. The listener's role of monitoring the accuracy of the other's recall also provides the listener with an additional pass through the material thus promoting further consolidation of the new material into memory structures and facilitating future recall. Both partners get further exposure to the material during feedback.

The activities of note-taking and elaboration induce additional cognitive, metacognitive and socio-cognitive processes. Note-taking provides not only another opportunity for further encoding of the material (see Kiewra, 1989), but also for encoding it in a different mode – writing. As a result, both recaller and listener encode the material through reading it in the initial text format, writing notes on it, and hearing it in the oral summary. Encoding through three different modes can result in a richer memory structure with numerous cues for subsequent recall. Through elaboration, the recaller and listener create additional and varied links to their existing knowledge. Elaboration not only extends understanding by adding additional links, it also provides a variety of *different* recall cues (the details added, examples generated, and images developed). The elaboration phase of Scripted Cooperation induces socio-cognitive processes as the elaboration is jointly accomplished by the partners; and products of that elaboration (the details, examples, and images generated) are available for encoding by both. Neither partner alone would generate the same details, examples, and images as they do by engaging in joint elaboration. Such thoroughly encoded and jointly-elaborated material will not soon be forgotten.

This whole procedure, with its set of roles and activities, benefits both partners as neither partner would accomplish the same level of learning without the other's assistance. Because this procedure is intended to support only knowledge acquisition (not higher-level learning), the focus on information processing and encoding activity is both appropriate to the task as well as effective.

Dansereau (1988) found that the partner who summarizes the content presented learns more than the partner who listens and checks for errors. Possibly self-checking of understanding during summarization made the difference; presumably such metacognitive processes enhance learning over and above the cognitive and socio-cognitive processes.

Dansereau (1988) points out that modeling can enhance learning in Scripted Cooperation also. Partners have the opportunity to improve their cognitive skills of summarizing, error detection, and elaboration through observation and imitation of each other's behavior.

3.2 Reciprocal Teaching

Palincsar and Brown's (1984) Reciprocal Teaching is another form of scripted collaboration designed to enhance text comprehension. In this approach learners in small groups take turns assuming roles (questioner, summarizer, clarifier, predictor) and follow a sequence of activity beginning with making predictions about the content and topic of a text segment to be read, reading the segment, asking questions about the content, summarizing and clarifying the content, followed by making new predictions about the next segment of text (Palincsar & Herrenkohl, 1999). This sequence is repeated with additional passages until the complete text is covered. Over time, during subsequent sessions of Reciprocal Teaching, learners get practice in all four activities by assuming different roles during subsequent reading sessions.

A great deal of research on the use of Reciprocal Teaching has been conducted with somewhat mixed results. Although pre-post achievement measures generally show learning gains in text comprehension, it is not always clear that those gains can be attributed to the roles students play rather than simply to the additional processing of material (Rosenshine & Meister, 1994). Rosenshine and Meister suggested that often there was merely rote application of the procedure, as many groups tended to be more interested in following the roles and rules in a routine manner and getting the task done than in fully comprehending the text.

Cognitive, metacognitive and socio-cognitive processes underlying Reciprocal Teaching. Because the roles in Reciprocal Teaching are not clearly defined, their underlying cognitive processes will vary depending on how the roles are taught and modeled by teachers. Each of the roles of questioner, summarizer, clarifier, predictor, has the potential to prompt different cognitive and metacognitive processes in readers. The extent of cognition involved in the questioner role is dependent on the kinds of questions asked. If higher-level questions are asked, then higher-level cognitive processes (such as analysis, inferencing, and making connections to prior knowledge) may be activated; however, since Reciprocal Teaching is designed to promote understanding and remembering of text, the kind of questions asked are generally factual and comprehension ones. The cognitive processes induced by asking fact questions are simple retrieval, while asking comprehension questions may involve some extent of reformulation of content retrieved. Moreover, when those questions are answered, the cognitive processes activated in the responder are likely to be straight retrieval and retelling of content from memory (for both kinds of question); although comprehension questions might induce more extensive reconstruction processes, verbalized as paraphrased or summarized content.

Playing the role of clarifier can trigger several cognitive and metacognitive processes. Clarifiers must constantly monitor their own understanding by comparing what they know with what is being asked and stated by others in the group; this self-checking involves continuous revision of clarifiers' mental representations of the text passage, which results in richer memory structures with a variety of cues for recall.

As in Scripted Cooperation, the summarizer role induces reconstruction of material read and consolidation of it in memory. Here too, summarizing involves metacognitive processes of self-monitoring for comprehension of the material read.

The predictor is a role that can activate higher-level cognitive processes of analysis and reasoning to generate real predictions (as opposed to guesses) about what will happen next. Metacognition comes into play here as the predictor must self-monitor comprehension to avoid making improbable predications.

The mutual regulation of learning that occurs during Reciprocal Teaching is the activity at the heart of this procedure. Mutual monitoring is built in to the use of the roles in conjunction with each other. As questions are asked and answered, as material is clarified and summarized, learning is monitored for accuracy; as predictions are made they are evaluated for consistency with text events. In this way the whole group monitors their on-going comprehension.

The scripted collaboration procedures discussed so far focus on scripting the task and its sequence; however, some attempts have been made to guide both the task sequence and the content of group communication. In these kinds of scripted collaboration the interaction in collaborating groups is structured by guiding the actual dialogue learners engage in during the task. The rationale for this approach is: if explaining, questioning, elaborating, arguing, and reconciling cognitive discrepancies are such effective cognitive and socio-cognitive learning activities, why not use actual dialogue prompts to elicit these particular forms of discourse? Dialogue prompts used to guide the interaction of the group would presumably result in socio-cognitive processes conducive to higher-level learning. For example, when King (e.g., 1989, 1991, 1994; King & Rosenshine, 1993) trained students in collaborative learning groups to ask each other task-related thought-provoking questions, she found that, as expected, those questions elicited explanations, inferences, speculations, hypotheses, comparisons, analyses, conclusions, and other high-level responses. This high level of discourse, in turn, had a direct positive effect on learning. In effect, guiding group discourse in such ways can be a means of controlling discussion content or of keeping discussion focused on a particular procedure or at a high cognitive level.

Two of King's scripted collaborative learning procedures that guide both the task sequence and group communication are presented below. Both procedures provide structured scaffolding for group discourse. The first one guides the discourse of partners during problem solving; the other one prompts partners to initiate, maintain, and regulate high-level discourse during complex collaborative learning tasks.

3.3 Guided Strategic Problem Solving

King's (1991) Guided Strategic Problem Solving (GSPS) procedure was designed to scaffold student interaction when solving complex problems[1]. GSPS is based on a sequence of "strategic" questions that guide learners' problem-solving activity by controlling the content of their interaction while solving problems together. The questions are designed to guide students to be strategic (intentional and planful – rather than resorting to guessing and trial-and-error) during their problem solving.

Learners in small groups or pairs engage in asking and answering these questions with each other to prompt their partners and themselves to plan, monitor, and evaluate their problem solving process and problem solution in a strategic manner. There are no specified roles, and either partner can ask or answer the questions, specific activities are prompted by the strategic questions, and there is a general sequence to. Both the format of the particular questions and the sequence of questions is structured to guide learners through the typical stages of problem solving (e.g., problem-identification and representation, search for a solution path, implementation of a solution,

[1] Complex problems are problems that are ill-structured and/or have several possible solutions.

and evaluation; Gick, 1986) and help them to monitor their progress towards solution. The general strategic questions are designed to prompt learners to clarify the problem, think about the problem in new ways, access their existing knowledge and strategies, formulate plans and strategies for solving the problem, and evaluate alternatives. Examples of the general strategic questions include: "What do we know about the problem so far?", and "Do we need a different strategy?" In addition to being trained to use the questions, learners are coached in developing elaborated responses (ones that are solution-oriented or strategy-oriented) during problem solving. King (1991) found that GSPS was very effective in promoting problem-solving success, for fifth graders in terms of both the problem-solving process and solutions achieved.

Cognitive, metacognitive and socio-cognitive processes underlying GSPS. The learning effects of GSPS can be accounted for by the cognitive, metacognitive, and socio-cognitive processes induced by elements of the collaboration script used. First of all, the question sequence was provided on a hand-held card and this alone assisted learners' information-processing in two ways: it reduced their cognitive load (they didn't have to remember the questions or general problem-solving sequence) as some of the cognitive load was distributed to the question prompt card (see Salomon's, 1997, concept of distributed cognition); and the content of those problem solving questions (as well as the sequence) kept attention focused on the problem space and its solution.

Furthermore, the GSPS script prompted the activities of questioning, explanation, elaboration, and resolution of cognitive discrepancies, which in turn induced cognitive and socio-cognitive processes conducive to problem-solving success. Analysis of transcripts of recorded GSPS student interaction during problem solving revealed that, as expected, when students asked strategic questions, those questions elicited explanations, elaborated responses (e.g., detailed directions for how to execute a specific move, analysis of a situation, and rationales for actions suggested), and follow-up thought-provoking questions relevant to the problem (King, 1991, 1999). Engaging in this questioning-answering dialogue during problem solving allowed students to share information and perspectives, negotiate understanding, resolve cognitive discrepancies, and truly co-construct their problem solving plans, strategies, solution paths, as well as improve their pair's problem-solving performance.

In GSPS, metacognition is overtly built into the script. The general strategic questions are arranged into three categories: plan, monitor, evaluate; this structure in and of itself induces and supports the kind of metacognitive processes that promote success during problem solving. Also, the specific strategic questions students asked each other (e.g., reason for why an attempt was successful or unsuccessful) prompted self-monitoring and regulation of their particular problem solving process and decision-making during problem solving.

3.4 ASK to THINK – TEL WHY®©

King's (1997) ASK to THINK – TEL WHY®©[2] is also a question-based collaborative learning procedure. It can be used for scripting collaboration in several forms of high-level learning where one partner assumes the role of *teacher* and the other the *learner* and partners alternate roles (see King, Staffiri, & Adelgais, 1998, for use of this procedure in mutual peer tutoring). The focus of the script is on use of five different types of questions which learning partners carefully sequence to scaffold their learning from comprehension checking and consolidation of prior knowledge to building new knowledge and monitoring thinking.

The learning partner in the teaching role (the ASK to THINK role) is called the *questioner* and the learning partner in the TEL WHY role explains (tells why and how) and elaborates (makes connections) and is referred to as the *explainer* (note that this role combines the activities of explanation and elaboration). This clear differentiation of roles makes it easier for the teaching partner to focus on asking questions rather than "lecturing" to the partner, which is more likely to elicit explanations and elaboration from the explainer.

The particular questions provided are generic questions that learners use to generate specific questions on the text or other material to be learned. A question-asking sequence begins with review questions and proceeds to more-sophisticated thought-provoking questions, with hint and probing questions as well as metacognitive questions interjected as needed. These questions, when posed, prompt partners to make corresponding responses. In this way learners continuously help each other build on their own and each other's previous contributions so as to "scaffold" knowledge construction to progressively higher levels (Vygotsky, 1978). During any learning session partners exchange roles. Collaborative learning with ASK to THINK – TEL WHY®© has been successful for students as young as fourth grade.

Cognitive, metacognitive and socio-cognitive processes underlying ASK to THINK – TEL WHY®©. To begin with, partners are provided with a

[2] ASK to THINK – TEL WHY®© is a registered trademark and the learning procedure itself is copyrighted by Alison King, 1994a, 1997, 1998 and 1999. Neither the names ASK to THINK – TEL WHY®© or ASK to THINK nor the particular learning procedure known by that name and described herin may be used for any commercial, teaching, or training purpose whatsoever or any other purpose without prior written permission from Alison King.

prompt card containing the questions and sequence as well as the TEL WHY role, and this serves as an external representation of the script to support cognitive activity. As with the GSPS procedure, use of a prompt card is meant to reduce learners' cognitive load by distributing some of the cognition to the external cards; at the same time the content of those questions (as well as their sequence) should keep learners' attention focused on the learning task at hand. For a fuller account of distributed cognition in ASK to THINK – TELWHY®©, see King (1998), where a description can be found of *what is being distributed* during the procedure and *how cognitions are distributed* across the learning pair and various aspects of their learning environment.

All of the cognitively effective activities of explanation, elaboration, asking thought-provoking questions, argumentation, and reconciling cognitive discrepancies are incorporated into the ASK to THINK – TEL WHY®© script. And the script is designed so that these activities induce a variety of cognitive, metacognitive, and socio-cognitive processes in learners. For example, when questioners begin the procedure by asking review questions (e.g., "What does ... mean?"); in so doing, they are not only assessing their partners' memory for the material and their understanding of it, they are also monitoring their own comprehension and repairing the knowledge base. Those review questions activate whatever knowledge partners have on the topic and elicit their definitions, descriptions, explanations, and elaborations. If an answer to a review question is incomplete, the questioner asks probing questions (e.g., "Tell me more about ...") to prompt the explainer to expand on an idea, clarify a point, be more explicit, or in some other way elaborate. When responses are incorrect or partial, hint questions (e.g., "Have you thought about ...?") are asked. Hint questions provide clues or partially-framed answers so as to guide explainers to repair any knowledge deficits or errors in reasoning and integrate the modification into their mental representations of the material. With a shared knowledge base firmly in place, learners proceed to construct new knowledge onto that base by asking and answering thinking questions (with hint and probing questions as needed). Examples of thinking questions and the specific cognitive processes they are intended to induce include: "What are the implications of ... for ...?" (analysis and inferencing), "What disadvantage might there be to using ...?" (speculation), "What is the difference between ... and ... in terms of ... ?" (analysis, comparison, and application of criteria), "What evidence is there to support the contention that ...?" (evaluation and evidence-based reasoning), and "What might be a counter-argument for ...?" (inferencing and logical reasoning). Such questions are designed to require going beyond the text material to induce higher-level thinking and learning. Thinking questions scaffold learners in creating links between ideas and between the new material and prior knowledge. Asking and answering thinking questions can not only increase the number of connections in learners' knowledge structures, it can also create a variety of different kinds of connections (such as comparison connections and evaluation connections), and therefore numerous and varied cues for retrieval and additional knowledge building.

Question sequencing from review questions through thinking and metacognitive questioning and responding serves to both control the progression of learning and monitor its extensiveness. The sequence also keeps the cognitive and socio-cognitive processing focused on the mutual scaffolding of learning to increasingly higher levels. Asking and answering metacognitive ("thinking-about-thinking") questions such as "How did you figure that out?" can function as a way for partners to monitor their own and their partner's thinking; they also serve to consolidate learning and make it more readily retrieved by generating additional cues for recall.

This questioning and answering is a socio-cognitive process characterized by mutuality and interdependence of roles. The question asked generally determines the response made which in turn dictates the next question, both its form (review, probing, hint or thinking) and its content (as the questioner builds on the explainer's response). Because of the interdependence and mutuality inherent in the activity of asking and answering thinking questions and hint and probing questions, partners' socio-cognitive processes can be induced, meaning is negotiated, and knowledge jointly constructed. The frequent exchange of roles required in this script also reinforces interdependence.

4. THE QUESTION OF SELF-REGULATION OF COLLABORATION SCRIPTS

For some face-to-face scripted collaboration approaches there is the expectation that the script itself will be internalized so that learners can become self-regulated in their use of it. According to Vygotskian thinking, the actions of the roles and any verbal prompts can be internalized as inner speech and then used by the learner to self-prompt actions in similar situations (Rogoff, 1990; Vygotsky, 1978).

However, because scripts vary in their complexity and goals, some lend themselves better than others to learners' appropriation (see Rogoff, 1990) through internalization. Some scripts simply label roles, activities or strategies, and sequence of activities and depend heavily on extensive teacher modeling and coaching in how those roles and activities are to be "played out" during interaction (e.g., Herrenkohl & Guerra's Cognitive Tools and Intellectual Roles [CTIR], 1998; Palincsar & Brown's Reciprocal Teaching, 1984). Other scripts (e.g., GSPS & ASKto THINK – TEL WHY®©)provide more specific guidance for interaction through use of explicit scaffolds that prompt specific kinds of dialogue that can, in turn, induce the intended cognitive and socio-cognitive processes. With some scripts the focus is on structuring the form of learner interaction, other scripts focus more on influencing the level of thinking and activity during that interaction. Some scripts must be rigidly adhered to and learners cannot modify the externally-provided roles, actions, or sequence (e.g., Scripted Cooperation) and this is necessary and appropriate (who can imagine checking the summary before summarizing?); while other scripts provide learners with a great deal of flexibility within the script's basic parameters. Some scripts are designed to be used in each and every instance of the collaborative activity (e.g., Scripted Cooperation, as well as essentially all computer-supported on-line scripts); in other scripted collaboration procedures, use of the script is less rigid and can be eventually faded because learners have internalized the basic script or have adapted it to fit their unique uses and internalized the modified script (e.g., Reciprocal Teaching; King's GSPS & ASK to THINK – TEL WHY®©).

Reciprocal Teaching is readily internalized by learners because of the separation of roles and simplicity of script. Indeed the goal of Reciprocal Teaching is for the procedures and roles to be learned and practiced in a group context with the intention that the procedure will eventually be used independently by learners to promote their reading comprehension. Because the roles are alternated during Reciprocal Teaching, learners get experience playing all four roles in the script so that over time those roles are internalized. Teacher and peer modeling of each role helps to make the roles easier to remember and assume later on. Over time roles are internalized to the extent that they can be played with self-prompting rather than external guidance. Presumably this can lead to learners being self-regulated in their reading comprehension when they are able to prompt their own execution of the entire script as they read independently.

Similarly, when GSPS is used over time external prompts can be faded as learners internalize the script and no longer need their question cards. It is expected that individuals will use their internalized GSPS guiding questions to self-regulate their problem solving either on their own or in collaboration with others.

Even some complex scripts can be appropriated by learners for later use. For example, ASK to THINK – TELWHY®© appears to be very complex because of its explicit dialogue prompts and sequencing; however several features of this procedure promote internalization. For example, role exchange provides opportunities for partners to play both roles and internalize them. Modeling supports this role appropriation. Also, the dialogue prompts (the questions) can be appropriated. An advantage of scripting dialogue is that aspects of the dialogue used during scripted collaboration, particularly the scaffolding prompts, are readily internalized as what Vygotsky (1978) refers to as *inner speech*. Appropriating the actual dialogue as inner speech can allow learners to engage in *self talk* (e.g., posing the questions to themselves) to prompt their own cognitive processes in subsequent similar situations (Rogoff, 1990; Vygotsky, 1978). Taking over the prompting role themselves facilitates learners' moving from being script-regulated to being self-regulated in many face-to-face collaborative learning contexts.

For designers of scripted collaborative learning and researchers assessing the effectiveness of these approaches it is important to be aware of what cognitive, metacognitive, and socio-cognitive processes their scripts are likely to induce in learners. Knowing more precisely what is learned and how it is learned through scripting can improve script design to facilitate collaborative learning and perhaps also promote eventual self-regulation of learning with those scripts.

REFERENCES

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

Bearison, D. J. (1982). New directions in studies of social interactions and cognitive growth. In F. C. Serafica (Ed.), *Social-cognitive development in context* (pp. 199-221). New York: Guilford.

Bell, P. (2004). Promoting students' argument construction and collaborative debate in the classroom. In M. C. Linn, E. A. Davis, & P. Bell (Eds.), *Internet environments for science education* (pp. 114-144). Mahwah, NJ: Erlbaum.

Britton, B. K., Van Dusen, L., Glynn, S. M., & Hemphill, D. (1990). The impact of inferences on instructional text. In A. C. Graesser & G. H. Bower (Eds.), *Inferences and text comprehension* (pp. 53-87). San Diego: Academic Press.

Brown, A. L., & Campione, J. C. (1986). Psychological theory and the study of learning disabilities. *American Psychologist, 41*, 1059-1068.

Chan, C. K. K., Burtis, P. J., Scardamalia, M., & Bereiter, C. (1992). Constructive activity in learning from text. *American Educational Research Journal, 29*, 97-118.

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

Chi, M. T. H., & VanLehn, K. A. (1991). The content of physics self-explanations. *Journal of the Learning Sciences, 1*, 69-105.

Cobb, P. (1988). The tensions between theories of learning and instruction in mathematics education. *Educational Psychologist, 23*, 78-103.

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

Dansereau, D. F. (1988). Cooperative learning strategies. In C. E. Weinstein, E. T. Goetz, & P. A. Alexander (Eds.), *Learning and study strategies: Issues in assessment, instruction, and evaluation* (pp. 103-120). New York: Academic Press.

De Lisi, R., & Golbeck, S. L. (1999). Implications of piagetian theory for peer learning. In A. O'Donnell & A. King (Eds.), *Cognitive Perspectives on Peer Learning* (pp. 3-37). Mahwah, NJ: Erlbaum.

Gick, M. L. (1986). Problem-solving strategies. *Educational Psychologist, 21*, 99-120.

Graesser, A. C. (1992). *Questioning mechanisms during complex learning.* Technical report, Cognitive Science Program, Office of Naval Research, Arlington.

Herrenkohl, L. R., & Guerra, M. R. (1998). Participant structures, scientific discourse, and student engagement in fourth grade. *Cognition and Instruction, 16,* 431-473.

Hogan, K., Nastasi, B. K., & Pressley, M. (1999). Discourse patterns and collaborative scientific reasoning in peer and teacher-guided discussions. *Cognition and Instruction, 17*, 379-432.

Johnson, D. W., & Johnson, R.T. (1993). Creative and critical thinking through Academic Controversy. *American Behavioral Scientist, 37*, 40-53

Kiewra, K. A. (1989). A review of note-taking: The encoding-storage paradigm and beyond. *Educational Psychology Review, 1,* 147-172.

King, A. (1989). Effects of self-questioning training on college students' comprehension of lectures. *Contemporary Educational Psychology, 14* ,1-16.

King, A. (1991). Effects of training in strategic questioning on children's problem-solving performance. *Journal of Educational Psychology, 83*, 307-317.

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

King, A. (1997). ASK to THINK - TEL WHY®©: A model of transactive peer tutoring for scaffolding higher-level complex learning, in *Educational Psychologist, 32*(4), 221-235.

King, A. (1998). Transactive peer tutoring: distributing cognition and metacognition. *Educational Psychology Review, 10*, 57-74.

King, A. (1999). Discourse patterns for mediating peer learning. In A. O'Donnell & A. King (Eds.), *Cognitive Perspectives on Peer Learning* (pp 3-37). Mahwah, NJ: Erlbaum.

King, A., & Rosenshine, B. (1993). Effects of guided cooperative questioning on children's knowledge construction. *Journal of Experimental Education, 61*, 127-148.

King, A., Staffieri, A., & Adelgais, A. (1998). Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning. *Journal of Educational Psychology, 90*, 134-152.

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

Kuhn, D., Shaw, V., & Felton, M. (1997). Effects of dyadic interaction on argumentative reasoning. *Cognition and Instruction, 15*, 287-315.

Larson, C. O., Dansereau, D. F., Goetz, E. T., & Young, M. D. (1985). *Cognitive style and cooperative learning: Transfer of effects.* Paper presented at the annual meeting of the Southwest Educational Research Association, Austin, TX.

Leont'ev, A. N. (1932). Studies in the cultural development of the child, 3: The development of vocabulary attention in the child. *Journal of Genetic Psychology, 37*, 52-81.

Lepper, M. R., Aspinwall, L. G., Mumme, D. L., & Chabay, R. W. (1990). Self-perception and social perception processes in tutoring: Subtle social control strategies of expert tutors. In *Self-inference processes: The Ontario Symposium* (pp. 217-237). Hillsdale, NJ: Erlbaum

Luria, A. R. (1928). The problem of the cultural behavior of the child. *Journal of Genetic Psychology, 35*, 493-506.

Mugny, G., & Doise, W. (1978). Socio-cognitive conflict and the structure of individual and collective performances. *European Journal of Social Psychology, 8*, 181-192.

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

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and monitoring activities. *Cognition and Instruction, 1*, 117-175.

Palincsar, A. S., & Herrenkohl, L. R. (1999). Designing collaborative contexts: Lessons from three research programs. In A. O'Donnell & A. King (Eds.), *Cognitive Perspectives on Peer Learning* (pp. 151-177). Mahwah, NJ: Erlbaum.

Perret-Clermont, A. (1980). *Social interaction and cognitive development in children.* New York: Academic Press.

Piaget, J. (1985). *The equilibrium of cognitive structures: The central problem of intellectual development.* Chicago: University of Chicago Press.

Pressley, M., McDaniel, M. A., Turnure, J. E., Wood, E., & Ahmad, M. (1987). Generation and precision of elaboration: Effects on intentional and incidental learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*, 291-300.

Pressley, M., Symons, S., McDaniel, M. A., Snyder, B. L., & Turnure, J. E. (1988). Elaborative interrogation facilitates acquisition of confusing facts. *Journal of Educational Psychology, 80*(3), 268-278.

Pressley, M., Wood, E., Woloshyn, V. E., Martin, V. King, A., & Menke, D. (1992). Encouraging mindful use of prior knowledge: Attempting to construct explanatory answers facilitates learning. *Educational Psychologist, 27*, 91-109.

Rogoff, B. (1990). Apprenticeship in thinking: *Cognitive development in social context.* New York: Oxford University Press.

Roschelle, J. (1992). Learning by collaboration: Convergent conceptual change. *The Journal of the Learning Sciences, 2*, 235-276.

Rosenshine, B., & Meister, C. (1994). Reciprocal teaching: A review of the research. *Review of Educational Research, 64*, 479-530.

Salomon, G. (1997). No distribution without individuals' cognition: A dynamic interactional view. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations* (pp. 111-138). New York: Cambridge University Press.

Salomon, G., & Globerson, T. (1989). When teams do not function the way they ought to. *International Journal of Educational Research, 13*, 89-99.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understandings.* Hillsdale, NJ: Erlbaum.

Spurlin, J. E., Dansereau, D. E., Larson, C. O., & Brooks, C. W. (1984). Cooperative learning strategies in processing descriptive text: Effects of role and activity level of the learner. *Cognition and Instruction, 1*, 451-463.

U. S. Department of Education (1986). *What works: Research about teaching and learning.* Washington, DC.

Vaughan J. L., & Estes, T. H. (1986). *Reading and reasoning beyond the primary grades.* Boston: Allyn & Bacon.

Vedder, P. (1985). *Cooperative learning: A study on processes and effects of cooperation between primary school children.* Gronigen, The Netherlands: University of Gronigen.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.* Cambridge, MA: Harvard University Press.

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

Webb, N., Ender, P., & Lewis, S. (1986). Problem solving strategies and group processes in small group learning computer programming. *American Educational Research Journal, 23*, 243-251.

Webb, N., & Farivar, S. (1994). Promoting helping behavior in cooperative small groups in middle school mathematics. *American Educational Research Journal, 31*, 369-395.

Webb, N. M., & Palincsar, A. S. (1996). Group processes in the classroom. In D. C. Berliner & R. C. Cafree (Eds.), *Handbook of Educational Psychology* (pp. 841-873). New York: Simon & Shuster Macmillan.

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

Wright, L. (1995). Argument and deliberation: A plea for understanding. *Journal of Philosophy, 92*, 565-585.

Chapter 3

CAN PEOPLE LEARN COMPUTER-MEDIATED COLLABORATION BY FOLLOWING A SCRIPT?

Nikol Rummel and Hans Spada
Albert-Ludwigs-Universität, Freiburg

Abstract: Our central hypothesis is that partners who jointly work on a task in a computer-mediated setting following a collaboration script, can acquire collaborative skills that will help to improve the collaboration in subsequent tasks as well as their outcome. In an experimental study, a collaboration script was provided for a first computer-mediated collaboration in one experimental condition. Meantime, in a different experimental condition, the collaborators observed a model-collaboration. Learning effects of script and model were expected to become evident in the process and outcome of a second, unscripted computer-mediated collaboration. Compared to two control conditions (a condition with unsupported collaboration during the learning phase and a condition without a learning phase) both the script condition and the model condition showed positive effects on process and outcome during the application phase. This leads to the conclusion that collaboration scripts can indeed constitute a promising instructional method to promote collaborative competences and to improve subsequent computer-mediated collaboration.

1. BACKGROUND

As outlined in the introductory chapter of this book and illustrated by the individual chapters, collaboration scripts have proven to be powerful strategies for supporting collaboration in learning and problem-solving contexts. Moreover, they have shown such beneficial effects on collaborations in a variety of face-to-face settings ranging from collaborative learning of science texts in college settings (e.g., O'Donnell, 1999; O'Donnell & Dansereau, 1992) to collaborative problem-solving and learning in mathematics in a school setting (Berg, 1993). Also in a variety of computer-mediated collaboration settings empirical evidence has been established that scripts can provide effective support for collaboration (for examples refer to the chapters of this volume).

However, Dillenbourg (2002) expressed concern that there may be a danger to "overscript" collaborative interaction. Scripting collaboration might prevent the independent, exploratory thinking required for generative learning or problem-solving. This, Dillenbourg argues, is especially true for highly coercive scripts which dictate interaction in a very detailed and inflexible way. A high degree of coercion might also decrease student motivation. We have argued along similar lines (see Rummel & Spada, 2005b) that the motivation theory of Deci and Ryan (1985) indicates that collaboration scripts may cause motivational problems and reactance towards the script as this theory regards self-determination as central for motivation. Observations pointing in the same direction have also been made by researchers who have successfully applied scripts to enhance computer-mediated collaboration in their own research (Bruhn, 2000; Kollar, 2001). Negative motivational effects can be expected in particular if collaboration is scripted over an extended period of time and over many collaborative sessions (Hron, Hesse, Reinhard, & Picard, 1997).

Against this background, an important question is, whether central elements of a collaboration script can be learned from a scripted session, and then serve to promote subsequent unscripted collaboration (in the following called *learning-from-script hypothesis*). Such learning effect of a collaboration script would make it unnecessary to continue the scripting and risk motivational drawbacks, but collaborators could themselves maintain a fruitful collaboration following their internalized script rules.

In the following paragraphs, we first discuss approaches to scripting collaboration in the literature that are relevant for our own approach. We characterize our collaboration script within a classification framework proposed by Dillenbourg (2002) and introduce the experimental paradigm it has been investigated with. We provide empirical support for our hypothesis from a recent study (see Rummel & Spada, 2005b). In the final section, the central results gained from our experimental study are evaluated in the light of the key question of this chapter: Can people learn computer-mediated collaboration by following a script?

2. SCRIPT APPROACHES RELEVANT TO THE LEARNING-FROM-SCRIPT HYPOTHESIS

The central idea of collaboration scripts is to foster fruitful collaboration by externally structuring the interaction process. The script guides the collaborating partners through a defined sequence of interaction phases. For each phase specific activities are prescribed like roles in a theater or movie script. By enforcing specific kinds of activities among the collaborators, the script is expected to prompt cognitive, metacognitive and social processes by participants that might otherwise not occur (see chapter by King, this volume). This description holds true for collaboration scripts at a very general level even though there are great differences in the specific ways collaboration scripts have been realized. In this section we present some illustrative examples of collaboration scripts relevant to our script approach.

2.1 Collaboration scripts in traditional collaboration research

Several of the classical script approaches have originated from the idea to improve individual learning by including collaborative elements in the instruction. As it had become obvious that collaboration would in many cases not facilitate learning just by itself (e.g., Azmitia, 1988; Cohen, 1994; Dillenbourg, Baker, Blaye, & O'Malley, 1995; Slavin, 1983), scripting approaches were developed with the goal to design collaboration in a way to make it fruitful for learning.

One of the most well-known approaches to scripting is the so-called MURDER script developed by Dansereau and colleagues (Dansereau, 1988; O'Donnell & Dansereau, 1992; for an overview see O'Donnell, 1999). This script is directed at helping two college students in learning collaboratively from text material in science. The script includes detailed instruction on how to proceed in jointly processing the text at hand. At the outset, the text is broken into sections. Then students first read a section individually. Next, they take turns in the role of the recaller (summarizing the major ideas of the passage) and the listener (monitoring the explanation: detecting errors, identifying omissions and asking for clarifications). Together, the partners elaborate on the contents of the section and try to make it more memorable by connecting it to previous knowledge and to mnemonic illustrations like images or analogies. This cycle is repeated for each section of the text. Finally, the students review the text once more. In sum, the central activities prompted by the script are (see O'Donnell, 1999): the overt verbalization of thinking about the text, the metacognitive activities involved in active listening (e.g., error detection), and the emphasis on continuous elaboration. Further, cross-modeling among the two peers is an important element.

In a similar way, the script developed by Palincsar & Brown (1984; see also chapter by King, this volume) provides support for the collaborative processing of text. The main difference to the scripting approach by Dansereau and O'Donnell is that the reciprocal teaching technique was developed for the classroom. The teacher and several students take turns in performing the different steps of the script. Thus, the teacher provides an *expert model*, particularly in the beginning. As the students become more proficient, the teacher retreats and the cross-modeling among peers becomes more and more important. The reciprocal teaching script involves four main activities: formulating questions on the text, summarizing, clarifying difficulties with the text, and making predictions about how the text will continue. These steps are repeated for the different passages of the text.

Many of the classical script approaches that were developed to facilitate collaborative learning are built on the assumption that through extended practice with the script, the learners would little by little internalize relevant elements of the the script so that the external scaffolding provided by the script could be faded out over time (e.g., Palincsar & Brown, 1984). In other words, similar to the hypothesis we pose in this chapter, the classical script approaches comprised the notion that good collaboration would be learned from scripted interaction. However, they did not assess this assumption directly, for example by analyzing subsequent, unscripted collaborations for script elements, but the internalization of the fruitful script was inferred from learning gains.

2.2 Collaboration scripts in CSCL research

In computer-mediated collaboration settings, scripts can be incorporated in the structure of the technical environment. Many computer-mediated settings include shared workspaces that may be prestructured by embedding script information that can guide the collaborators and enhance content-specific negotiation (Bruhn, Fischer, Gräsel, & Mandl, 2000; Suthers, 2001; see also Ertl, Kopp, & Mandl, this volume; Weinberger, Ertl, Fischer, & Mandl, 2005; Weinberger, Stegmann, Fischer, & Mandl, this volume). Thus, in order to support computer-mediated collaboration, interaction design by means of scripts can be combined with interface design to provide an optimal environment for productive collaboration (Hesse, Garsoffsky, & Hron, 1997).

For example, the collaboration script implemented by Hron et al. (1997) regulated the interaction of two people in a text-based computer-mediated setting. Collaborative task of the dyad was to perform corrections on a diagram depicting some biological structure. The script dictated a dialog cycle, prompting each step that had to be performed in the interface of the collaborative environment. First, one partner was asked to propose a correction; then, the other partner was requested to express his approval or disapproval of the proposition. If disapproving, he was asked to give an explanation, which the first partner had to concur with in turn. This cycle went on until they had agreed on a correction. Only then would the system allow them to actually perform the correction in the graphical tool of the interface.

Pfister and Mühlpfordt (2002; see also Haake & Pfister, this volume) have developed a collaboration script structuring the discourse among learners in different knowledge domains in a similar way as the script by Hron et al. (1997). They call their script approach a *learning protocol*. In their computer-mediated collaboration setting, the interface requires participants to choose from a predefined menu of contribution types (e.g., question, explanation) before typing their specific contribution. Also, participants are asked to indicate which previous contribution in the discourse their contribution is relating to. When the message is then added to the dialog history, the chosen contribution type and the reference are indicated. In addition, the system assigns alternating roles to participants (e.g., tutor), which then again have an impact on the contribution types available to that person.

It is obvious that this kind of scripting exerts a high degree of coercion (Dillenbourg, 2002) on the collaborators as the script is enforced by the collaborative environment. In consequence, stronger negative motivational reactions to the scripting would be expected as compared to the traditional script approaches presented above. In further contrast to the above approaches, the script approaches in computer-mediated settings have concentrated exclusively on providing online support during a particular ongoing collaboration. The expectation that the script would be internalized or learned by the collaborating partners and would thus also affect subsequent collaborations has not been in the focal point.

3. A SCRIPT FOR LEARNING TO COLLABORATE

As we have pointed out above, the hypothesis that scripted collaboration should lead to an internalization of relevant aspects of the script is not new. Script approaches developed in research on collaborative learning (e.g., O'Donnell & Dansereau, 1992; Palincsar & Brown, 1984) assumed that the scaffolding provided by the script could be faded out because learners would internalize the script over time. However, this hypothesis was not tested systematically. And in the context of computer-mediated collaboration, collaboration scripts have so far been applied as *online* support measures for ongoing collaboration. Dillenbourg (2002, p. 81) has presented some initial thoughts on the idea that the cognitive processes instructed by a collaboration script in computer-mediated collaboration may be internalized by the collaborators. However, he also acknowledged that empirical evidence for this consideration has yet to be provided.

Our hypothesis is that scripts structuring computer-mediated collaboration online can also trigger *learning about collaboration*. We think that partners who follow a collaboration script while jointly working on a problem-solving task can acquire collaborative skills, which will then improve the collaborative process and outcome in a subsequent unscripted collaboration. Some evidence in support of our hypothesis can be found in the literature on the problem-based learning approach in medicine (e.g., Barrows, 1986; Cameron, Barrows, & Crooks, 1999). The central goal of this approach is to involve the students in constructive knowledge-building activities while solving authentic problems. In addition to the acquisition of contextualized domain knowledge, learners are expected to develop *procedural knowledge of the clinical reasoning process* (Barrows, 1986). It is this emphasis on the acquisition of procedural skills in addition to domain knowledge where the problem-based learning approach shares ground with our hypothesis that scripted collaboration may promote collaborative process skills. Moreover, the situated learning approach (Greeno and MMAP, 1998; Lave & Wenger, 1991) provides support for our learning-from-script hypothesis from a different angle: it supports our notion that meaningful collaborative activities guided by a script should yield much better learning effects (including better transferability to new collaborations) than direct instructions of the relevant script contents could.

3.1 Testing the learning-from-script hypothesis: The experimental framework of our collaboration script

The *collaborative scenario* we chose for our research was the computer-mediated solving of complicated psychiatric cases that required both medical and psychological expertise. Dyads of advanced medical and psychology students were asked to make use of their complementary expertise and jointly develop a diagnosis and sketch a suitable therapy plan for the cases. The two partners collaborated computer-mediated via a desktop-videoconference including personal text editors and a shared text editor.

Why choosing to investigate the interdisplinary collaboration among dyads of medical and psychology students, and why using a desktop video conference system for the collaboration?

The collaboration of psychologists and medical doctors (or of medical doctors with different specialization) is increasingly regarded to be of importance for the well-being of patients. A successful treatment is only possible if a correct diagnosis has been deduced from the symptoms of a patient. However, some symptoms can indicate both a physical as well as a mental diagnosis. Moreover, there is a high comorbidity of mental and physical disorders. While the interdisciplinary collaboration on the treatment of inpatients in hospitals is one topic, the other relevant question is how to encourage and support collaboration among locally distributed medical and psychological practices. In this context video conference systems have been advocated as a particularly suitable solution (Köhler & Trimpop, 2004).

In a desktop videoconference, participants at different locations each sit at their individual computer and communicate with one another via an audio-video connection. On the computer screen they can see video pictures of the remote partners. Each video picture is captured by a small camera sitting on top of the computer screen or placed directly to the side of the screen. A continuous audio channel provides the possibility to talk to the remote partners. In addition, desktop videoconferences support application sharing, which adds the important chance to not only view, but also jointly edit text or visual material. Moreover, the possibility to combine a shared application (e.g., a text editor) with an individual one (e.g., an individual text editor for each partner) offers ideal conditions to include both joint and individual work phases in a remote collaboration. In sum, video-mediated communication systems support complex synchronous interactions with an exchange of both verbal as well as nonverbal information (Finn, Sellen, & Wilbur 1997). However, there are also particular challenges that collaborators in such a setting are likely to experience and that a collaboration script ought to help them overcome

Depending on the quality of the audio and video transmission, delays in the transmission of sound and picture cause specific communication problems such as breaks and overlaps in the dialogue structure (Angiolillo, Blanchard, Israelski, & Mané, 1997). But even with a very good technical quality, the expenditure of any form of collaborative activity in videoconferences is increased by an additional and more explicit effort (Anderson et al., 1997) concerning, for example, the processes of grounding (Clark & Brennan, 1991), turn-taking, or giving feedback. O'Conaill and Whittaker (1997) found that video-mediated communication is more "lecture-like", that is, handing over turns is done in a very formal way by using questions or naming the next speaker. One reason for this finding might be that the visual contact possible in desktop-videoconference settings is in most cases limited to seeing the face or upper body part of the partner; usually eye contact is not possible, neither is gaze awareness (Angiolillo et al., 1997; Joiner, Scanlon, O'Shea, Smith, & Blake, 2002). It has also be criticized that joint awareness of and attention towards objects in the environment are not supported by videoconference systems (Kato et al., 2002). This may lead to problems for example when jointly using shared applications. It can be concluded that in a desktop videoconference setting the collaborative process requires extra effort, and good and explicit coordination is necessary.

3.2 Testing the learning-from-script hypothesis: A script to teach collaboration

We have adopted a particular *experimental paradigm* (see Rummel & Spada, 2005b) to test our hypothesis that scripts can promote the acquisition of collaborative skills. The paradigm comprised two phases of computer-mediated collaborative problems-solving: one task was solved during the so-called *learning phase*; a second task was solved during the *application phase*. In the learning phase, a collaboration script was provided to structure the interaction and to build up collaborative competences, which were then expected to become evident in the process and outcome of the second – unscripted – collaboration during the application phase.

A detailed script prescribing specific phases for their interaction was provided to the partners during their first collaborative case (i.e., during the learning phase). Table 3-1 gives an overview of the phases instructed by the script. Participants received the script instructions in written format. The instructions in the script were given in the following way: "Please, use the following 5 minutes to ask your partner any questions you might have about the case. Make use of each other's knowledge to clarify information given to you about the patient in the case description before turning to the diagnosis."

With the classification framework proposed by Dillenbourg (2002) our collaboration script can be characterized as follows:

The script defines phases with specific *tasks/activities* for each phase. Completion criteria are particular results that have to be achieved in a given phase (for example, individual notes on the diagnosis ought to be taken in Phase 4, see Table 3-1), but also the time limits that are set for each phase. The criterion for *group formation* (or better *team formation* since we are looking at dyads here) is the complementarity of domain knowledge (psychology vs. medicine). This criterion is relevant to form the dyads at the outset of the collaboration. The *group size* varies between phases: the partners are instructed by the script to either work jointly or individually. The *distribution of input* is preset by the complementarity of expertise in the dyad and further increased experimentally by a domain-specific distribution of text material. The input distribution then induces the activity distribution (Dillenburg, 2002, p. 74). For example, the medical student is expected to know more about possible side effects of the current medication of the patient than the psychologist. Consequently, he is going to be the one to explain those to the psychologist. Also, he will have to make sure that the side effects are taken into account when diagnosing the patient. On the other hand, the psychologist has knowledge about psycho-therapeutic treatments. Hence, he is going to be in charge of planning the psychotherapy for the patient. A corresponding *distribution of activities* across the partners is facilitated by the division of labor prescribed by the script. The *mode of interaction* is synchronous. However, this does not preclude phases where the partners work individually. To the contrary, particularly in the given collaboration scenario with relevant knowledge distributed over the partners, taking time for individual reflection and work is of great importance (see Hermann, Rummel, & Spada, 2001). Given the distinction King (this volume) makes between cooperative and collaborative learning, one might thus argue that our script should be labeled a *cooperation script*. However, as we have discussed before (Rummel & Spada, 2005b) making this distinction on the basis of the task division is somewhat arbitrary (see also Dillenbourg, 1999), and it would be difficult to make for our script – and the given task – as it comprises both collaborative as well as cooperative elements. It is cooperative, because a division of labor and individual work on subtasks is inevitable at some points given the complementary expertise of the partners. On the other hand, one could also define the script as collaborative, because the partners will need to work jointly a great deal to integrate both the medical and the psychological perspective. And the script does provide instruction for socio-cognitive processes to occur during the collaboration. Overall, we would characterize our script as a *collaboration script* rather than a cooperation script. As already mentioned, *time management* is supported by the script by prescribing a particular time frame for each phase in addition to the task definition.

Table 3-1. The phases of the collaboration script in Rummel & Spada (2005b)

Phase		
1.	👤 👤	short initial coordination: define objectives of task
2.	👤	scan case description for potential problems with understanding, formulate questions to the partner
3.	👤 👤	mutually answer questions, coordination: determine course of action (content, time, roles)
4.	👤	individually work on **diagnosis**, take individual notes
5.	👤 👤	exchange notes, discuss individual ideas
6.	👤	revise individual solutions and formulate final solution for diagnosis
7.	👤 👤	copy individual parts of solution (diagnosis) in shared editor, integrate
8.	👤 👤	formulate goals for the **therapy**
9.	👤	individually work on therapy plan (division of labor!), take individual notes
10.	👤 👤	exchange notes, discuss individual ideas
11.	👤	revise individual solutions and formulate final solution for therapy
12.	👤 👤	copy individual parts of solution (therapy) in shared editor, integrate final check of entire joint solution

In addition to the above *syntactical* (i.e., structural) attributes of our script, it can also be characterized along the lines of four *semantic* dimensions (Dillenbourg, 2002).

The *design rationale* behind our collaboration script is based on assumptions of what aspects characterize a good collaboration in the given type of scenario. We integrated empirical findings from different strands of research and came up with three levels merging in a good collaboration (see Rummel & Spada, 2005a): a level concerning the coordination of the joint work; a level concerning aspects of the communication; and a level concerning domain-specific demands for a good joint solution. Above all, the *learning objective* of our script was to promote the acquisition of meta-cognitive knowledge: we aimed at improving our participants' knowledge and skills in collaborating by providing them with a script.

Our collaboration script exerts a medium to high *degree of coercion* on the collaborators as it gives very detailed instructions of who should do what and how much time is available for the activity. Thus it prescribes the interaction of the collaborating partners to a great extent. However it is not equally as coercive as, for example, the script by Hron et al. (1997) because the communication interface does not enforce the script. Our script was provided on paper which entailed the danger that the collaborators could simply not adhere to its instructions. Indeed it was sometimes the case that the experimenter had to intervene and reprove the collaborators to stick to the script. In other words, the *adoption* of the script was not a trivial issue (first level of *appropriation* according to Dillenbourg, 2002). Not so much because people had problems following the script instructions, but because sometimes they did not want to follow them. The second level of appropriation according to Dillenbourg (2002), the *internalization* of the script over (scripted) time is in the focus of our research. As has been stated above, we have developed an experimental paradigm to investigate precisely this question. We have tested the effects of our collaboration script on the application phase at different levels (see Table 3-2): the collaborative process, its outcome – the joint solution for Case 2 – as well as an individual posttest.

4. RESULTS IN SUPPORT OF THE LEARNING-FROM-SCRIPT HYPOTHESIS FROM AN EMPIRICAL STUDY

4.1 Method

In an *experimental study* (see Rummel & Spada, 2005b) we compared learning effects of scripted collaboration (script condition) to three other conditions (see Table 3-2): a condition in which the collaborators observerd a worked-out collaboration example during the learning phase (model condition), a condition with unscripted collaboration during the learning phase (unscripted condition), and a condition without a learning phase (control condition). The unscripted condition served as a second control condition testing the learning effects of our script against the potential learning effects of merely gaining experience in collaborating. The model condition on the other hand was testing the alternative hypothesis that learning to collaborate from observing a model collaboration might be more successful than learning from scripted interaction, because the observation of the model allows to dedicate more cognitive capacity to elaborative meta-cognitive activity on the rationale of the different phases in the collaborative process.

Table 3-2. Experimental design in Rummel & Spada (2005b)

	Experimental variation			
	Script condition	Model condition	Unscripted condition	Control condition
Learning phase (Case 1)	Learning from scripted computer-mediated collaboration	Observational learning from a worked-out example of collaboration	Learning from unscripted computer-mediated collaboration	No learning phase
	Assessment of effects of instruction provided in the learning phase			
Application phase (Case 2)	Computer-mediated collaborative problem-solving in all four conditions → Data on collaborative process → Data on outcome (joint solution)			
	Posttest	→ Data on individual knowledge		

As can also be seen from Table 3-2, the experimental variation was implemented during the first phase of the experiment, the learning phase. With exception of the control condition all dyads were engaged with the first psychiatric case during this phase – either solving it themselves or watching it being solved by the models. In the application phase, effects of the experimental intervention were assessed as dyads were collaboratively solving the second psychiatric case. Both the collaborative process itself and the joint solution were analyzed as dependent variables. In addition, an individual posttest was administered testing for the participants' knowledge on what makes a good collaboration in the present scenario on two subscales.

In all four conditions nine dyads were tested each consisting of a medical student and a student of psychology (a total of 72 participants). Students were recruited during university lectures and seminars, and received a financial compensation for their voluntary participation. All students were at an advanced level of proficiency in their studies at the time of participation.

4.2 Results

The overall comparison of all four conditions yielded the following results. In the application phase (i.e., test phase) dyads in the script and the model condition outperformed dyads in both the unscripted and the control condition at three levels (see Table 3-2): the collaborative process, its outcome (the joint solution of case 2), and an individual posttest administered after the application phase. The results are summarized in Table 3-3 (for a more detailed account of the results, see Rummel & Spada, 2005b).

Table 3-3. Summary of results*

	Script condition	Model condition	Unscripted condition	Control condition
Logfile analysis on individual time	+	+	-	-
Analysis of dialogs on time management and coordination	+	+	-	?
Quality of joint solution				
Diagnosis	-	+	-	-
Therapy plan	+	+	-	-
Posttest				
Scale A	+	+	-	-
Scale B	+	+	-	-

*The table summarizes the results on all dependent variables. In the table "+" denotes a positive result, "-"denotes a negative one, and "?" a result which is difficult to interpret.

With regard to the *collaborative process*, an analysis of the activity patterns gained from log-file data revealed that dyads in the script and model condition adhered to the proportions of individual and joint work instructed during the learning phase. They showed a substantial *amount of individual work*, which – as had been hypothesized (see Rummel & Spada, 2005b) – proved to be an important predictor of successful performance on the joint solution. The script and the model conditions did not differ substantially on this variable. A detailed analysis of the *dialog data* with a coding scheme assessing the frequencies of utterances on a number of categories pertaining to aspects of good collaboration, like coordination and time management, revealed that dyads in the script and model condition did engage more in such process management than did dyads in the unscripted condition.

With regard to the *joint solution* for Case 2, dyads in the script condition produced the best therapy plan, however, they were outperformed by the model condition with regard to the diagnosis. We will come back to this point in the discussion.

Participants in the script and model conditions further showed better performance on the two subscales of the individual *posttest* testing knowledge about aspects relevant for good collaboration and problem-solving: subscale A, asking for knowledge on what makes a good collaboration in the given type of scenario at a more general level, and subscale B, asking participants to describe elements of a good therapy plan. Again, the script and the model condition did not differ in their performances.

In sum, the results support our hypothesis that *collaboration scripts can trigger learning about collaboration* and thus improve subsequent collaboration. The results attained by the model condition were similar to those of the script condition. Thus, observing a worked-out collaboration example during the learning phase also proved to be an effective instructional measure to improve the subsequent collaboration in the application phase.

5. CONCLUSIONS: CAN PEOPLE LEARN COMPUTER-MEDIATED COLLABORATION BY FOLLOWING A SCRIPT?

In analogy to Salomon's distinction between effects *of* technology and effects *with* technology (Salomon, 1993), we propose that one could differentiate between effects *of* the script versus effects *with* the script (A. Ertelt, personal communication, September 29, 2004). Salomon (1993) has argued that particularly in education improved performance while a tool is used should not be the primary goal. Rather, tool use should be aimed at improving the learners' abilities independent of continued tool use. In a similar way we have argued in this chapter, that we believe that the ultimate goal of supporting collaboration with a script should not lie in the improved, scripted performance, but in an improved ability to collaborate in fruitful ways, of course, leading to a good performance.

The promising results of the above study lead us to the conclusion that in future research the learning effects of collaboration scripts on the acquisition of collaborative skills should be investigated more systematically. We further propose that if collaboration scripts foster learning about collaboration, they can constitute a means to improve computer-mediated collaboration in the long term. Of course, in the study reported here we have only provided evidence of the learning effects on one delayed collaboration and, therefore, potential long-term effects cannot be claimed yet, but require further research on a greater number of more delayed collaborations.

An area that demands further consideration and empirical investigation is the question of how to design collaboration scripts from an instructional point of view to yield the best learning effects.

Guiding the collaborating partners to reflect on the relevant features of the script might be a promising measure to promote its internalization and acquisition as a standard of subsequent collaboration. Some indication that reflection of the scripted activities could improve learning is provided by the cognitive apprenticeship approach (e.g., Collins, Brown, & Newman, 1989). In this research, open verbalization and reflection accompanying the own behavior have proven to be important scaffolding strategies to support the acquisition of complex cognitive skills. It is our assumption that to foster the reflection of a scripted cooperation could provide such *procedural facilitation* (Scardamalia & Bereiter, 1985) while learning to collaborate from the script. More support for the beneficial effects of reflection and elaboration can be found in research on worked-out examples, where self-explanations (Renkl, Stark, Gruber, & Mandl, 1998) and instructional explanations (Renkl, 2002) have shown to improve the processing of worked-out examples and in consequence learning of the demonstrated problem-solving strategies.

Moreover, reference to the literature of learning from worked-out examples gives important indications for how to design the transition from scripted collaboration to independent unscripted problem-solving in order to achieve the best learning effects. The collaboration could initially be guided by a script like the one implemented in the present study. The scripting should then be faded out giving way to increasingly independent problem-solving. However, this transition should not happen abruptly from one task to the next as in the study described above, but in a supported step-by-step procedure similar to the transition from studying worked-out examples to solving problems proposed by Renkl, Atkinson, Maier and Staley (2002). Such a transition from collaborating with script to collaborating without script support is particularly interesting with regard to a potential long-term intervention with collaboration scripts

One issue that has to be addressed, particularly when aiming at script-learning, are motivational problems that collaboration scripts may cause and that may impede the adoption and, consequently, the internalization of the script. Although we did not directly assess motivational effects as part of our study, we think that negative motivational effects of the scripting in the learning phase became evident in the initial phase of the collaboration in the application phase. These negative effects are reflected in the performance of dyads in this condition on the diagnosis, which was poor compared to the outstanding performance they then showed on the therapy plan. However, the fact that dyads in the script condition then yielded very good results with regard to their therapy plans, and also showed high performances in the posttest, supports the conclusion that collaboration scripts can constitute a powerful method to promote collaborative skills. Yet, precautions have to be taken to avoid motivational problems arising from the scripting during the learning phase. The above proposal to guide the collaborating partners to reflect on the relevant features of the script might also be a promising measure to prevent negative motivational effects of scripting. If reflection on the script is promoted, the collaborating partners might gain a better understanding of the relevance of the scripted activities. This might then improve their openness towards the script and promote its internalization.

REFERENCES

Anderson, A. H., O' Malley, C., Doherty-Sneddon, G., Langton, S., Newlands, A., Mullin, J., et al. (1997). The impact of VMC on collaborative problem solving: An analysis of task performance, communicative process, and user satisfaction. In K. E. Finn, A. J. Sellen & S. Wilbur (Eds.), *Video mediated communication* (pp. 133-155). Mahwah, NJ: Lawrence Erlbaum Associates.

Angiolillo, J. S., Blanchard, H. E., Israelski, E. W., & Mane, A. (1997). A Technology constraints of video mediated communications. In K. E. Finn, A. J. Sellen, & S. B. Wilbur (Eds.), *Video mediated communications* (pp. 51-74). Mahwah, NJ: Lawrence Erlbaum Associates.

Azmitia, M. (1988). Peer interaction and problem-solving: When are two heads better than one? *Child Development, 59*, 87-96.

Barrows, H. S. (1986). A taxonomy of problem-based learning methods. *Medical Education, 20*, 481-486.

Berg, K. F. (1993). *Structured cooperative learning and achievement in a high school mathematics class.* Paper presented at the annual meeting of the American Educational Research Association, Atlanta.

Bruhn, J. (2000). *Förderung des kooperativen Lernens über Computernetze: Prozess und Lernerfolg beim dyadischen Lernen mit Desktop-Videokonferenzen.* Frankfurt a. M.: Peter Lang.

Bruhn, J., Fischer, F., Gräsel, C., & Mandl, H. (2000). Kooperatives Lernen mit Mapping-Techniken. In H. Mandl & F. Fischer (Eds.), *Wissen sichtbar machen. Wissensmanagement mit Mapping-Techniken* (pp. 119-132). Göttingen: Hogrefe.

Cameron, T., Barrows, H. S., & Crooks, S. M. (1999). Distributed problem-based learning at Southern Illinois University School of Medicine. In C. Hoadley & J. Roschelle (Eds.), *Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference* (pp. 86-94). Mahwah, NJ: Lawrence Erlbaum Associates.

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

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction* (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum Associates.

Dansereau, D. F. (1988). Cooperative learning strategies. In C. E. Weinstein, E. T. Goetz, & P. A. Alexander (Eds.), *Learning and study strategies* (pp. 103-120). San Diego, CA: Academic Press Inc.

Deci, E. L., & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior.* New York: Plenum.

Dillenbourg, P. (1999). Introduction:What do you mean by "collaborative learning"? In P. Dillenbourg (Ed.), *Collaborative learning. Cognitive and computational approaches* (pp. 1-19). Amsterdam: Pergamon.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen: Open Universiteit Nederland.

Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. (1995). The evolution of research on collaborative learning. In P. Reimann & H. Spada (Eds.), *Learning in humans and machines: Towards an interdisciplinary learning science* (pp. 189-211). Oxford: Elsevier/Pergamon.

Finn, K. E., Sellen, A.J., & Wilbur, S. B. (Eds.) (1997). *Video-mediated communication.* Mahwah, NJ: Lawrence Erlbaum Associates.

Greeno, J. G. & the Middle School Mathematics through Applications Project Group. (1998). The situativity of knowing, learning and research. *American Psychologist, 53*, 5-26.

Hermann, F., Rummel, N., & Spada, H. (2001). Solving the case together: The challenge of net-based interdisciplinary collaboration. In P. Dillenbourg, A. Eurelings, & K. Hakkarainen (Eds.), *Proceedings of the First European Conference on Computer-Supported Collaborative Learning* (pp. 293-300). Maastricht: McLuhan Institute.

Hesse, F. W., Garsoffsky, B., & Hron, A. (1997). Interface-Design für computerunterstütztes kooperatives Lernen. In L. J. Issing & P. Klimsa (Eds.), *Information und Lernen mit Multimedia* (pp. 253-267). Weinheim: Psychologie Verlags Union.

Hron, A., Hesse F. W., Reinhard, P., & Picard, E. (1997). Strukturierte Kooperation beim computergestützten kollaborativen Lernen. *Unterrichtswissenschaft, 25*(1), 56-69.

Köhler, T., & Trimpop, R. (2004). Sehen und gesehen werden: Teleradiologie mittels Desktop- Videoconferencing. In W. Bungard, B. Koop, & C. Liebig (Eds.), *Proceedings zur 3. Tagung der Fachgruppe Arbeits- und Organisationspsychologie.* München: Rainer Hampp.

Kollar, I. (2001). *Gewissheits- und Ungewissheitsorientierung beim kooperativen Lernen mit Videokonferenzen - der Einfluss verschiedener Strukturierungsmaßnahmen.* Unveröffentlichte Magisterarbeit. Ludwig-Maximilians-Universität München: Institut für Empirische Pädagogik und Pädagogische Psychologie.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambrigde, UK: Cambrige University Press.

O'Conaill, B., & Whittaker, S. (1997). Characterizing, predicting and measuring video-mediated communication: A conversational approach. In K. E. Finn, A. J. Sellen, & S. Wilbur (Eds.), *Video-mediated communication* (pp. 107-131). Mahwah, NJ: Lawrence Erlbaum Associates.

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

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

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 1*(2), 117-175.

Pfister, H.-R., & Mühlpfordt, M. (2002). Supporting discourse in a synchronous learning environment: The learning protocol approach. In G. Stahl (Ed.), *Proceedings of the Computer Support for Collaborative Learning (CSCL) 2002 Conference* (pp. 581-589). Hillsdale, NJ: Lawrence Erlbaum Associates.

Renkl, A. (2002). Worked-out examples: Instructional explanations supplement self-explanations. *Learning and Instruction, 12*, 529-556.

Renkl, A., Atkinson, R. K., Maier, U. H., & Staley, R. (2002). From example study to problem solving: Smooth transitions help learning. *Journal of Experimental Education, 70*, 293-315.

Renkl, A., Stark, R., Gruber, H., & Mandl, H. (1998). Learning from worked-out examples: The effects of example variability and elicited self-explanations. *Contemporary Educational Psychology, 23*, 90-108.

Rummel, N., & Spada, H. (2005a). Instructional support for collaboration in desktop videoconference settings: How it can be achieved and assessed. In R. Bromme, F. W. Hesse, & H. Spada (Eds.), *Barriers and biases in computer-mediated knowledge communication - and how they may be overcome*. Dordrecht, NL: Kluwer Academic Publishers.

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

Salomon, G. (1993). On the nature of pedagogic computer tools: the case of the Writing Partner. In P. Lajoie & S. J. Derry (Eds.), *Computers as cognitive tools* (pp. 289-317). Hillsdale, NJ: Lawrence Erlbaum Associates.

Scardamalia, M., & Bereiter, C. (1985). Fostering the development of self-regulation in children's knowledge processing. In S. F. Chipman, J. W. Segal, & R. Glaser (Eds.), *Thinking and learning skills: Research and open questions* (pp. 563-577). Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Suthers, D. (2001). Collaborative representations: Supporting face to face and online knowledge-building discourse. *Proceedings of the 34th Hawaii International Conference on the System Sciences (HICSS-34), January 3-6, 2001*. Maui, Hawaii (CDROM), Institute of Electrical and Electronics Engineers (IEEE).

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

Chapter 4

SCRIPTING IN NET-BASED MEDICAL CONSULTATION: THE IMPACT OF EXTERNAL REPRESENTATIONS ON GIVING ADVICE AND EXPLANATIONS

Anne Runde, Rainer Bromme, and Regina Jucks
Westfälische Wilhelms-Universität, Münster

Abstract: In this chapter, a distinction is made between three concepts of scripting communication: 1) social roles as a non-deliberative, non-instructional form of scripting, 2) explicit and 3) implicit scripting. Both of the latter are forms used in instructional collaborative settings to influence and change behavior. As we established in a previous study, external representations both structure and constrain asynchronous expert-layperson communication (Bromme, Jucks, & Runde, 2005). According to Suthers (e.g., Suthers & Hundhausen, 2003), external representations guide discourses. Because shared external representations have the potential to influence learning and collaboration processes in a non-directive manner, we define the concept of *representational guidance* as implicit scripting. In the present study, we focused on the potential to support shared decision making when patients seek advice from medical doctors through the Internet. When communicating via computers, it is easy to make external representations available to both communication partners. Therefore, whether or not shared graphic representations function as an implicit script and have an impact on the communication content was tested empirically. Our main hypothesis is as follows: with a shared external representation in the background more specialist arguments are brought forward than without such a representation. In accordance with this hypothesis, we found that the external representation had a considerable influence on content selection during the discourse.

1. BACKGROUND

More and more people with varied degrees of expertise turn to the Internet as a source of medical information. Many web sites offer medical advice. In addition there are a number of other options available for using the Internet in the context of health communication, such as mailing lists, communities, newsgroups, chats or conventional email. It is possible to differentiate still further: who provides information and who communicates with whom? Some self-help groups, for example, use the Internet. Although most of these people have no medical training, they usually have some expertise about a particular disease. Additionally, there is the option of communicating via the Internet with professional people who *have* had medical training. Net-based communication scenarios between medical experts and laypersons have begun to complement traditional doctor-patient communication to a significant degree. Uncertainty is the most common reason for visiting a website to obtain health information (Pezza, 1990). The Internet is used to obtain a second opinion, complementing what has already been learned face-to-face (Koc, 2002) as well as a first opinion in order to decide wether a consultation is necessary or not. After a television program on health matters, people frequently take up the offer of communicating in a chat room with a medical expert from the program. Health Internet portals, such as health.yahoo.com, not only provide up-to-date information on many diseases or access to encyclopedias with information on topics such as laboratory results, medication and symptoms, but also allow the user to consult an expert via e-mail or chat on a particular health topic (sometimes on payment of a fee).

An extensive use of the Internet is concomitant with a changed role for the patient in the doctor-patient relationship. Patients might have gathered relevant information themselves and this might lead to a more symmetric doctor-patient-relationship. Traditionally, the doctor-patient-relationship has been perceived in a paternalistic manner. Accordingly, the patient was totally dependent on the doctor, who assumed the dominant role and decided on the course of action with respect to the disease. The patient was then expected to follow the doctor's advice. In this scenario, the personal opinions and values of the patient played only a minor role in decision-making processes. Over the last few years, an alternative understanding of the relationship between doctor and patient has become accepted, in which the patient is "allowed" to assume a more active role. As proposed in the shared-decision-making model and as the name implies, decision-making is shared by doctor and patient, both during the consultation process and throughout the treatment. In 2001, the German Federal Ministry of Health and Social Security launched shared decision-making as its main research topic (http://www.patient-als-partner.de [patient as a partner]). The shared-decision-making model has various different facets and conditions that must be met for it to succeed. One of these is integrating the patient's preferences into the decision-making process with respect to treatment or prevention, together with providing medical or scientific knowledge from the doctor. Moreover, if the model is to succeed, a mutual exchange of information and sufficient relevant expert advice to patients is a sine qua non. Only then can the patient be considered 'medically mature' and able to take an active role in decision making. Ideally, the advice should comprise a detailed explanation of the disease (risks, potential complications, other related diseases, etc.) and possible treatments (What therapies are there and what are the various advantages and disadvantages, etc.?).

The shared-decision-making approach seems essential for ensuring patient compliance and for improving other relevant patient-related variables such as satisfaction (e.g., Dowell & Hudson, 1997). However, the practical application of the approach appears problematic. Several empirical studies demonstrate that conveying expert information often plays an insignificant role in doctor-patient communication, even in dedicated counseling settings. Such counseling is frequently conducted on a superficial level as far as expert knowledge is concerned, the advice given being restricted to behavioral instructions (Tulsky, Chesney, & Lo, 1995). Many patients report dissatisfaction with their communication with physicians. They complain, for example, of a lack of information about possible alternative treatments. There are undoubtedly a number of situations where it might be more effective not to indulge in long and informative dialogues. Conversely, it is possible for patients to end up with unsuitable treatment if they, along with their opinions and preferences, are not included in the decision-making process (Coulter, Entwistle, & Gilbert, 1999).

Consequently, deploying the shared-decision-making model successfully requires a change of behavior on the part of both communication partners. If doctors wish to help patients make an informed decision and share the decision-making with them, it is imperative that they not only give behavioral advice, but also supply expert content and related background information. An informed decision is defined as the reasoned choice by a 'reasonable' individual on the basis of relevant information about the advantages and disadvantages of all possible courses of action and in conformity with individual attitudes (Bekker et al., 1999). To sum up, there are conflicting and diverging goals and processes in doctor-patient communication. On the one hand – and in the sense of the shared-decision-making model – it is necessary to communicate relevant medical information and to improve patient involvement. On the other hand, traditional social roles create communication patterns which prevent patients from assuming more responsibility.

How then should the behavior necessary for shared decision-making be fostered?

There are different approaches to fostering patient involvement in the communication process with physicians. Such processes usually consist of training one or other of the communication partners (the patient or the doctor). Various empirical studies provide evidence that it is worth attempting to integrate new thinking about the role of the patient into these schemes. After training patients, Kaplan, Greenfield, and Ware (1989) found indications that there was more patient involvement during interaction with the doctor and better health outcomes when patients had been taught to ask more appropriate questions during consultation, and had been given more specific information about treatment options than patients in a control group who had received only general information. As a result, diabetes patients who had received training and became more involved, achieved better control over their blood sugar levels.

Training schemes generally focus on improving the quality of advice doctors give to patients and on increasing patient knowledge by furnishing them with more material about the relevant health topic. However, most training is cost-intensive and not easily adapted to specific patients. It is neither possible to train all potential patients to ask appropriate questions, nor does the provision of expert information such as brochures on display in the waiting room guarantee that they will be read, let alone understood. Training doctors has also proven difficult, primarily because they have such tight schedules. Adapting training courses for doctors along these lines and educating patients in areas relevant to their complaints through training schemes certainly makes sense, but needs to be supplemented by further measures, starting with general communication skills and an appropriate structuring of the communication situation itself.

In this connection, Internet-based communication between doctor and patient is a good starting point, given that, as mentioned earlier, it enables patients to seek information, and because the technical environment makes it possible to offer direct support.

In the following section, we present a method which uses external representations (graphics) in Internet-based communication settings. We refer to this method as an *implicit script*. In order to clarify this approach, it is necessary to explain the concept of implicit scripts in more detail (for a general introduction to the concept scripts see Fischer, Kollar, Haake, & Mandl, this volume, and King, this volume). Therefore, we describe three concepts of scripting – social roles, explicit and implicit scripts – and differentiate the first two clearly from the concept of implicit scripting.

2. THREE CONCEPTS OF SCRIPTING: SOCIAL ROLES, EXPLICIT SCRIPTS, AND IMPLICIT SCRIPTS

2.1 Social roles as determinants of the structure of interaction

Social roles are prototypical scripting concepts, first introduced to psychology (by Schank & Abelson, 1977). In this "original" sense, the behavior of physician and patient follows a well-established structure of which both parties are aware. Scripting through social roles determines their behavior inherently, whereas instructional concepts of scripting refer to a planned impact on specific behavior, guided by instructional objectives. The chapters of this book are mostly based on this instructional understanding of *scripting*. In contrast to the concept of scripts in instructional settings, social roles lead to a pre-structuring of communication situations without being *deliberately influenced* by someone else (except for the process of learning to act according a social role, for example, during an apprenticeship). As the very familiar example of the restaurant script illustrates (Schank & Abelson, 1977, see King, this volume), agents follow a script and in so doing, enact their social roles. The roles of doctor and patient are scripts in that traditional sense (Schank & Abelson, 1977). They are relatively fixed roles which do not have to be negotiated afresh at the start of every new interaction. There is little scope for individual structuring, which derives mainly from the typical dependency relationship between the two speech partners. Roles of this kind reduce the complexity of interactional possibilities (Luhmann, 1999), which implies that roles do help facilitate and simplify interaction processes. They pre-structure them and provide patterns of behavior. In this manner, the social roles of communication partners function as scripts.

The substantial difference in knowledge between the two sides is largely responsible for these behavior patterns. The expert is the adviser, explainer, and helper. Laypersons find themselves in a position where they are dependent on the expert. They have a problem with respect to which they turn to a communication partner. They ask questions, hoping for advice and help. The clear allocation of roles determines the selection of communication content, which in turn structures the communication. In addition to the cognitive processes involved in communicating information, expert-layperson communication is characterized by a number of typical social processes and features. Wintermantel (1991) stresses the asymmetry of this communication situation (compare Watzlawick, Beavin, & Jackson, 2000), also describing it as an instructional dialogue. On the basis of the unequal distribution of knowledge between the two communication partners, the expert dominates the discourse. "This particular dominance relation, due to knowledge superiority, provides a regularity which is accepted by both participants at the outset. For the one delivering the instruction, it should be clear that s/he is ready to transfer her/his knowledge. For the one who wants to learn in the course of the dialogue it implies acceptance of the dominance of the expert" (Wintermantel, 1991). An awareness of this asymmetry and its implications can be regarded as common ground between the communication partners (Stalnaker, 1978). As a further characteristic feature, Wintermantel (1991) describes the goal orientation of communication: "... the explicit intention of the two participants to contribute to a common goal, namely that of equalizing the initial unequal knowledge distribution". However, these characteristics are understood as inevitable in such a situation. Apart from the fact that some communication tasks and situations where the layperson assumes the more dominant role are conceivable (e.g., during the course of an anamnesis in the doctor-patient communication), it is questionable whether mutual goal orientation can always be assumed, in particular a mutual goal orientation where both partners are interested in equalizing the differing knowledge levels. A number of conflicting goals are conceivable which may not always lead to the expert giving comprehensive expert information.

As with other roles (e.g., gender roles), notions of the ideal roles of doctor and patient have changed over time (Coulter, 1997). Presently, the generally prevailing view is that patients should be (put) in a position where they are able to make decisions regarding their health in cooperation with the doctor. However, so far, there is little evidence of this happening in practice – needs and reality are still poles apart. The traditional roles still seem to apply, dominating communication structures. As already mentioned, the long-established roles and their accompanying scripting can prevent communication from being as effective as it should be. Therefore, it is necessary to implement alternative scripts which influence the behavior in the appropriate manner.

2.2 Cooperation scripts as explicit instructions

In contrast to the inherent structuring of communication through social roles, instructional principles are implemented when designing goal-oriented communication (cooperation scripts are a good example). Instructional designs use these scripts primarily to support participants in collaborative learning situations so as to encourage the selection of appropriate learning and communication strategies.

Cooperation scripts regulate the sequence and timing of learning and interaction activities. "The roles and the nature and timing of the activities of the participants are specified." (O'Donnell & Dansereau, 1992, p.122). For instance, they prescribe when and how one learner should give feedback to another learner, and how the latter in turn should react. It is possible to differentiate between cooperation scripts which focus more on interaction and those which focus more on communication content. Ertl, Fischer, and Mandl (in press) also differentiate between support for collaborative learning on a conceptual level, comparable to the focus on content, as opposed to a socio-cognitive level, which is comparable to the interaction focus. Interactional scripts, which foster appropriate cooperation patterns, are concerned mainly with the construction of knowledge. Content-based scripts focus on supporting the processing of the task content, by, for example, repeatedly asking learners to make certain inferences about the text they have studied.

Hence, cooperation scripts can be characterized as follows: a) they are deployed deliberately and b) they contain specific instructions. We therefore define this type of scripting as *explicit* scripting, which directly regulates the communication process and the structuring principles of which the communication partners have conscious knowledge.

Recently, studies have confirmed the effectiveness of various cooperation scripts in computer-mediated environments (e.g., the contributions in this volume; Dillenbourg, 2005; Rummel & Spada, 2005; Weinberger, Reiserer, Ertl, Fischer, & Mandl, 2005). Hence, it can be argued that this form of scripting communication situations has proven valid. Due to their explicit character, these scripts always create a specific instruction setting which in turn demands the communication partners' full attention and a high level of motivation. Intervening in a "natural" communication process and structuring the process by coercion can, of course, also have negative consequences, too. Moreover, Weinberger, et al. (2005) discuss "overscripting effects" (compare Dillenbourg & Jermann, this volume) which "may ease the learning task in an exaggerated manner, reducing the complexity of learning tasks, and hampering productive discourse of learners" (p.35). Particularly in real-life situations, there is a danger that explicit scripts, like cooperation scripts, could have these disadvantages (Baker & Lund, 1997).

2.3 External representations as implicit scripts

Apart from using explicit scripts for structuring the communication process, there is also the option of *implicit* scripting. By means of shared external representations (available to both communication partners), it is possible to facilitate an implicit structuring of the content of a communication process.

We believe that the concept of representational guidance suggested by Suthers et al. (e.g., Suthers, 2005; Suthers & Hundhausen, 2003) describes this script effect, which will be explained in more detail below. In fact, Suthers introduced the concept in another context and from another perspective, but embedded in the theoretical debate about scripting, representational guidance clearly works as implicit scripting. Suthers (2005) uses the concept of representational guidance (or affordances) to explain collaborative learning processes in computer-mediated communication. He emphasizes that external representations on the one hand constrain the communication processes (in the past, he labeled this effect as *representational biases*), but on the other hand, this restriction may lead to a task relevant focus. For this reason, we describe this effect as scripting. Depending on the specific content and character of a shared external representation in a collaborative setting, the structure and content of the interaction will be *guided.*

Suthers and Hundhausen (2003) examine the impact of external representations on collaborative learning processes. They claim that different representation formats each have their special characteristics which influence the cognitive processes of "readers". Accordingly, the attention necessary to acquire information is guided along different paths, depending on the representation format. In addition to the effects of external representation on individual learning and problem-solving, the authors assume that external representations provide special supportive effects in connection with collaborative learning scenarios: they (a) stimulate negotiating of meaning (b) provide points of reference for abstract concepts and (c) play a role in the implicit assessment of common ground.

Collaborative learning processes and their results depend on the specific manner in which external representations have been presented. Depending on the format chosen, different types of information are emphasized. Suthers and Hundhausen (2003), for example, investigated the influence of three different representation formats (graphics, table, text) on collaborative discourse and learning outcome. Learners were asked to summarize information about public health issues, formulate hypotheses and identify relationships, with the aim of finding a solution to a particular problem. They were required to perform the task in one of the three representation formats. The representation format did, indeed, influence the focus of the discourse. If, for example, the information was presented to learners in tabular form, they gave valid relationships far more frequently than learners in dyads where graphics and texts were used. Furthermore, the representation format influenced overall learning success. The influence was strongest for those learners who had worked on the graphic format. The authors argue that not every format is equally well suited to a particular task. It could, for example, be observed that learners working with the graphic format exchanged many irrelevant kinds of information as well.

However representations already containing expert information can also influence communication processes. In one of our own studies, we examined the impact of external representations on the recipient-orientation of experts in an asynchronous email study (Jucks, Bromme, & Runde, 2003, 2006). The investigation entailed medical experts receiving an email inquiry from a (fictitious) patient. They had an external representation on hand in the form of an illustration containing the expert information necessary for their answer. In one condition, they were informed that the same information was also available to the patient. In another condition, participants were told that the patient did not have the same information. Informing participants of the shared availability of the illustration had a significant effect on the choice of content used in the explanations: medical experts with a *shared* illustration on hand gave a more detailed answer. They used more technical terms and explained more of the specialized interrelationships depicted in the illustration. These results can be interpreted as representational guidance effects. If shared, the external representation determines the choice of communication content and consequently functions as an implicit content script. Whereas Suthers and Hundhausen (2003) pointed out that different representation formats lead to different main focuses, Bromme et al. (2005) stressed the importance of *sharing* a representation in a communication scenario.

As well as establishing a mutual cognitive framework, it can be assumed that this implicit form of intervention does not have disadvantages as demotivating or distraction which sometimes come along with explicit forms of cooperation scripts. It does not intervene directly in the "natural" communication process. On the one hand, the implicit script of a shared external representation does not have the negative effects of "classic" scripting in the form of explicit cooperation scripts, but on the other hand, it is doubtful whether implicit scripting has any effect at all on communication, because it is non directive.

To sum up, we described three kinds of scripting communication. The non-deliberative, non-instructional form of scripting through social roles is a culturally-formed anticipation of behavioral patterns. It cannot be regarded as an instructional intervention and is not limited to a specific communication scenario, unlike deliberative, instructional scripting. This kind of scripting is generally introduced by a third party within a collaborative learning scenario. We differentiated between two kinds of deliberative scripts in instructional settings: explicit and implicit. Explicit scripting through cooperation scripts intervene directly in the communication process and this may be noticed by the participants. Implicit scripting represents a communication structure which is optional and introduced indirectly.

3. RESEARCH QUESTIONS

The results of the abovementioned studies of Suthers and Hundhausen (2003) as well as of Bromme, et al. (2005) provide evidence of the impact of shared external representation in the form of implicit scripts. In the study described below, this impact was tested in a synchronous expert-layperson scenario.

In the above section, an issue in doctor-patient communication was described and discussed: how can traditionally structured doctor-patient communication be shaped toward a more balanced and shared decision-making?

The question arises as to whether there are economical efficient ways and means of making the communication between doctor and patient more effective by stressing the relevance of content. For this purpose, we draw on the concept of representational guidance. If, in doctor-patient communication, a graphic illustration with relevant content is available, are the communication partners more likely to use the information given in this form during the discourse? As discussed above, this kind of intervention can be seen as an implicit content script. Illustrating the content imposes a certain *content* structure, which leads, in turn, to a formal *discourse* structure. It functions, in effect, as a third "speech partner" which can focus attention on the matter at issue again and again, thus helping to create both external and cognitive frameworks.

The representation of relevant content by means of logically-structured diagrams seems to be a particularly meaningful form of intervention in the context of instructional content scripts. For a communication situation between two speech-partners with different knowledge backgrounds, content representation by means of a logical diagram, such as a concept map, for example, can create an external cognitive framework.

We build on the results of Suthers et al. (e.g., Suthers & Hundhausen, 2003) and other researchers (e.g., van Boxtel, van der Linden, Roelofs, & Erkens, 2002) and our own email study (Bromme et al. 2005). These studies demonstrated the effect of shared external representations or the mutual externalization of expert content on the discourses structuring in collaborative learning settings. This resulted in higher task orientation and suggested the following assumption. If an illustration is provided to both speech partners in Internet-based doctor-patient communication, it encourages participants to use more specialist information during the interaction. In this manner, the shared illustration contributes to the success of shared decision-making and reduces the conflict arising from traditional social roles. Therefore, the key question is as follows: does a shared illustration function as an implicit content script and thus support shared decision-making? And how extensive is its impact?

In our study referred to above, we found evidence that shared external representations influence the content of asynchronous, Internet-based communication settings. We examined the influence of external representations in a synchronous communication setting (see also Runde, 2004).

4. MAIN FINDINGS

For this purpose, we asked a medical doctor to advise a medical layperson in a chatroom. In one condition, both doctor and layperson had a concept map at their disposal (experimental condition). In a second condition, only the doctor had this concept map (control condition). The concept map contained various relevant expert concepts and associations (see Figure 4-1). We used the CoolModes software (by Hoppe & collaborators, University of Duisburg; see Pinkwart, Hoppe, Bollen, & Fuhlrott, 2002) for the environmental design. The communication partners in the experimental condition were able to make written annotations on the concept map which were also visible on the other person's monitor.

Prior to the communication with the doctor, the medical laypersons were instructed to think of themselves as patients who had to make a decision on the choice of medication for hypercholesterolemia. They were also told that it would be possible for them to consult a medical doctor in the chatroom and that after the consultation, they would have to make a decision for or against a particular cholesterol-reducing medication. After the chatroom consultation, the subjects were asked a few questions and then asked to make their decision. We expected the availability of a shared concept map to influence the communication content.

In all, 36 dyads were examined. Half had a shared concept map on the monitor at their disposal (see Figure 4-1). Each dyad consisted of an advanced medical student and a medical layperson, that is, a student studying some other non-medical subject. The focus of the assessment was on the analysis of formal and, in particular, content-related aspects of the discourses. In the experimental condition with the shared concept map, discourses were more detailed in terms of the number of words used. Subjects also employed more specialist terms in this condition. These results can be attributed mainly to the fact that communication partners made more use of the specialist terms which were also contained in the map. A similar result can be found in the analysis of the specialist arguments. In the experimental condition, the medical experts used more specialist arguments and, above all, arguments which were also contained in the external representation. However, not only the experts were influenced by the shared concept map, but also the laypersons. They asked more medical questions about the content of the concept map. By contrast, there were no differences between the two conditions with regard to personal data given by laypersons and the behavioral instructions given by medical experts.

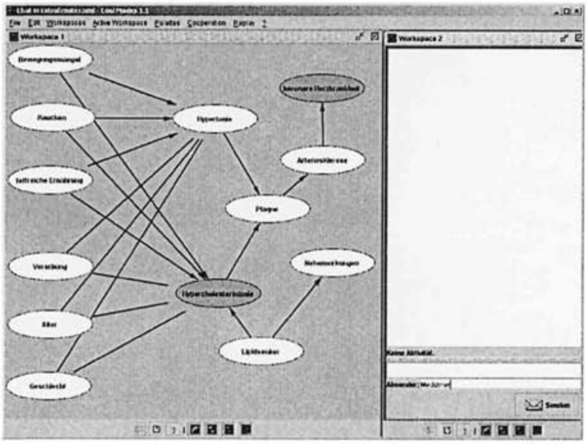


Figure 4-1. Environment of the experimental condition

The results confirm the hypothesis. The shared external representation, here in the form of a concept map, exerted a considerable influence on content selection in the discourse. The communication partners were guided towards the information depicted in the concept map, making it the content of their communication. Additionally and independently of sharing the concept map, there seems to be a basic "stock of information" that was conveyed in the communication, e.g., the anamnesis information and behavioral tips, and a selection of relevant specialist information. However, more extensive expert information was given more often in the experimental condition.

However, the influence of the shared external representation does not extend beyond the immediate subject matter. It could be assumed that the increased use of specialist arguments leads to a specialist focus in general like a priming effect. However, we are dealing with a very specific effect. We did not find a transfer effect. The content of the dyad discourses with the shared concept map did not generally include any more expert information. It seems, therefore, that the communication process is guided in a specific direction. Because of the selection of a control group in which the medical experts also had the specialist illustration, we can exclude any notion that the experts merely ticked off the information it gave them as a checklist for formulating their replies. If that had been the case, there would not have been any differences between the conditions.

All in all, the results reveal an implicit script effect analogous to the representational guidance concept of Suthers and Hundhausen (2003). The communication was influenced in the intended manner by sharing the structure and content of the external representation. The communication partners used more of the specialist information depicted in the external representation. The constrained effects on the content of the shared external representation also replicate the results from our previous email study (Bromme et al. 2005). Our earlier study demonstrated an increasing use of the illustrated information in an asynchronous communication setting when the external representation was shared with the communication partner.

5. CONCLUSIONS

The results of our study mentioned above, show that external representations guide discourse and, above all, the content of the medical advice in a certain direction and influence the selection of information. The representations influenced the communication between doctor and patient. In this connection, the content and the manner in which they were represented play an important role in the level and nature of the discourse (compare Jucks, Bromme, & Becker, 2006 for the impact of word use in graphic representations on experts' communication). Restrictively, it must be point out that we did not analyse real doctor-patient-communication. It is assumed that in such communication settings many other factors affect communication such as the strong dependence on the doctor, time and institutional limitation, anxiety and other emotions on part of the patients etc. From this it follows that we can not translate our findings directly into the 'real world' doctor-patient-communication. Nevertheless we can conclude that comparable effects occur different communication scenarios with great knowledge differences between the communication partners.

Although, apart from one compulsory topic, the tasks were open-ended and the intervention was not very directive, the external representation contributed towards guiding discourse content in the assumed direction. That is, it increased the amount of relevant expert knowledge content in the contributions. This is even more remarkable when we consider that no explicit request was made to use the representation. In this sense, the external representation functioned as an implicit content script which determined the issues discussed. The results of both the study using asynchronous communication scenarios, and the email study reported elsewhere (Jucks et al., 2006) point to this direction.

Despite the differences between our study design and that of Suthers and Hundhausen (2003), the reported results are quite comparable. Hence, the concept of representational guidance can be extended to expert-layperson communication scenarios and communicating with already-existing external representations.

The implicit form of scripting tested in this study has the advantage of influencing the communication process in a very unobtrusive manner without interrupting it, as is the case with such alternative forms of scripting as cooperation scripts. The specialized information is available to participants via the shared external representation throughout the communication. Although the shared illustration constantly reminds the speech partners of the meaning of certain specialized terms during the course of their communication, it does not interrupt them. The illustration becomes part of the "natural" communication process.

The question arises as whether the concentration on the specialized content can be fostered if the communication is scripted explicitly or at least partly explicitly. For example, the communication partners might be briefed to take the concept map into account during discourse or, as in Suthers' learning settings, asked to adapt the relevant content so as to produce a concept map themselves. This might probably have been at the expense of other, equally relevant discourse content, such as the exchange of anamnesis information and behavioral tips. Therefore, further research is necessary to investigate and differentiate between the conditions of those communication settings in which external representations are useful and those in which they can do more harm than good.

ACKNOWLEDGEMENTS

The study was part of a research project on expert-layperson communication in net based medical advice, (funded by the DFG, PIs R. Bromme & R. Jucks). We thank Brian Bloch for native speaker advice.

REFERENCES

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

Bekker, H., Thornton, J. G., Airey, C. M., Connelly, J. B., Hewison, J., Robinson, M. B., et al. (1999). Informed decision making: An annotated bibliography and systematic review. *Health Technology Assessment (Winchester, England), 3*(1), 1-156.

Bromme, R., Jucks, R., & Runde, A. (2005). Barriers and biases in computer-mediated expert-layperson-communication. In R. Bromme, F. W. Hesse, & H. Spada (Eds.). *Barriers,* *biases and opportunities of communication and cooperation with computers* (pp. 89-118). New York: Springer.

Coulter, A., Entwistle, V., & Gilbert, D. (1999). Sharing decisions with patients: Is the information good enough? *British Medical Journal, 318*, 318-322.

Coulter, A. (1997). Partnership with patients: The pros and cons of shared clinical decision-making. *Journal of health services research & policy, 2*(2), 112-121.

Dillenbourg, P. (2005). Designing biases that augment socio-cognitive interactions. In R. Bromme, F. W. Hesse, & H. Spada (Eds.). *Barriers, biases and opportunities of communication and cooperation with computers* (pp. 243-264). New York: Springer.

Dowell, J., & Hudson, H. (1997). A qualitative study of medication-taking behavior in primary care. *Family Practice, 14*(5), 369-375.

Ertl, B., Fischer, F., & Mandl, H. (in press). Conceptual and socio-cognitive support for collaborative learning in videoconferencing environments. *Computers & Education.*

Jucks, R., Bromme, R., & Becker, B.-M. (2006). Lexical entrainment – Is Expert's Word Use Adapted to the Addressee? Manuscript accepted for publication (major revisions needed). *Discourse Processes.*

Jucks, R., Bromme, R., & Runde, A. (2006). Explaining with non-shared illustrations: How they constrain explanations. To appear in *Learning and Instruction.*

Jucks, R., Bromme, R., & Runde, A. (2003). Audience Design von Experten in der netzgestützten Kommunikation: Die Rolle von Heuristiken über das geteilte Vorwissen. *Zeitschrift für Psychologie, 211*(2), 60-74.

Kaplan, S. H., Greenfield, S., & Ware, J. E. (1989). Assessing the effects of physician-patient interactions on the outcomes of chronic disease. *Medical Care, 27*(3), 110-127.

Koc, F. (2002). *Medizin im Internet. Evidenz-based-Medicine und Qualitätsmanagement online.* Berlin: Springer.

Luhmann, N. (1999). *Soziale Systeme. Grundriß einer allgemeinen Theorie* (7. Auflage). Frankfurt a. M.: Suhrkamp.

O'Donnell, A. M., & Dansereau, D. F. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interaction in cooperative groups: The theoretical anatomy of group learning* (pp. 120-144). New York: Cambridge University Press.

Pezza, P. E. (1990). Orientation to uncertainty and information seeking about personal health. *Health Education Research, 21*(2), 34-36.

Pinkwart, N., Hoppe, H. U., Bollen, L., & Fuhlrott, E. (2002). Group oriented modelling tools with heterogeneous semantics. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Lecture notes in Computer Science 2363, Intelligent Tutoring Systems* (pp. 21-30). Berlin: Springer.

Runde, A. (2004). *Die Rolle externer Repräsentationen in der netzgestützten Arzt-Patienten-Kommunikation.* Unveröffentlichte Dissertation der Westfälischen-Wilhelms Universität, Münster.

Rummel, N., & Spada, H. (2005). Instructional support for collaboration in desktop videoconference settings. In R. Bromme, F. W. Hesse, & H. Spada (Eds.). *Barriers, biases and opportunities of communication and cooperation with computers* (pp. 59-88). New York: Springer.

Schank, R., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Stalnaker, R. C. (1978). Assertion. *Syntax & Semantics: Pragmatics*, 315-332.

Suthers, D. D., & Hundhausen, C. D. (2003). An experimental study of the effects of representational guidance on collaborative learning processes. *The Journal of the Learning Sciences, 12*(2), 183-218.

Suthers, D. D. (2005). Technology affordances for intersubjective learning, and how they may be exploited. In R. Bromme, F. W. Hesse, & H. Spada (Eds.). *Barriers, biases and opportunities of communication and cooperation with computers* (pp. 295-314). New York: Springer.

Tulsky, J. A., Chesney, M. A., & Lo, B. (1995). How do medical residents discuss resuscitation with patients? *Journal of general internal medicine, 10*(8), 436-442.

van Boxtel, C., van der Linden, J., Roelofs, E., & Erkens, G. (2002). Collaborative Concept Mapping: Provoling and supporting meaningful discourse. *Theory into Practice, 41*(1), 40-46.

Watzlawick, P., Beavin, J. H., & Jackson, D.D. (2000). *Menschliche Kommunikation. Formen, Störungen, Paradoxien.* Bern : Huber 2000,

Weinberger, A., Reiserer, M., Ertl. B., Fischer, F., & Mandl, H. (2005). Facilitating collaborative knowledge construction in computer-mediated learning environments with cooperation scripts. In R. Bromme, F. W. Hesse, & H. Spada (Eds.). *Barriers, biases and opportunities of communication and cooperation with computers* (pp. 15-37). New York: Springer.

Wintermantel, M. (1991). Dialogue between expert and novice: On differences in knowledge and their reduction. In I. Markova & K. Foppa (Eds.), *Asymmetries in dialogue* (pp. 124-142). Hertfordshire: Harvester WheatsheafBarnes & Noble Books/Bowman & Littlefield Publishers, Inc.

Chapter 5

SCRIPTING LAYPERSONS' PROBLEM DESCRIPTIONS IN INTERNET-BASED COMMUNICATION WITH EXPERTS

Matthias Nückles, Anna Ertelt, Jörg Wittwer, and Alexander Renkl
Albert-Ludwigs-Universität, Freiburg

Abstract: In the information age, laypersons have to rely on experts in many domains and situations. Expert advice can be invaluable, for example, when new and complex software has to be learned, or an unexpected technical problem with the computer suddenly occurs. In order to communicate effectively with experts, laypersons should be able to provide the expert with a concise and comprehensive description of their problem. However, previous research on computer helpdesks has shown that laypersons' problem descriptions often suffer from a number of serious drawbacks. Their deficient and fragmentary knowledge makes it hard for them to formulate their queries in a way that would make it possible for the expert to understand their problem. Based on an analysis of these deficiencies, a problem formulation script was developed that supports laypersons in describing their problems with the computer. An experimental study showed that computer experts reconstructed the actual problem from the layperson's description best if the laypersons were prompted to describe successively (1) the aim of their interaction with the computer, (2) the steps they had so far undertaken, and (3) a hypothesis why they had failed to reach the aim. The script helped the laypersons to provide the expert with the relevant context information necessary to develop an adequate mental model of the layperson's problem.

1. INTRODUCTION

Expertise in using computers, so-called *computer literacy,* has gained the status of a cultural skill that is regarded almost as fundamental as the ability to read and to write (Süß, 2001; Wirth & Klieme, 2002). However, as hardware and software become ever more complex and powerful, acquiring expertise in the computer domain remains a daily challenge even for the experienced computer user (Kiesler, Zdaniuk, Lundmark, & Kraut, 2000).

Hence, reliance on expert advice can be invaluable, especially, when new and complex software has to be learned, or an unexpected technical problem with the computer suddenly occurs (Nückles & Stürz, 2006; Nückles, Wittwer, & Renkl, 2005). Internet-based helpdesks for hardware and software are a common and comfortable way to get expert advice. Such e-consulting services are maintained, for example, by large companies or university computing centers.

According to Alty and Coombs (1981) as well as Raskutti and Zukerman (1997), the users' description of their problem is the very starting point of every counseling and information seeking process. Therefore, the effectiveness of the advice the expert can give depends heavily on the user's ability to adequately present their query, that is, to provide the expert with a concise and comprehensive description of the problem at stake. Transcript analyses of face-to-face and asynchronous advisor-user interactions show, however, that users tend to be inappropriate in the presentation of their problems (Alty & Coombs, 1980, 1981; Coombs & Alty, 1980; Pollack, 1985). They often do not know what information they need to obtain in order to achieve their goals. Consequently, advisors must identify inappropriate queries and infer and respond to the goals behind them (Pollack, 1985). Only if the experts succeed in constructing a valid and coherent mental model of the problem from the client's description, can they provide instructions that help the user to understand and solve their problem.

Against this background, a scripting approach was developed to support laypersons in producing concise and comprehensive descriptions of their problems with the computer (Nückles & Ertelt, 2006). This approach is based on the idea that – despite their lack of domain specific knowledge – laypersons can draw on metacognitive knowledge from everyday problem-solving (Sinnott, 1989) that may help them generating better representations of their computer problems. Thus, the script approach presented in this chapter makes use of culturally shared knowledge about everyday problem-solving (Schank & Abelson, 1977) to support laypersons in their communication with experts in a domain-specific problem-solving context – a situation which is typically experienced as demanding and often also frustrating by many laypersons. The script consists of several prompts (Collins, Brown, & Newman, 1989; King, 1992) intended to induce the steps necessary for the composition of a concise and comprehensive problem description. Hence, in some respects, our problem formulation script (PFS) is comparable to King's guided strategic problem solving (GSPS) procedure (cf. King, this volume). However, whereas King intended to promote students' problem solving success by scaffolding their interaction when solving complex problems, our main intention was to help laypersons improving their description and presentation of computer problems to a computer expert.

Inasmuch as the problem formulation script is intended to support the layperson in representing the semantic aspects of a problem according to a prescribed sequence of steps, it can be classified as a *content-based* script (Weinberger, Fischer, & Mandl, 2003) or *content schema* (cf. Ertl, Kopp, & Mandl, this volume). Nevertheless, its primary objective is to facilitate the layperson's communication with the expert. The prompts used in the script are derived from empirical analyses of the deficiencies typical of laypersons' problem descriptions. The question of whether they can successfully compensate for these deficiencies was addressed by an experimental study. The major findings of this study will be reported in this chapter (for a complete account cf. Nückles & Ertelt, 2006).

2. UNFAVORABLE FEATURES OF LAYPERSONS' PROBLEM DESCRIPTIONS

Compared to expert users, laypersons may be in a more difficult situation when seeking advice. Their deficient and fragmentary knowledge makes it hard for them to formulate their queries in a way that would make it possible for the advisors to understand their problem (Allwood, 1986). Alty's and Coombs's (1981) classic analysis of advisory interactions shows that the query is usually presented in a single and brief utterance, which is rarely questioned by the advisor. Rather than providing the advisor with a detailed description concerning the aim of their interaction with the computer and the actions they have so far undertaken to accomplish this aim, lay users prefer to present a particular portion of their problem that often fails to convey its real nature. In order to be brief (cf. maxim of manner; Grice, 1975), a layperson typically fails to provide enough context when presenting their query (Clark & Carlson, 1981), thus making comprehension hard for the expert (e.g., "I don't understand why I haven't got any output; there aren't any error messages"; Alty & Coombs, 1981, p. 29). A layperson often fails to mention key concepts indispensable for comprehending the problem (e.g., which application program or which operation system is the user actually referring to?), or simply assumes that the helpdesk expert is able to see what they see on their screen (e.g., "I have clicked on that button, but nothing happened..."). The task of formulating their problem is cognitively very demanding because the layperson normally lacks the specialist knowledge necessary for generating an adequate representation of the problem. Besides the deficits of providing insufficient context information, it is another frequent drawback that, instead of giving a description of what has happened and what is observable, laypersons tend to present their opinion about the nature or possible solutions of the problem, which is often misleading (Alty & Coombs, 1981; Pollack, 1985).

In face-to-face communication, most of the above mentioned deficits concerning the laypersons' problem descriptions can be compensated by clarification questions the advisor asks in response to the user's initial query (Aaronson & Carroll, 1987). However, such grounding behavior (Clark & Brennan, 1991) can easily be realized in verbal communication, but is less feasible in asynchronous email communication where the opportunity to provide feedback is seriously limited. First, the costs of producing a message, for example, a clarification question, are higher compared with verbal communication, because every message has to be typed on the keyboard (Clark & Brennan, 1991). Second, nonverbal feedback is practically impossible because the communication partners can neither see nor hear one another (lack of visibility and audibility). Third, there is often no set sequentiality of a message and its reply, which makes comprehension harder.

Considering the deficiencies of lay users' problem descriptions on the one hand, and the constraints of asynchronous communication as set out by Clark and Brennan (1991) on the other hand, it is evident that the layperson's initial presentation of the problem is crucial with regard to the effectiveness and the potential success of the advice the expert will be able to offer. The more detailed and comprehensively the laypersons describe their problem, the easier it should be for the expert to correctly diagnose the "real" nature of the problem – and the more effectively the computer expert would be able to help the client. Hence, laypersons who consult a helpdesk expert should be supported in stating their problem as detailed and completely as possible right from the start of the advisory dialogue. The problem description should in particular represent the user's problem as closely as possible so that the expert can infer a complete and coherent mental model of the problem.

3. SUPPORTING A LAYPERSON IN PROVIDING PROBLEM DESCRIPTIONS

How could such a support method operate? It certainly cannot replace the domain-specific problem-solving competence, which the layperson does not possess. Laypersons typically seek advice from a computer expert in order to solve or get solved a concrete computer problem. However, in doing so, laypersons unlike novices usually do not intend to become a computer expert (cf. Patel et al., 1999, and Bromme, Rambow, & Nückles, 2001, for the distinction between the notions of novice and layperson). Hence, the aim cannot be to turn the layperson into a computer expert. However, problem-solving theory distinguishes between so-called *weak* problem-solving strategies, which are domain-independent and *strong* strategies, which are domain-specific (Jonassen, 2000). Typically, strong strategies are used by domain experts. Weak strategies, on the other hand, such as general heuristics like means-ends analysis, are usually part of the everyday problem-solving competences of laypersons (Arlin, 1989). A key element of means-ends analysis is, for example, the comparison between the desired target state and the actual knowledge state. This heuristic is particularly relevant for the formation of a problem representation. The problem solver tries to summarize the actual state of the problem (e.g., "How far have I already come, which are the barriers that prevent me from proceeding?") and formulates the desired goal state ("Where do I want to get to?"). Inasmuch as such a general, that is, domain-independent, heuristic for generating problem representations can be assumed to be part of the metacognitive knowledge of laypersons about everyday problem-solving (Arlin, 1989), it should be possible to support laypersons in applying this heuristic to their description of problems in the computer domain.

According to this rationale, the laypersons should specifically be supported in formulating the goal they want to reach through their interaction with the computer. They should further be encouraged to provide a detailed description of their actual problem state, including, for example, information about the software or the operating system they are working with, and the actions they have so far executed in order to reach the intended goal or solve the problem. Supporting laypersons this way should counteract their inclination to merely present a single portion of their problem as has been observed by Alty and Coombs (1981). Thus, helping laypersons to apply familiar heuristics from everyday problem-solving to the description of their computer problem, should result in more representative and comprehensive problem descriptions, which are easier to reconstruct for the computer expert. This approach might be successful precisely because it makes use of laypersons' preexisting metacognitive knowledge about representing problems. Helping the layperson to conceive of a difficult problem in terms of a familiar scheme might also facilitate learning of how to compose representations of problems in the computer domain.

4. THE PROBLEM FORMULATION SCRIPT

How could such a support method concretely look like? Inasmuch as laypersons can be assumed to possess the relevant metacognitive knowledge necessary for the composition of concise and comprehensive problem descriptions, it seems to be promising to *prompt* them how to proceed in formulating the problem. For this purpose, we provided laypersons with a problem formulation script that comprised several prompts. Each prompt was designed to trigger a different aspect of the problem description. In the context of our study, these prompts can be termed as *strategy activators* (Reigeluth & Stein, 1983) because they were intended to elicit specific problem solving activities that laypersons should in principle be capable of doing but which they do not spontaneously demonstrate, or demonstrate to an unsatisfactory degree (King, 1992; Pressley et al., 1992). We conducted an experimental study in which we tested two different versions of the problem formulation script. Both versions contained the same four prompts: First, the laypersons were asked to be as explicit and detailed as possible about their problem. Second, on a more concrete level, the laypersons were prompted to explain their goal they wanted to accomplish with the computer. Third, they were prompted to list their previous actions and to describe what they actually see on the computer screen. Fourth, they were also encouraged to speculate about a probable cause concerning their failure to accomplish the task. This last prompt was introduced to do justice to the users' inclination to present their inferences about the cause of the problem. However, because the contrast between this prompt and the previous ones made the difference between *inference* and *description* explicit, the tendency of mainly presenting inferences instead of observable facts and actions should lessen accordingly (cf. Figure 5-2).

It should be noted that this approach to support laypersons' problem descriptions by means of a problem formulation script is similar to the way *process worksheets* are used in recent computer-based instructional approaches to guide instruction (cf. van Meeriënboer, 1997). Like the problem formulation script suggested here, process worksheets provide a description of the phases one should go through when solving a problem as well as hints or rules of thumb that may help to successfully complete each phase. However, process worksheets are typically employed to help novice students adopt "strong" domain-specific problem solving strategies (cf. Nadolski, Kirschner, van Meeriënboer, & Hummel, 2001). The script approach suggested here, however, encourages laypersons to apply a weak and domain-independent strategy, which they are familiar with, to their description of problems in the computer domain.

The two versions of the script differed in the *sequencing* of the prompts offered to the layperson (Kollar et al., in press). In the *non-sequenced version*, the prompts were listed in the header of the email form sheet and the layperson was encouraged to start by carefully reading through all prompts and to bear each in mind while composing the problem description. The *sequenced version*, in contrast, required the laypersons to respond to each prompt separately in succession (except for the first prompt which referred to the general *style* of how to write the description rather than to a particular semantic aspect of it). We introduced this experimental distinction regarding the sequencing of the prompts because we wondered whether asking the laypersons to keep the prompts in working memory during text production might demand too much of them. Consider that laypersons are typically not used to describing their computer problems according to the schema outlined above – although they may in principle be capable of doing so. Thus, as the non-sequenced prompting version required the laypersons to keep the prompts in working memory during text production, this additional demand might impair the quality of the descriptions. The sequenced version, in contrast, encouraged the layperson to proceed in a step-by-step fashion and each prompt could be dealt with individually. Hence, only one prompt at a time had to be kept in working memory and no decision was required from the layperson concerning the linearization of the text (cf. Levelt, 1989), that is, the sequence by which the prompts were processed.

5. TESTING THE PROBLEM FORMULATION SCRIPT EXPERIMENTALLY

5.1 Research questions

In our experiment study (cf. Nückles & Ertelt, 2006), we addressed the basic question whether a problem formulation script as outlined above would effectively support laypersons' composition of problem descriptions in the computer domain. In concrete terms, we expected that the script should counteract laypersons' tendency to be too brief ("maxim of manner"; Grice, 1975) and to describe merely a particular portion of the problem (Alty & Coombs, 1981). Consequently, laypersons following the script should produce more extensive problem descriptions compared with laypersons having no script available (*extensiveness prediction*). While *extensiveness* is primarily a quantitative aspect of problem descriptions, it is of course important to show that the script also improves the quality of the descriptions. The previous theoretical discussion suggests that *representativeness* is a central qualitative aspect of problem descriptions. It can be defined as the extent to which a layperson's description reflects her or his actual problem state. Thus, we predicted that prompting the laypersons to report their goal, the actions previously accomplished, as well as what they see on the screen, should particularly improve the representativeness of the problem descriptions compared with a control condition without any prompts available (*representativeness prediction*). Consequently, more extensive and more representative problem descriptions should make it easier for computer experts to reconstruct the layperson's problem from the written description (*quality-of-reconstruction prediction*). This prediction concerning the *quality of reconstruction* is crucial because counseling a layperson in asynchronous communication settings usually implies that the advisor is blind to the client's actual situation and has to reconstruct the client's problem from the client's email.

Beyond the basic question regarding the effectiveness of a problem formulation script, we asked on a more specific level whether the *sequencing* of the prompts would make a difference. In particular, we expected that the *sequenced* prompting version should be more advantageous than the *non-sequenced* version (*sequencing prediction*) because in the sequenced prompting condition the laypersons were encouraged to work off each prompt individually. Accordingly, formulating a problem description should be less demanding because only one prompt at a time had to be kept in working memory and no decision was required concerning the linearization of the text.

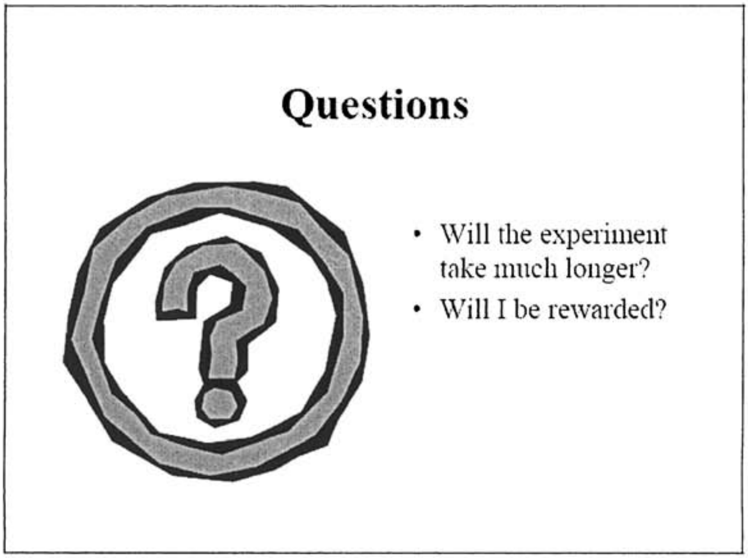
5.2 Participants and research design

Laypersons were recruited among undergraduate students of psychology. Experts were recruited among advanced students of computer science. The participants' expertise status as experts or laypersons was ascertained by a questionnaire that included self-ratings of computer expertise and estimations regarding the frequency of computer software usage. In order to test the effectiveness of our scripting approach, a one-factorial between-subjects design was used with "prompting version" as the independent variable: For the task of writing problem descriptions laypersons received either a) no prompts (*non-prompting condition*), b) prompts *without* a specified sequence (*non-sequenced prompting condition*), or c) prompts *with* a specified sequence (*sequenced prompting condition*). Dependent variables included the *extensiveness* of the problem descriptions as measured by the number of words, their *representativeness* in respect to the layperson's actual problem state, and the *quality of reconstruction*, that is, the extent to which the experts were able to reconstruct the layperson's actual problem from the description of it.

5.3 Materials and procedure

The laypersons worked individually on a personal computer equipped with the application software required for solving several experimental tasks. These tasks covered problems one typically encounters when using common desktop software such as Microsoft Word, Microsoft PowerPoint, Adobe Acrobat Reader and graphics software such as Adobe Photoshop (for an example, see Figure 5-1). The tasks were selected in a preexperiment in which the difficulty for laypersons of a large number of tasks was determined. Only tasks were selected which could not be solved by any of the participants of the preexperiment.

Please try to imitate the following slide. Your copy should look **exactly** like the original. Use the program „Microsoft PowerPoint".



If you can't solve the task, please write an e-mail to the helpdesk expert and describe your problem.

Figure 5-1. Example task of the experiment.

In the experiment, the laypersons tried to solve each of the six tasks one after the other. The maximum time to be spent on a task was 5 minutes. When the time was up, the experimenter asked the participants to prepare an email for the helpdesk and describe their problem so that a computer expert who is unknowledgeable of the participant's problem situation would be able to give advice. Participants were given 10 minutes to finish their problem description email before the next task had to be tackled. For each problem description, a separate email form sheet in Microsoft Outlook format had been prepared. In the non-prompting condition, the text fields of the Microsoft Outlook form sheets were blank.

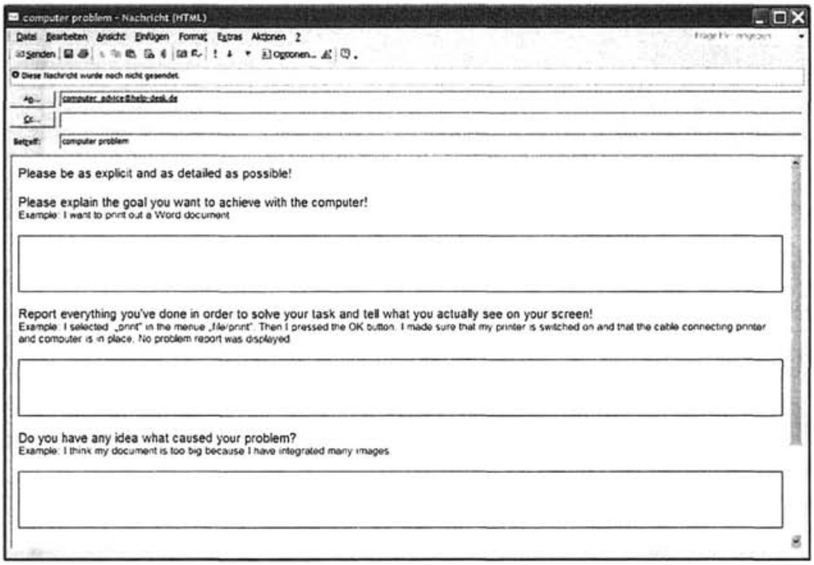


Figure 5-2. Screenshot of the sequenced prompting version.

In the non-sequenced prompting condition, the four problem description prompts were presented in the upper part of the text field. The meaning of each prompt was illustrated by an example except for the general prompt, which emphasized explicitness and detailedness. The prompts were accompanied by an introductory sentence, which asked the participants to carefully read the four prompts and to use them in formulating their problem description. The same four prompts and the accompanying sentence were also used in the sequenced prompting condition. However, the difference was that the sequenced version encouraged the participants to process the promptings separately and one after another. This was accomplished by providing a textbox, directly below each prompt, in which the participants could write their answer. Of course, there was no textbox for the explicitness/detailedness prompt because it referred to the *manner* of writing and not to a specific semantic feature of the problem description such as goal state or actions. Figure 5-2 presents a screenshot of the form sheet used in the sequenced prompting condition.

In the second part of the experiment, the laypersons' problem descriptions were given to computer experts who were ignorant to the tasks the laypersons had tried to solve. Each expert received the problem descriptions of one layperson only because every layperson had treated the same set of computer tasks. This assured that the set of problem descriptions an expert had to evaluate referred to different computer tasks. Thus, the problem descriptions did not overlap or complement each other, which might have considerably facilitated the reconstruction of the underlying task. The prompts in the corresponding experimental conditions were removed from the email form sheets the laypersons had used for delivering their descriptions. Hence, merely the text that the layperson had produced was available to the experts. This was done in order to make the descriptions produced in the prompting conditions comparable to those in the non-prompting condition. The experts were asked to reconstruct the layperson's specific problem from each of the problem descriptions at hand. The instructions told them to write down in complete sentences what they thought the layperson's problem would be and to be as explicit and elaborate as possible in doing so.

Coding. Two blind and independent raters determined the degree to which a layperson's problem description matched the corresponding "objective" reference description on a 5-point rating scale. The quality of the experts' reconstructions was determined in a similar way. The interrater reliability for both rating scales was very good.

6. MAIN FINDINGS

6.1 Test of the extensiveness prediction

The extensiveness of each individual problem description was determined by counting the number of words. To test the *extensiveness prediction*, an a priori contrast was calculated, which compared the mean of the two prompting conditions with the non-prompting condition. The test of this contrast was highly significant and yielded a large effect. Evidently, supplying laypersons with a script how to proceed in describing the problem in fact led them to produce more extensive problem descriptions than laypersons who had no script available.

To examine the sequencing prediction, that is, sequenced prompting results in more extensive descriptions than non-sequenced prompting, another planned contrast was calculated. The analysis showed that the descriptions in the sequenced prompting condition were indeed significantly more extensive than the descriptions in the non-sequenced prompting condition. Nevertheless, the non-sequenced version compared with the non-prompting version substantially raised the extensiveness of laypersons' descriptions as well. Thus, both prompting versions effectively influenced the extensiveness of the laypersons' problem descriptions. The sequenced prompting version, however, which required the laypersons to elaborate on each prompt separately one after the other turned out to be the most successful method.

6.2 Test of the representativeness prediction

To test whether the availability of prompts improved the representativeness of the problem descriptions (*representativeness prediction*), another a priori contrast was computed, which compared the mean of the two prompting versions with the non-prompting version. This contrast was also significant and yielded a large effect, thus showing that the provision of prompts in fact helped the laypersons to produce problem representations that were substantially more representative of the underlying problem than the descriptions of laypersons in the non-prompting version.

Analogous to the previous analysis of the extensiveness scores, the second contrast test showed that the problem descriptions in the sequenced prompting condition were clearly more representative than the descriptions in the non-sequenced prompting condition. However, at the same time, the non-sequenced prompting condition did not significantly differ from the non-prompting condition. Hence, providing laypersons with a script that told them how to proceed in describing their computer problem did not per se enhance the representativeness of the descriptions. Only when the laypersons were encouraged to process the prompts in a prescribed sequence could the prompts unfold their potential to effectively support the layperson's text production.

6.3 Test of the quality-of-reconstruction prediction

Finally, we tested whether the descriptions produced in the prompting conditions facilitated the task for computer experts – ignorant to the computer problems – to reconstruct the layperson's actual computer problem from the mere description of the problem. The planned comparison of the two prompting versions with the non-prompting version clearly confirmed our prediction: Prompted problem descriptions facilitated the reconstruction of the problem compared with non-prompted descriptions. Consistent with the previous results, the effect on the quality of reconstruction was mainly due to the sequenced prompting version. Accordingly, the sequenced prompting condition clearly differed from the non-prompting condition, but there was no significant difference between the non-sequenced prompting condition and the non-prompting condition. All in all, these results underscore the conclusion that simply offering laypersons a script without prescribing the sequence when to process the individual prompts was not sufficient to improve the quality of their problem descriptions. Instead, the prompts had to be processed by the layperson one after the other in order to improve the quality of the descriptions and their comprehensibility for experts who had no direct access to the problem that was described.

7. DISCUSSION

The experimental study provided clear evidence that the problem formulation script effectively supported laypersons in how to describe their problems with the computer. Both the extensiveness and the representativeness prediction were confirmed. Accordingly, the prompted problem descriptions were significantly more extensive and they represented the underlying problem much better compared with descriptions that had not been prompted. Consequently, in line with the quality-of-reconstruction prediction, it was considerably easier for computer experts to reconstruct the problem from the layperson's written description. Evidently, the promptings helped to remedy typical deficiencies of laypersons' problem descriptions. First of all, they counteracted laypersons' tendency to be too brief when presenting the help-desk a problem (cf. Alty & Coombs, 1981). Second, laypersons' descriptions were substantially more representative; thus, the tendency to report only a particular portion of the problem was lessened (cf. Alty & Coombs, 1981). Third, experts who were completely blind to the layperson's problems were much more successful in developing a mental model of the problem from the laypersons' descriptions. Hence, the script apparently supported the laypersons in writing descriptions that were less misleading (cf. Pollack, 1985), less incomplete (e.g., lack of key concepts) and less egocentric with regard to the way they were formulated. It is noteworthy that in the non-prompting version, the match between the reference description of a problem and an expert's reconstruction (i.e., the quality of reconstruction) was 42% on average whereas in the sequenced prompting condition it was raised to almost 68%.

Nevertheless, it has to be emphasized that the prompts were mainly effective when presented in a sequenced version. Although the non-sequenced prompting version raised the extensiveness of the laypersons' problem descriptions, it had practically no effect on the representativeness of the descriptions and, even more importantly, on the experts' ability to reconstruct the problem. Consequently, just asking laypersons to report the goal they want to reach, to tell the actions undertaken so far and their idea of the reason for their failure, did not affect the quality of their text production unless they were encouraged to answer each prompt separately one after another. Asking the laypersons to work off each prompt individually and consecutively evidently facilitated the task to produce representations of problems in a domain where the participants had only a very low level of experience.

It may be speculated that the way the sequenced problem formulation script supported laypersons in describing computer problems is comparable to the way process worksheets guide students' problem-solving activities in computer-based learning environments (cf. Nadolski et al., 2001; van Merriënboer, 1997). Accordingly, it is possible that the sequenced version of the problem formulation script reduced the cognitive load induced by the demand to keep the prompts in working memory during text production (Sweller, van Merriënboer, & Paas, 1998). However, it has to be acknowledged that we did not measure cognitive load in this experimental study. Thus, future research is needed in order to identify the exact cognitive mechanisms that mediated the effectiveness of the sequenced version of the problem formulation script.

What are the broader practical and theoretical implications of this research? Guiding laypersons' problem descriptions by a problem formulation script has proved to be a successful approach to support asynchronous communication between computer experts and laypersons. Interestingly, the research by Alty and Coombs (1981) suggests that the script approach presented here might also be useful to support face-to-face counseling. In most of the conversations they analysed, a stage where the advisor tried to clarify the user's query was lacking (Alty & Coombs, 1981). Thus, given that in face-to-face settings advisors tend to abstain from questioning the clients' presentation of their problem, it seems to be crucial that the clients present their problem as adequately and comprehensively as possible. In order to support the clients' problem descriptions in face-to-face communication, a problem formulation script could be used by the advisor to initiate the advisory dialogue with the client. Accordingly, the promptings could serve expert and client as a *collaboration script* that supports the *presentation* phase of advice-giving dialogues (Alty & Coombs, 1981). On the other hand, inasmuch as the advisors consciously use the prompts to initiate and control the dialogue with the client, they may be stimulated to monitor more carefully their own understanding of the client's problem. Hence, scripting communication between computer experts and laypersons that way may not only support the *presentation* phase but also the *clarification* phase of the advisory dialogue.

Another implication of the script approach presented here refers to the theoretical distinction between effects *with* the script and effects *of* the script (Salomon, 1993). In the present experiment, our main intention was to investigate the effects *with* the script, particularly, whether the availability of the problem formulation script in the email form sheet would facilitate the task of writing more representative and more comprehensive problem descriptions. However, it might be further interesting to investigate the effects *of* the script, for example, whether its availability and continued application triggers the internalization of the script and thereby – on the long run – improves laypersons' ability to generate problem representations in the computer domain (cf. the chapters by King, this volume, and Rummel & Spada, this volume). Hence, experimental settings would be interesting where laypersons' ability to create problem representations is assessed after the promptings have been faded out (cf. Collins et al., 1989). Last but not least, future research should also explore the generalizability of the problem formulation script. While the script presented in this chapter might easily apply to slightly different technical domains, such as electronic devices, supporting laypersons in communication with experts in other knowledge domains seems to be of equal importance. As there is, for example, a growing reliance on health-related information in the Internet, laypersons in this domain could also benefit from improved problem descriptions that allow experts to give more effective and individualized medical advice (see Runde, Bromme, & Jucks, this volume). Future research is needed to investigate this promising avenue to supporting laypersons communication with experts. In conclusion, one can say that laypersons should by no means act as a passive recipient in communication with experts. Rather, it has been our intention to show that despite their lack of domain specific knowledge laypersons can actively contribute to reaching their goal of getting adequate and satisfactory expert advice.

AUTHOR NOTE

The data reported in this chapter was collected by Anna Ertelt as partial fulfillment of the requirements for her Diploma degree at the University of Freiburg. All data were completely reanalyzed in preparation for this book chapter. We would like to thank Kristen Drake for her proofreading as a native speaker.

REFERENCES

Aaronson, A., & Carroll, B. M. (1987). The answer is in the question: A protocol study of intelligent help. *Behaviour and Information Technology, 6*, 393-402.

Allwood, C. M. (1986). Novices on the computer: A review of the literature. *International Journal of Man-Machine Studies, 25*, 633-658.

Alty, J. L., & Coombs, M. J. (1980). Face-to-face guidance of university computer users: A study of advisory services. *International Journal of Man-Machine Studies, 12*, 389-405.

Alty, J. L., & Coombs, M. J. (1981). Communicating with university computer users: A case study. In M. J. Coombs & J. L. Alty (Eds.), *Computing skills and the user interface* (pp. 7-71). London: Academic Press.

Arlin, P. K. (1989). The problem of the problem. In J. D. Sinnott (Ed.), *Everyday problem solving* (pp. 229-237). New York: Praeger.

Bromme, R., Rambow, R., & Nückles, M. (2001). Expertise and estimating what other people know: The influence of professional experience and type of knowledge. *Journal of Experimental Psychology: Applied, 7*, 317-330.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 127-149). Washington, DC: American Psychological Association.

Clark, H. H., & Carlson, T. B. (1981). Context for comprehension. In J. Long & A. Baddeley (Eds.), *Attention and performance* (pp. 313-330). Hillsdale, NJ: Lawrence Erlbaum Associates.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing and mathematics. In L. B. Resnick (Ed.), *Knowing, learning and instruction* (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum Associates.

Coombs, M. J., & Alty, J. L. (1980). Face-to-face guidance of university computer users: Characterizing advisory interactions. *International Journal of Man-Machine Studies, 12*, 407-429.

Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and semantics* (pp. 41-58). New York: Academic Press.

Jonassen, D. H. (2000). Toward a meta-theory of problem solving. *Educational Technology: Research & Development, 48*, 63-85.

Kiesler, S., Zdaniuk, B., Lundmark, V., & Kraut, R. (2000). Troubles with the Internet: The dynamics of help at home. *Human-Computer Interaction, 15*, 323-351.

King, A. (1992). Comparison of self-questioning, summarizing, and notetaking-review as strategies for learning from lectures. *American Educational Research Journal, 29*, 303-323.

Kollar, I., Fischer, F., & Hesse, F. W. (in press). Collaboration scripts – a conceptual analysis. *Educational Psychology Review.*

Levelt, W. J. M. (1989). *Speaking. From intention to articulation.* Cambridge: MIT Press.

Nadolski, R. J., Kirschner, P. A., van Merriënboer, J. J. G., & Hummel, H. G. K. (2001). A model for optimizing step size of learning tasks in competency-based multimedia practicals. *Educational Technology Research and Development, 49*, 87-103.

Nückles, M., & Ertelt, A. (2006). The problem of describing a problem: Supporting laypersons in presenting their queries to the Internet-based helpdesk. *International Journal of Human Computer Studies, 64*, 648-669.

Nückles, M., & Stürz, A. (2006). The assessment tool. A method to support asynchronous communication between computer experts and laypersons. *Computers in Human Behavior, 22*, 917-940.

Nückles, M., Wittwer, J., & Renkl, A. (2005). Information about a layperson's knowledge supports experts in giving effective and efficient online advice to laypersons. *Journal of Experimental Psychology: Applied, 11*, 219-236.

Pollack, M. (1985). Information sought and information provided: An empirical study of user/expert dialogues. In *Proceedings CHI'85: Human Factors in Computer Systems* (pp. 155-159). San Francisco, CA: April 14-18. New York: Association for Computing Machinery.

Pressley, M., Wood, E., Woloshyn, V. E., Martin, V., King, A., & Menke, D. (1992). Encouraging mindful use of prior knowledge: Attempting to construct explanatory answers facilitates learning. *Educational Psychologist, 27*, 91-109.

Raskutti, B., & Zukerman, I. (1997). Generating queries and replies during information seeking activities. *International Journal of Human-Computer Studies, 47*, 689-734.

Reigeluth, C. M., & Stein, F. S. (1983). The elaboration theory of instruction. In C. M. Reigeluth (Ed.), *Instructional-Design theories and models: An overview of their current status* (pp. 335-382). Hillsdale, NJ: Lawrence Erlbaum Associates.

Salomon, G. (1993). No distribution without individuals' cognition: A dynamic interactional view. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations* (pp. 110-138). Cambridge: Cambridge University Press.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Sinnott, J. D. (1989). *Everyday problem solving. Theory and applications.* New York: Praeger.

Süß, D. (2001). Computers and the Internet in school: Closing the knowledge gap? In S. Livingstone & M. Bovill (Eds.), *Children and their changing media environment: A European comparative study* (pp. 221-241). Mahwah, NJ: Lawrence Erlbaum Associates.

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

Van Merriënboer, J. J. G. (1997). *Training complex cognitive skills.* Englewood Cliffs, NJ: Educational Technology Publications.

Weinberger, A., Fischer, F., & Mandl, H. (2003). Gemeinsame Wissenskonstruktion in computervermittelter Kommunikation: Wirkungen von Kooperationsskripts auf den Erwerb anwendungsorientierten Wissens [Cooperative construction of knowledge in net-based communication: Effects of cooperation scripts on the acquisition of application-oriented knowledge]. *Zeitschrift für Psychologie, 211*, 86-97.

Wirth, J., & Klieme, E. (2002). Computer literacy im Vergleich zwischen Nationen, Schulformen und Geschlechtern [Computer literacy in comparison between nations, school-type and gender]. *Unterrichtswissenschaft, 30*, 136-157.

Chapter 6 – Discussion

BEING TOLD TO DO SOMETHING OR JUST BEING AWARE OF SOMETHING? AN ALTERNATIVE APPROACH TO SCRIPTING IN CSCL

Friedrich W. Hesse
Knowledge Media Research Center, Tübingen

Abstract: It might be easily accepted that learning and understanding can be positively influenced by some forms of social interaction. In the past, different aspects of social interaction have already been proposed for this purpose. The concrete form of social interaction can be communicated or even enforced by providing a script or might be subject to an individual process of self-guidance which is based on being well informed about the partner, the situation, and the development of the ongoing social interaction. Some chapters of this book as well as this one address these alternatives.

In this chapter, I will try to go beyond a mere summary of the contributions in this part of the book. What follows can rather be characterized as a reflection on very basic alternative strategies for influencing computer-supported collaborative learning. However, before explaining the essence of these alternative strategies and discussing how the studies reported above actually make use of them, I would like to bring to mind some premises on which they are built. In the following I try to sketch a rough framework that outlines the underlying rationales.

What has to be kept in mind first is that from the perspective taken here, *learning* is the final goal of the activity and all other accomplishments, above all good collaboration, are only means to this end. This has to be stressed explicitly in order to differentiate this perspective from others that view collaboration as an end in itself. This is another strand of research addressing the fact that many activities in education and work life build upon the ability to cooperate and collaborate effectively. In particular, members of collaborative dyads or teams have to make sure that they understand each other quite well, know enough about each other, each other's prior knowledge and way of understanding (mental representations etc.). Clark's (1996) interest in the "common ground" that is built when partners communicate has, for instance, proven to be a very fecund approach to research preconditions for efficient collaboration (see also Clark & Marshall, 1981). CSCL research has taken up this line of thought by investigating the support and constraints that computers can bring into collaborative scenarios. Thus, collaboration as a more generic topic is addressed as well as the special case of having the support of a computer. This line of research includes studies about shared knowledge distributed over people as well as over technical databases.

On the other hand, research focusing more exclusively on learning regards collaboration as a mere vehicle to make cognitive processes of learning and knowledge building more explicit, more reflective, and more structured – which might otherwise not occur or only in a less intensive way. As a vehicle to improve learning and knowledge processes, slightly different forms of collaboration might be required, which raises again the question of how to influence the process of collaborating and thus of learning appropriately. Again, this generally applies to situations both with and without computer support. But not in the same way: if a computer is available, other forms or conditions of collaboration have to be taken into account. Considering that computer-mediated collaboration might be an extra task and cause extra cognitive load, the goal to benefit from using the computer might not be easy to reach in the context of CSCL.

What we know up to now is thus that fostering interaction for the sake of learning might mean something different than fostering interaction in its own right and that, additionally, anything we can say about the matter must be differentiated further as soon as computers are involved. But before turning to the issue of computer support, let's consider for a moment the underlying premise of why collaboration or, more generically, social interaction is at all relevant to learning.

The very idea that social interaction and learning or knowledge processes are genuinely linked is quite widespread and shows up in quite different forms. An instance is Hutchins' (1995) single-system perspective which considers interacting persons, their interactions and objects (artifacts) as a single socio-technological system. In this framework, individual actions are always seen from the angle of the whole unit. Another perspective which also considers a group (and its technical possibilities to store information) acting as a whole is Wegner's (1987, 1995) transactive memory approach where someone just knows where something can be found, be it in a technical database or in the knowledge base of other persons. Subsequently, persons can collaborate in a way allowing knowledge to be stored in different persons or technical sources under the condition it will be accessible in the future.

Turning to the classics of education theory, it is Vygotsky (1978) who points out that even processes happening within a single person are always a consequence of inter-psychological social processes. One interpretation of his zone of proximal development looks at the difference in the state of knowledge of an individual learner to that of a more advanced other person (e.g., a teacher). The benefits of the diversity which social interaction entails are also emphasized in the tradition of Piaget (Piaget & Inhelder, 1969; Doise & Mugny, 1984). Based on his assimilation and accommodation approach, it is assumed that a cognitive conflict comes up more often in a social situation and can cause active and reflective ways of solving the conflict which then can lead to deeper understanding and better learning.

Social interaction as a means of deeper elaboration has also been the main idea of O'Donnell and Dansereau's (1992) "scripted cooperation" approach, which leads us to the more detailed ideas of *how exactly* social interaction must be designed to result in better learning. To achieve this goal, two learners are shifting the role of explaining (summarizing and giving examples) and listening (giving feedback in form of asking for clarification).

Collins, Brown, and Newman (1989) and Palincsar and Brown (1984) are not part of the tradition of scripted cooperation, but develop somewhat similar approaches by also starting from a social situation and determining who does what – this time by assigning roles. One could even call these approaches special forms of scripting. The Cognitive Apprenticeship approach by Collins et al. starts with the social interaction between a tutor and an apprentice and is followed by a script dealing with the collaboration between the two when following the sequence of modeling, coaching, and scaffolding. In each step there is a clear division of labor which serves the purpose of teaching by the tutor and learning by the apprentice. The "Reciprocal Teaching" approach by Palincsar and Brown is mainly applied to text understanding but follows a similar idea concerning social interaction to support learning and understanding and offers a similar script as Collins et al.

Up to now, one can assume that there is a benefit when becoming engaged in some structured forms of social interaction with the purpose of learning and further knowledge processing. However, we did not explicitly discuss the use of *computer-supported* ways of collaboration. In general, this will bring up the question if things are just similar or different, better or worse when we use computer support. There might be no single general answer to it as it depends on the concrete form of social interaction. However, there are some features of computer support which, if brought to bear, can be beneficial for collaborative learning. To a certain extent, these features have to do with overcoming space and time limitations. But there's more to it than just simply being at different places at different times when collaborating. Especially having control over the time dimension allows to stop a process, trace it back and revise something and thus become more reflective. Furthermore, face-to-face communication and collaboration without any technical means does not allow for having permanent protocols to refer to or to read again. This also might lead people to write better structured and organized messages because they know they are permanent. And, finally, computer-mediated scenarios offer opportunities for guiding and influencing the interaction that face-to-face cannot provide.

This leads directly to considerations about how one can make sure that, by using the means of a computer, collaboration is beneficial for learning and understanding. At this point, two different strategies could be taken. Strategy I (which I refer to as the 'scripting approach') assumes that the designer has enough and solid knowledge about what is the right way, the best procedure, the best support to guide an output-oriented collaboration. If this is true, it would be possible either to tell or instruct subjects how to proceed or to design a computer-based environment in a way that certain procedures are enforced or at least elicited. Subjects are thus guided or scripted to follow the right procedure.

This strategy has to face a couple of possible drawbacks:

- The "right way" might be wrong
- The "right way" might be quite different for different persons and different conditions/situations
- The "right way" might be a good advice only for a very limited time, e.g., in the very beginning
- The "right way" might lead to less motivating activities
- The "right way" asks for following certain rules and is thus adding to the cognitive load of the learner and distracting from the "real task" to be done.

What could be an alternative to scripted cooperation? Just not scripting the cooperation? In CSCL, this doesn't seem to be an option as the need for a coordinating structure appears to be even more urgent in computer-mediated settings than in face-to-face ones. In that respect, computer mediated communication is deficient as compared to face-to-face communication. So there is a need to compensate for it. However, the compensating features implemented in the computer environment at the same time offer a potential for influencing interactions that face-to-face situations cannot provide. Both the 'scripting approach' and the alternative strategy which I will present in the following exploit this unique potential, but they do it in different ways.

Strategy II, which I will call the "awareness approach", differs from the scripted cooperation approach in strategy I. Awareness might be defined as the perception of or knowledge about situational affordances (Buder &

Bodemer, 2005), or as simply "knowing what is going on" (Endsley, 1995, p. 36).

This approach does not deliver explicit instructions, but instead enriches the available information about the group, participation of group members, activities, and e.g., even interest of the collaborators. Strategy II relies completely on making relevant features of the collaborators and activities "aware" and expecting that the collaborators either know themselves how to proceed or are able to develop a good way of collaboration by themselves (Gutwin & Greenberg, 2002).

Awareness also has to take into account that, in order to avoid cognitive overload, only that information should be made available/aware which can be helpful to adequately organize one's own proceeding. This of course is based on a difficult decision to be taken. The technology itself might be much more powerful than human senses are in a face-to-face setting. Thus, in principle, it would be possible to deliver a broad range of measures as there is e.g.

- group awareness (who is around, who is active and participating and to which extent),
- situation awareness (where one is located, how the conditions are and what a task looks like),
- history awareness (how have things developed, what has been done before).

For other forms or taxonomies of awareness see Gross, Stary, and Totter (submitted), Gutwin and Greenberg (2002), Jermann, Soller, and Mühlenbrock (2001), Carroll, Neale, Isenhour, Rosson, and McCrickard (2003).

Publishing a book about "Scripting Computer-Supported Collaborative Learning" has to reason about these two strategies. This book does it and especially in this section. All four chapters address scripting as well as becoming self-regulated as it is needed if the learner is not told, but just being aware of something. In the following overview, I will not highlight in which ways scripting is addressed in these chapters. To change the perspective, I reveal the presence of the second aspect in each of them, that is: awareness and its effects.

The first chapter by Alison King about "Scripting Collaborative Learning Process: A Cognitive Perspective" has a very clear understanding about the purpose of scripting as she reflects explicitly what cognitive, meta-cognitive, and socio-cognitive processes are intended to be supported. So well known cognitive activities like e.g., repetition, rehearsal, retelling, summarizing, and paraphrasing are addressed. Even more complex activities like analogical thinking, integration of ideas and reasoning are part of the intended cognitive activities. However, above all these very reasonable cognitive activities she is questioning to what degree these activities can be accomplished by the learner alone – without social interaction – and to what degree scripting can be turned into self-regulation.

The second chapter in this section by Nikol Rummel and Hans Spada about the question if "People Can Learn Computer-mediated Collaboration By Following A Script" tries to compare the script approach with a model approach. Interestingly, having a model to observe does lead to the same advantages as explicit scripting. Obviously people can derive on their own how to process without an explicit guidance. Some of the questions discussed by the authors of this study were concerned with how the motivational situation might be when being scripted. They did not say, the motivation is better when only getting a model to see, but this could be an interesting speculation. A model approach is not directly something we might call an awareness approach but it equally relies on having partners being able to decide on their own how to proceed instead of being scripted.

"Scripting In Net-based Medical Consultation: The Impact Of External Representations On Giving Advice And Explanations" by Anne Runde, Rainer Bromme, and Regina Jucks in the next chapter conceives scripting more in the sense of representational guidance and thus as a form of implicit scripting. One could even interpret their approach rather as a form of making something aware as of scripting. If there is scripting then it is more self-developed and self-regulated. Their main expectation has been that the information depicted in the concept map used in this study helps to focus the communication more strictly to the depicted content and terms. These results support the idea of having this special form of "implicitly scripting" as they call it even if it might be more in the tradition of making something aware as scripting someone's behavior.

"Scripting Laypersons' Problem Descriptions In Internet-based Communication With Experts" by Matthias Nückles, Anna Ertelt, Jörg Wittwer, and Alexander Renkl is based on the idea that the dialogue partners can profit from following a problem formulation script. Probably two mechanisms have been effective in their study. One mechanism can be seen in having a "template" available about what constitutes a complete problem description. One of their results was due to having this more complete description. Additionally, the authors point explicitly to the effect of sequencing the steps to lower the cognitive load in developing a problem description. Insofar, we are close to the awareness perspective if one refers to a shared understanding what constitutes a complete problem description by having a template for it available. However, we are beyond this perspective if one looks at the effects of *sequencing* which is a quite original feature of scripting.

Getting back to the title of this chapter, we now can state that what I referred to as the 'awareness approach' can indeed be regarded as an alternative to scripting. When reflecting on critical aspects of scripting – as scripting might be e.g., too rigid or lead to motivational problems – the chapters in this section often come up with alternatives which contain a lot of aspects of the awareness approach. Neither the four chapters before nor this chapter can definitely answer if one approach is superior to the other.

However, scripting approaches are facing their limits, whereas the awareness approach seems to have some potential which has not been tried out enough. Perhaps future research might even plead for the two of them – scripting and awareness features – in a balanced combination.

REFERENCES

Buder, J., & Bodemer, D. (2005). Augmented group awareness tools for collaborative learning: In T. Okamoto, D. Albert, T. Honda, & F.W. Hesse (Eds.), *The 2nd Joint Workshop of Cognition and Learning through Media-Communication for advanced e-Learning (JWCL)* (pp. 34-39). September 28-30, 2005, Tokyo, Japan.

Carroll, J. M., Neale, D. C., Isenhour, P. L., Rosson, M. B., & McCrickard, D. S. (2003). Notification and awareness: Synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies, 58 (5)*, 605-632.

Clark, H. H. (1996). *Using language*. Cambridge: Cambridge University Press.

Clark, H. H., & Marshall, C. E. (1981). Definite reference and mutual knowledge. In A. K. Joshi, B. L. Webber, & I. A. Sag (Eds.), *Elements of discourse understanding* (pp. 10-63). New York: Cambridge University Press.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction. Essays in the honour of Robert Glaser* (pp. 453-494). Hillsdale: Erlbaum.

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

Endsley, M. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors, 37*, 32-64.

Gross, C., Stary, C., & Totter, A. (submitted). User-centered awareness in computer-supported cooperative work-systems: structured embedding of findings from social sciences. *International Journal of Human-Computer Interaction*.

Gutwin, C., & Greenberg, S. (2002). A descriptive framework for workspace awareness for real-time groupware. *Computer Supported Cooperative Work, 11*, 411-446.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge: MIT Press.

Jermann, P., Soller, A., & Mühlenbrock, M. (2001). From mirroring to guiding: A review of the state of art technology for supporting collaborative learning. In P. Dillenbourg, A. Eurelings, & Kai Hakkarainen (Eds.), *Proceedings of the European Conference on Computer-Supported Collaborative Learning* (EuroCSCL-2001) (pp. 324-331). Maastricht, The Netherlands, March.

O'Donnell, A. M., & Dansereau, D. F. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interaction in cooperative groups: The theoretical anatomy of group learning* (pp. 121-140). New York: Cambridge University Press.

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 1*, 117-175.

Piaget, J., & Inhelder, B. (1969). *The psychology of the child*. NY: Basic Books.

Vygotsky, L. S. (1978). *Mind in society*. Cambridge, MA: Harvard University Press.
Wegner, D. M. (1987). Transactive memory: A contemporary analysis of the group mind. In B. Mullen & G. R. Goethals (Eds.), *Theories of group behavior* (pp. 185-208). New York: Springer.
Wegner, D. M. (1995). A computer network model of human transactive memory. *Social Cognition*, *13*, 319-339.

PART II

COMPUTATIONAL PERSPECTIVES

Chapter 7

SCRIPTING COLLABORATIVE LEARNING IN AGENT-BASED SYSTEMS

Gerardo Ayala
Universidad de las Américas-Puebla

Abstract: The chapter discusses an agent-based perspective of scripting for CSCL environments. It presents two approaches for supporting scripting collaborative learning in agent-based systems: (1) on the macro-level of collaborative learning, where agents may support the organization of the learning process through facilitating group configuration and task selection; and (2) on the micro-level of collaborative learning, where agents may support awareness and coordination of activities. Both approaches are presented for two kinds of domains: (i) pedagogically structured and (ii) not pedagogically structured domains. For both cases, the types of support on the macro and micro levels are examined.

1. INTRODUCTION

Cooperation scripts can be designed for organizational processes, at the macro level, and for detailed work processes, at the micro level. Both approaches have their differences when we have a pedagogically structured knowledge domain or a non structured knowledge domain. These two approaches are complementary; at the macro level, scripts support the structure of the collaborative process in order to promote productive interactions, and at the micro level by coordinating the collaboration. This chapter discusses the role of software agents for these two approaches, from the computer science perspective of scripting for CSCL environments, supporting communication, cooperation and coordination, which are the fundamental issues for effective collaborative learning.

From the computer science perspective of scripting, cooperation scripts are integrated in the components of the learning environment and may be *imposed* or *induced*. An imposed cooperation script is presented explicitly to the learners, who have to carry out a set of activities in a specific order. It may cause in the learner a loss in motivation due to a loss of autonomy in the learning activity. An induced cooperation script is embedded in the design of the learning environment, and provides learners with a high amount of freedom, but it is based on the assumption that learners have an *internal culture-acquired* cooperation script, and are aware of the learning opportunities and benefits of collaboration.

Following an imposed script implies a coercion degree, which is the degree of freedom that the learners have in following the script (Dillenbourg, 2002). It is reported in this volume, by Lauer and Trahasch, that, for adult learners, a high degree of coercion might affect motivation. Also it is believed that scripts in a CSCL environment increase the cognitive load of the learner and have the risk to make the groups interact in a non natural way (Dillenbourg, 2002).

1.1 Software agents and cooperation scripts

One of the benefits of implementing software agents is that they can release the cognitive load of the user. The semiautonomous nature of interaction between the learners and software agents provides a low coercion degree, providing one step towards a shift from paternalism to autonomy in inducing cooperation scripts, as proposed by Runde, Bromme, and Jucks in this volume. From this perspective of semi autonomy in user-agent interaction (Norman, 1994) a software agent presents proposals and the user decides among those.

Considering the risk that cooperation scripting in CSCL environments can lead us away from the genuine path of collaborative learning (Dillenbourg, 2002) we believe that the role of agents should be to induce collaborative scripts that regulate collaborative learning without interfering with the social dynamics of the group.

Software agents must support collaborative scripts that are simple to follow and easy to adopt. Therefore, the role of software agents supporting scripting in CSCL environments should be:

- Work on behalf of the learner in order to reduce her cognitive load while she follows a cooperation script.
- Distribute the coercion load over the interaction, coordination and task levels, maintaining a low coercion degree by inducing the appropriate collaborative interaction patterns in the learners.
- Keep the learner aware of activities, resources and the collaboration opportunities by following a cooperation script in the activities of a collaborative learning task.

1.2 Pedagogically and not pedagogically structured domains

The nature and representation of the domain knowledge plays an important role in the design of the cooperation script and the modelling of the agent. We may have a *pedagogically structured domain*, which is a knowledge repository that consists of identified knowledge elements organized in a pedagogical structure (i.e., a knowledge base). On the other hand, we may have a *non pedagogically structured domain*, which is a repository of digital documents organized in a taxonomy but not necessarily with a pedagogical structure (i.e., a digital library).

1.3 Organizational and detailed work processes

The organizational processes that we consider in this chapter, for cooperation scripts at the macro level, are *group configuration, learning plans* and *tasks assignment*. The detailed work processes for cooperation scripts at the micro level are *coordination* and the *social construction of knowledge*. While the organizational processes of group configuration and task assignment are induced scripts, implemented with proposals from the software agent, the scripts for the coordination, for collaborative problem solving and the social construction of knowledge, are imposed scripts.

2. SCRIPTING ORGANIZATIONAL PROCESSES

This section presents the considerations to model software agents supporting collaborative scripting for organizational processes. Because of the social nature of these processes and the characteristics of an agent, it is more feasible that software agents induce the script. For a CSCL environment, software agents can be modeled in order to induce a cooperation script that allows the learners to:

1. Make an appropriate group configuration, based on the capabilities and learning interests of the learners.
2. Assign learning tasks that ensure the existence of zones of proximal development (Vygotsky, 1978) and maintain the productivity of the learner in the community or group.

This requires a learner model, as a set of beliefs the agent has about its learner. For pedagogically structured domains, software agents can propose groups and determine the zones of proximal development of the users by keeping a representation of the learners capabilities in the learner model. In the case of not pedagogically structured domains, software agents propose discussion groups and individual learning plans to the users, by keeping a representation of the interests of the community members in the learner model, together with a record of those popular and relevant digital documents for the community.

2.1 Designing a cooperation script for the organizational processes

For the organizational processes, the attributes of the script are group configuration and task assignment. In general terms, the core mechanism, at this macro level, is to maintain learning opportunities for the learners.

For pedagogically structured domains, in order to induce a cooperation script for task assignment, the software agent requires a learner model, considered as a set of beliefs the agent has about the capabilities of the user, based on her application of the knowledge elements in the domain knowledge. With this information the agent is able to propose learning tasks that generate zones of proximal development for all the learners in the group. For non pedagogically structured domains, the software agent maintains a learner model as a set of beliefs about her interests, in order to propose discussion groups of people with common interests and to keep an individual learning plan.

The design of the interface implies spaces for group configurations, access to open learner models, a space for a learning task proposal, and for the establishment and communication of commitments between learners.

2.2 GRACILE and CASSIEL

In GRACILE, a collaborative learning environment for Japanese grammar, we have a domain knowledge base representing the grammar rules (Ayala & Yano, 1995). The rules are considered knowledge elements and are organized in a pedagogical structure that relates them, according to their internal structure, components, complexity and use. Cooperation scripts in an environment like GRACILE can be defined for group configuration and task assignment.

For GRACILE we implemented a *mediator agent* that assists the learners in the group configuration and in maintaining zones of proximal development in the group (Ayala & Yano, 1996a, 1996b). Using a learner model the mediator agent is able to determine a *structural knowledge frontier*, as a set of knowledge elements pedagogically related to those the agent believes the learner already internalized. In order to determine the learner's zone of proximal development, the agents cooperate and determine a *social knowl-*

edge frontier of the learner, which is defined as the set of knowledge elements the agent believes have been internalized by other members of the learning group. The mediator agent uses the knowledge frontier in order to generate group proposals with zones of proximal development, and task assignments determined to keep the existence of zones of proximal development in the group members.

As an example of an agent based learning environment with a non pedagogically structured knowledge domain, we have developed CASSIEL. CASSIEL was designed based on concepts of lifelong learning and web based repositories, for virtual communities of practice (Ayala, 2002). Cooperation scripts in environments like CASSIEL can be defined for the configuration of discussion groups and the maintenance of an individual learning plan.

The theoretical foundation of collaborative knowledge construction in CASSIEL is the theory of Nonaka and Takeuchi (1995) from the area of knowledge management. Nonaka and Takeuchi (1995) have proposed a theory of the creation of knowledge that has been applied for learning organizations (Morabito, Sack, & Bhate, 1999) and, in general terms, proposes that knowledge in an organization is constructed through the phases of *socialization*, *externalization*, *combination* and *internalization*.

Our interpretation of that theory is as follows: The individual's personal ideas are shared (socialization). When these ideas become of the interest of other participants they are justified and formalized (externalization) becoming shared beliefs, which once validated by the group (combination) are considered new knowledge in the community. Those new knowledge resources in the repository are promoted and, with their use, become knowledge of the participants (internalization). Internalized knowledge plays a role in the generation of new ideas, and the process begins again (Ayala, 2003).

In CASSIEL, a *user agent* maintains a learner model as a set of beliefs about the interests of the learner. We consider that the interests of the learners are the basic issue for successful collaborative learning and interaction in a virtual community based on a digital collection of educational resources (Ayala & Paredes, 2003). The user agent supports collaboration and adaptability, maintaining the model of the learner and assisting her in the configuration of discussion groups. The user agent is designed to help the user to maintain a personalized learning plan, by maintaining a list of resources considered of the interest of the learner as well as of the interests of other participants in the community. The learning plan represents also those resources considered popular in the community, relevant for the learner or new. Following a learning plan, the learner maintains her membership in the community, being aware of what is going on and being able to participate in a more productive way.

2.3 Group configuration

The learner model is the key element for the group configuration proposal generated by the agent. For pedagogically structured domains, the learner model is revised any time the learner makes a right or a wrong construction or answer that implies the application of a knowledge element. For non pedagogically structured domains, the learner model is mainly a representation of the interests of the learner, which are inferred from the navigation of the user in a digital repository, her bookmarks and annotations.

In an agent based CSCL environment, a group configuration should be generated as a proposal from the software agent to the learner. Such proposal must be constructed considering the beliefs of the agent about the capabilities and learning interests of the learners. The agent works on behalf of the user by proposing a selection of participants among all members of the community that have the potential to collaborate with her, and provides information concerning other learners. The group configuration may be seen as an imposed script or an induced one, if it is negotiable.

In the case of a pedagogically structured domain, as in GRACILE, the mediator agent holds a learner model as a set of beliefs organized as follows:

- The learner's capabilities
- The learner's goals
- The opportunities of assistance to the learner from other learners in the current group, generated based on the beliefs of other mediator agents about the capabilities of their learners.
- The registration of the knowledge elements applied in the tasks selected by the learner, so the agent can infer the selection criteria of the learner: selecting learning tasks based on the feasibility (pedagogically related with those already learned), popularity (already learned by others) or relevance (importance for advancing in the domain) of the knowledge elements with respect to the group. This information is useful for constructing a group proposal.

Learner modeling in GRACILE was based on Vygotsky's theory of social learning (Vygotsky, 1978). We represent the learner's actual development level as the set of knowledge elements which the mediator agent believes can be applied by the learner without any assistance. The learner's potential development level is represented by the knowledge used by the learner with the assistance of other learners or from the domain agents. Vygotsky defined the zone of proximal development as the distance between the actual and the potential development level of the learner and it is considered the space of knowledge elements with more possibilities to be internalized by the learner.

In order to propose a group configuration the agent needs to represent the learner's assistance and learning opportunities in a CSCL environment. We have defined the learner's *group-based knowledge frontier* (here after referred as GBKF) (Ayala & Yano, 1996a) as the *union* of the following two sets:

1. *Structural knowledge frontier*: the set of complex domain knowledge elements related to simpler elements believed to be already internalized by the learner.
2. *Social knowledge frontier*: the set of domain knowledge elements believed to be internalized by the members of the current learning group but still not believed to be internalized by the learner.

The learner's *candidate knowledge for relevant collaboration* (hereafter referred as CKRC) consists of the *intersection* of these two sets. The CKRC is then a subset of the GBKF which represents those still not internalized knowledge elements that other learners have internalized and which are structurally related to the learner's internalized elements.

In order to construct the GBKF and the CKRC the mediator agents cooperate by exchanging their beliefs about their learners' capabilities. Upon request the mediator agent informs other mediator agents in the network about the changes in its beliefs about the capabilities of its learner (Ayala, 1996a).

According to our results the mediator agent should, if possible, propose heterogeneous small groups (4 participants) formed by 2 advanced learners, 1 novice and 1 intermediate level learner (Ayala & Yano, 1997). The mediator agent will present a group proposal configured by two advanced learners who have learning goals corresponding to knowledge elements not necessarily pedagogically related with those already internalized (leadership), one intermediate level learner that has learning goals corresponding to knowledge elements pedagogically related with those already internalized (criteria of feasibility), and a novice learner that may have learning goals by any criteria (popularity, feasibility or leadership).

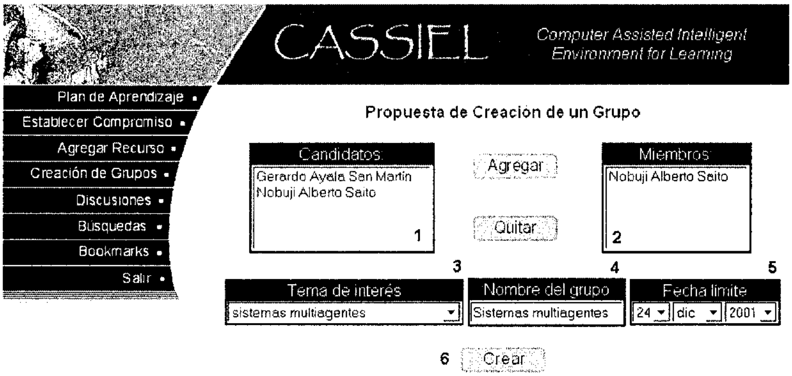


Figure 7-1. The user agent in CASSIEL proposes its learner the establishment of a discussion group. 1: list of candidates; 2: list of selected peers; 3: topic of interest; 4: name of the group; 5: deadline to respond to the invitation; and 6: create and send the group proposal to the candidates.

For non pedagogically structured domains, the group configuration is based on the interests of the learners. The user agent in CASSIEL maintains a learner model based on the interests of the learners, considering the taxonomy of a repository of digital documents. The user agent supports the knowledge socialization phase by proposing the configuration of groups for the exchange of new ideas with those members believed to share common interests with her. The members of the community which are believed to have similar interests to the learner's are shown in a proposal list (see Figure 7-1). The learner selects from them those members to be invited to the discussion group. They will receive an invitation via an email message. They can accept or refuse to participate.

2.4 Task assignment

The approach for a pedagogically structured domain of an interaction script implies task assignment in the joint problem to be solved by the group. An intelligent task assignment by the agent is necessary in order to ensure collaborative learning opportunities for the participants.

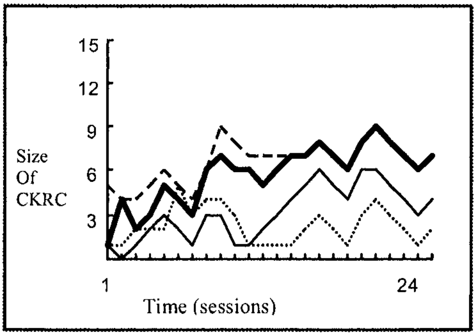


Figure 7-2. Maintenance of zones of proximal development (Candidate Knowledge for Relevant Collaboration, CKRC) by conforming an ideal group by the mediator agents in GRACILE. (Ayala & Yano, 1996b)

In the case of GRACILE each mediator agent proposes to its learner those tasks where knowledge elements in the CKRC set are applied. This results in the enhancement of her assistance opportunities in the group, promoting the creation of zones of proximal development within which she can work and be assisted by more experienced learners in the group. Figure 7-2 presents the results of intelligent task assignments that maintain the zones of proximal development for a group that was configured as described in the previous section. Each one of the four lines refers to a group member. During 24 sessions (horizontal axis), the mediator agents cooperate in order to maintain the size of the CKRC for their learners, by proposing tasks that imply the maintenance of the social knowledge frontier. In this way, all group members have collaborative learning opportunities and are motivated to participate.

In the case of non pedagogically structured domains, as in CASSIEL, a user agent supports the knowledge socialization and the knowledge internalization phases in the knowledge creation process, supporting social awareness and concept awareness. Social awareness is necessary for socialization, and it is provided by the communication of the interests, intentions and capabilities of the members of the community. Keeping social awareness requires an appropriate configuration of groups with people who share interest. Concept awareness is provided by information concerning the new, relevant and popular resources in the repository, so the learner is invited to make annotations and discuss their content with the community. It allows the reflection of the learner about her current level with respect to those members with the same interests. Concept awareness is necessary for maintaining a learning plan and therefore the competitive advantage of the learner in a lifelong learning context.

3. AGENTS SUPPORTING SCRIPTING DETAILED WORK PROCESSES

This section presents the considerations for modeling software agents that support collaborative scripting as a detailed work process, at the micro level. The detailed work process that we consider here is the learners' collaboration or cooperation. While the organizational processes of group configuration and task assignment are based on proposals from the software agent, the script for the coordination of collaborative problem solving can be seen as an imposed one, complemented by the necessary awareness. In order to follow a script for the collaboration process, the software agent requires to make its learner aware of the collaborative process in the joint problem, and also aware of the collaboration opportunities in the group.

For a pedagogically structured knowledge domain, we present how software agents can be modeled in order to support a cooperation script for learners' coordination and make the learners aware of their collaboration opportunities in the group. For a non pedagogically structured knowledge domain, software agents can support the coordination of activities in the construction of knowledge in a community of practice, based on a digital repository in the web. These activities are the establishment of a digital resource, the recommendation of a resource, provide annotations, reject a resource, to establish relations between resources in the repository, and collaboratively organizing the repository.

3.1 Agents supporting the detailed work processes

The workspace in GRACILE, as a CSCL environment for Japanese grammar and expressions, is a dialogue to be constructed by the group, applying those knowledge elements that correspond to grammar rules that refer to the learner's zones of proximal development in the group (Ayala & Yano, 1996a).

In GRACILE the joint problem is the construction of a dialogue, as a collaborative learning activity where learners share knowledge helping them to act and to understand sequences in specific situations, as speech acts. Each learner is committed to the group, by the task assignment, to construct a sentence for the dialogue, which must correspond to a speech act (requesting, answering, greeting, etc.).

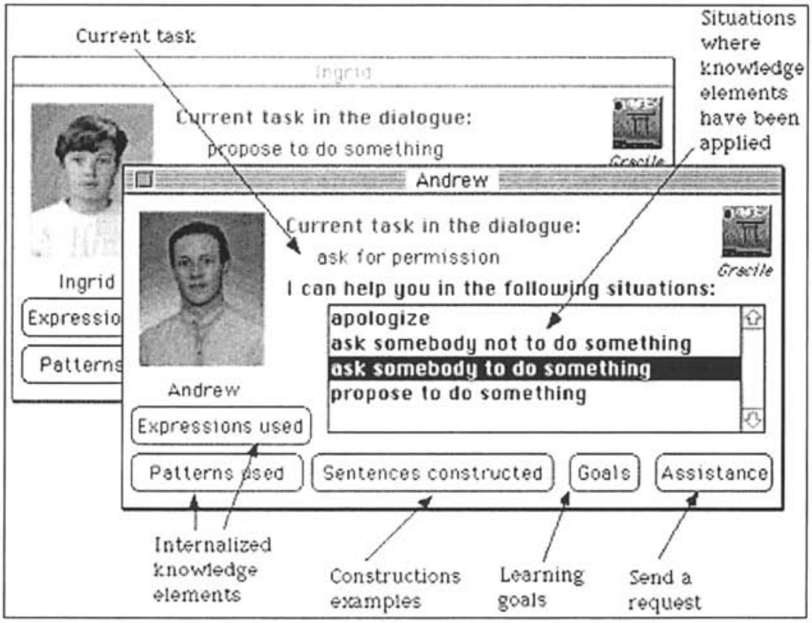


Figure 7-3. Access to open learner models in GRACILE (Ayala & Yano, 1996a)

The dialogue is the problem to be solved by the group, and its structure can be considered as a script that indicates the tasks assigned to each learner. The mediator agent in GRACILE can keep the learners' awareness on the environment and promote the collaboration possibilities of the learners in the group by supporting the communication of the capabilities of the learners, as well as their constructions considered as a correct application of knowledge elements.

For non pedagogically structured domains, as in CASSIEL, an *information agent* assists the learner in the location of relevant resources in a repository and keeping her aware of the changes and annotations by other group members. Also, a *facilitator agent* assists the learner in the organization of her ideas, beliefs and knowledge to be provided to the rest of the community. It supports the phase of knowledge externalization and provides workspace awareness (Ayala, 2003).

3.2 The learners' coordination script

Scripts for these detailed work processes are implemented by communication interfaces that prompt learners to participate in collaborative activities, and do not allow discussions out of context of the joint problem. In the approach for pedagogically structured domains, the environment, not the agent, maintains the control of the participants in the problem solving process. The role of software agents here is concerned with the maintenance of awareness that allows the learners to know their collaboration possibilities in the group.

The mediator agent in GRACILE keeps the learner aware of the environment and promotes the collaboration possibilities of the learners in the group by supporting the communication of:

1. The learning goals of the learners, so they can be aware of the intentions of each other.
2. The learners' commitments, so they know the tasks the learners are going to perform and who is going to assist whom.
3. The learners' capabilities, so they will understand who would be able to assist them in a given situation.

The mediator agent keeps the learners aware of the collaboration possibilities in the group by allowing the access to the information in the open learner models (see Figure 7-3).

In GRACILE, when constructing a sentence for the common dialogue, a learner can make a request for assistance from other learners in the network via her mediator agent. After consulting the capabilities and commitments in the learner model of a given group member, the learner may decide to send her a request of assistance. Also, the learner may ask the mediator agent to send requests of assistance to anybody. In such a case, the mediator agent sends the requests only to the members of the learning group considered able to help. A request of assistance consists of:

1. The situation in which assistance is needed for the application of domain knowledge (i.e., a situation in the dialogue represented by communicative acts like "apologize", "ask somebody not to do something", "make a proposal", etc.).
2. An additional message (text) explaining details of the help needed.

During collaborative writing of a dialogue, the mediator agent allows the learners to discuss the appropriateness of their constructions. When learners disagree about a construction in the dialogue they cooperate constructing a new sentence, discovering the differences between their prior beliefs and alternative applications of domain knowledge in a given situation. In order to support an alternative viewpoint for the common workspace, the learner has to justify it with an example considered valid, previously accepted by the learners. In this way the environment allows the identification and resolution of differences in the application of domain knowledge.

In the case of non pedagogically structured domains, coordination is necessary for the process of knowledge construction by the members of the community. Software agents should be modeled in order to guide learners through the complex work of knowledge construction.

From this perspective of scripting, in CASSIEL the agents assist the learners in structuring the social construction of knowledge, maintaining a model of the learner's interests in order to support diverse types of awareness. The approach adopted for knowledge construction is the one for *constructive web-based learning environments*, based on the development of new documents, annotations and relations between them (Wolf, 1996).

An *information agent* is important in order to keep the learner aware of the location of relevant resources in the shared knowledge repository and of the changes and annotations in the repository by other group members. Keeping workspace awareness requires information about new documents added by the community and their corresponding annotations. Annotations are required in order to preserve quality and social acceptance of the digital documents in the repository constructed by the members of the community. Workspace awareness is necessary during knowledge externalization and knowledge combination.

A *facilitator agent* assists the learner in the organization of her ideas, beliefs and knowledge to be provided to the rest of the community. It supports the phase of knowledge externalization and provides workspace awareness. It also supports the construction of the relations between a given situation and a resource, representing the correct application of knowledge for a given problem. This task awareness is required in order to promote the application of knowledge in the community, in the form of annotations to the resources, mainly as stories of success.

4. CONCLUSIONS

There are two approaches from the computer science perspective of scripting for collaborative learning in agent-based CSCL environments: one as an organizational process, at the macro level, and the other as a detailed work process, at the micro level. Agent modeling and cooperation script design are the issues in the development of processes at the macro level for group configuration and task assignment, and at the micro level, for learners coordination.

The pedagogical organization of the knowledge domain is an important issue in the modeling of cooperation scripts and their support by software agents.

The role of agents in supporting scripting in CSCL environments must be to reduce cognitive load in the learner, following the script, and keep her aware of learning opportunities and available resources. At the macro level, the agent modeling issue for supporting the scripting must be oriented to provide learning opportunities for the users, while at the micro level the key aspects are coordination and awareness.

At the macro level, agents' support for cooperation scripts refers to group configuration and task assignment. For pedagogically structured domains group configuration is based on the learners' capabilities, and task assignment is based on the maintenance of zones of proximal development for the participants. For non pedagogically structured domains, group configuration is based on the interests of the learners, and task assignment on community membership.

At the micro level, cooperation scripts refer to coordination, and awareness. For pedagogically structured domains, agents provide support by presenting open learner models representing the capabilities and constructions of the participants, making people aware of collaboration possibilities during problem solving. For non pedagogically structured domains, the support consists in coordinating the participation of learners in the construction of a digital repository, being aware of the changes in the repository and being assisted while including new documents, annotations and relations between resources.

REFERENCES

Ayala, G., & Yano, Y. (1995). GRACILE: A Framework for Collaborative Intelligent Learning Environments. *Journal of the Japanese Society of Artificial Intelligence, 10*(6), 156-170.

Ayala, G., & Yano, Y. (1996a). Learner Models for Supporting Awareness and Collaboration in a CSCL Environment. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.), *Intelligent Tutoring Systems, Lecture Notes in Computer Science 1086*, (pp. 158-167). Berlin: Springer.

Ayala, G., & Yano, Y. (1996b). Intelligent Agents to Support the Effective Collaboration in a CSCL Environment. In P. Carlson & F. Makedon (Eds.), *Proceedings of the ED-TELECOM 96 World Conference on Educational Communications* (pp. 19-24). Boston, MA: AACE.

Ayala, G., & Yano, Y. (1997). Evaluating the performance of agents that support the effective collaboration of learners in a CSCL environment. *IEICE Transactions on Information Systems, E80-D*(2), 125-134.

Ayala, G. (2002). Intelligent Agents for a Lifelong Learning Environment in Information Technologies. *International Journal of Continuing Engineering Education and Life-Long Learning, 12*(1-4), 189-200.

Ayala, G. (2003). Towards lifelong learning environments: Agents supporting the collaborative construction of knowledge in virtual communities. *Designing for Change in Networked Learning Environments: Proceedings of the International Conference on Computer Support for Collaborative Learning CSCL 2003* (pp. 141-148). Bergen, Norway.

Ayala, G., & Paredes, R. (2003). Learner model servers: Personalization of web based educational applications based on digital collections. *ED-MEDIA 2003, World Conference on*

Educational Multimedia, Hypermedia & Telecommunications [CD-ROM]. Honolulu, USA: AACE.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.) *Three Worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen: Open Universiteit Nederland.

Kollar, I., & Fischer, F. (2004). Internal and external scripts in web-based collaborative inquiry learning. In P. Gerjets, P. A. Kirschner, J. Elen, & R. Joiner (Eds.), Instructional design for effective and enjoyable computer-supported learning. *Proceedings of the first joint meeting of the EARLI SIGs "Instructional Design" and "Learning and Instruction with Computers"* (pp. 37-47) [CD-ROM]. Tübingen: Knowledge Media Research Center.

Morabito, J., Sack, I., & Bhate, A. (1999). *Organization modeling: Innovative architectures for the 21st century*. Upper Sadle River, NJ: Prentice Hall PTR.

Nonaka, I., & Takeuchi, H. (1995). *The Knowledge Creating Company*. Oxford: Oxford University Press.

Norman, D. (1994). How might people interact with agents. *Communications of the ACM, 37*(7), 6-71.

Vygotsky, L. S. (1978). *Mind in society: The Development of higher psychological processes*. London: Harvard University Press.

Wolf, K. D. (1996). Analyzing the process of learning in a web based community of learners. In P. Carlson & F. Makedon (Eds.), *Proceedings of the ED-TELECOM 96 World Conference on Educational Communications* (pp. 337-342). Boston, MA: AACE.

Chapter 8

MODELING CSCL SCRIPTS – A REFLECTION ON LEARNING DESIGN APPROACHES

Yongwu Miao, Andreas Harrer, Kay Hoeksema, and Heinz Ulrich Hoppe
Universität Duisburg-Essen

Abstract: The design of collaboration scripts is a new focus of research within the CSCL research community. In order to support the design, communication, analysis, simulation and also the execution of collaboration scripts, a general specification language to describe collaboration scripts is needed. In this chapter, we analyze the suitability and limitations of IMS LD for modeling collaborative learning processes. Based on the analysis, we propose an approach to designing a CSCL scripting language. This chapter presents the conceptual framework of this modeling language and the solutions to the identified problems of IMS LD for formalizing collaboration scripts. Especially, we compare the two approaches through modeling the same collaboration script by using IMS LD and our own CSCL scripting language.

1. INTRODUCTION

According to O'Donnell & Dansereau (1992) a collaboration script is a set of instructions specifying how the group members should interact and collaborate to solve a problem. The term *script* was initially used in schema theory by Schank and Abelson (1977). According to schema theory, a script is a mental structure representing the people's knowledge about actors, objects, and appropriate actions within specific situations. When members of a learning group interact with each other, a shared script can help them to reduce the uncertainty about coordination efforts (Mäkitalo, Weinberger, Häkkinen, & Fischer, 2004), because they know how to behave and what to expect in particular situations. By providing learners with a collaboration script, it is also possible to support learners in aiming at cognitive objectives like fostering understanding or recall (Rummel & Spada, this volume). Additionally, collaboration scripts might also foster the development of metacognitive, motivational, or emotional competence (Kollar, Fischer, & Hesse, in press). A collaboration script is normally represented in the learners' minds (internal representation) and can be represented somewhere in the learning environment (external representation) with complex interplay between these two levels of representation (Carmien, Kollar, Fischer, & Fischer, this volume). King elaborates on the cognitive perspective of CSCL scripts (King, this volume). Because we focus on using collaboration scripts in computer settings, we are interested in representing collaboration scripts in a formal way so that they can be processed by the computer. Such a computational representation of a collaboration script is called a CSCL script.

The conceptual components of a collaboration script and their relations have been discussed in literature (Dillenbourg, 2002; Kollar et al., in press). However, a general modeling language for formalizing collaboration scripts is still missing and most CSCL scripts are embedded or encoded into the learning support environment. Furthermore, there are only few corresponding authoring tools for CSCL practitioners to create, reuse, integrate, and customize CSCL scripts without substantial prerequisites of technical knowledge; there are some proposals for script modeling based on finite automata (Haake & Pfister, this volume) or statecharts (Harrer & Malzahn, 2006) to represent more complex learning processes than linear ones, yet this representation might still be unfamiliar to the educational practitioner. As a first step in the direction of a general CSCL scripting language we investigate in existing learning process modeling languages. The most important attempt in the current discussion in this direction is IMS Learning Design (IMS LD; see IMS LD Website), a standard published by the IMS consortium based on the earlier Educational Modeling language (EML) developed at the Open University of the Netherlands OUNL (Koper, 2001). It is claimed that IMS LD can formally describe any design of teaching-learning processes for a wide range of pedagogical approaches (Koper, 2001; Koper & Olivier, 2004). This modeling language has strengths in specifying personalized learning and asynchronous cooperative learning. However, IMS LD provides insufficient support to model group-based, synchronous collaborative learning activities. Caeiro, Anido, and Llamas (2003) criticized IMS LD regarding CSCL purposes and suggested a modification and extension of the specification. This modification and extension focuses on the elements role-part and method part. Hernandez, Asensio, and Dimitriadis (2004) suggested adding a special type of service, called "groupservice" to extend the capacity of IMS LD. Such an extension at service level, rather than at activity level, cannot appropriately capture the characteristics of collaborative learning activities, because different services may be able to support the same collaborative activity.

The research work presented in this chapter aims at developing a scripting language for formalizing CSCL scripts and exploring their potential types of usage and system support possibilities. In this chapter, first we explain how a scripting language can help CSCL practitioners (e.g., teachers and students) in the design phase (e.g., editing, communicating, predicting, simulating) and in the execution phase (e.g., configuration, monitoring, scaffolding). Then, we clarify the limits of IMS LD when working on a computational methodology for the scripting of collaborative learning processes . Based on the analysis, we propose an approach to design a CSCL scripting language. Rather than a systematic description of the CSCL scripting language, we present it by focusing on how the identified problems of IMS LD for CSCL scripts are solved. In order to compare these two approaches, we present how to model an example collaboration script with IMS LD and by using our CSCL scripting language.

2. POTENTIAL USES AND SYSTEM SUPPORT OF CSCL SCRIPTS

In the following we divide the potential uses of a CSCL modeling language and the computer support it can enable into *usage types during design time* and *usage types while students are performing the learning activities* defined by a designed model. The first category is mainly oriented towards the support of the designer in creating CSCL scripts, while the latter category targets the amount of help a computer system can provide in implementing effective scripts. Dillenbourg and Jermann provide a more general discussion of the added value of computer support for learning scripts in (Dillenbourg & Jermann, this volume).

2.1 Design time uses

The specification of learning processes using a modeling language may have a broad variety of purposes on the part of the designer. Some educational designers use it as a note taking tool for lesson planning. Created models can be saved and used (complete or partially) as a basis for further development. Models can be used for communication between designers. Even at an early state of development, when the model is far from being operational, it can already express educational ideas. Though, due to the complexity of collaborative learning processes, the models get excessively complex and hard to understand. Therefore either reduction of the complexity (by applying projections of specific elements or filtering techniques) or the separation into different perspectives is a typical way to cope with the complexity. The designer can switch between the different perspectives to keep an overview, always choosing the perspective most suitable for further authoring. For learning processes typically the following aspects are relevant and thus candidates for special perspectives:

Procedural/Temporal Perspective. Naturally the sequencing and timing, that is, the process related aspects of the whole learning process, should be represented explicitly

Artifacts Perspective. Artifacts given as resources, used as temporary results and the final outcome of learning activities constitute an important aspect of learning processes. Especially the change of artifacts over time (version history) is information to consider by all participants of a learning process.

Roles Perspective. For organization of specific tasks in group processes the various roles needed for the tasks are an essential information, not only during design time.

Individual/Group Perspective. To get an impression of the workload of one specific member or one subgroup within a group process a perspective stressing these individual aspects is a valuable information for the designer to keep balance between the participants of the process.

The more details of a collaborative learning process are defined, the more the authoring system can provide help to the designer. For example, dependencies or constraints between elements can be highlighted, such as necessity of sequential phases or synchronizing the flow after a split into cooperative sub processes. If the designer specified temporal constraints (minimum or maximum time) for elements of the process, techniques from operations research, such as optimization in network flows or critical path analysis can be applied. A simulated execution of the specified learning process can give the designer a more profound feedback on "what works and what does not?". Imagine the benefit of doing a *simulation run* with information about sequence, time requirements, and produced artifacts before applying the whole design to a real learning situation. The plausibility of the design can be checked much easier than just based on the static structure of the model. Deadlocks (e.g., when subgroups are waiting for each others' input) in the process specification can be detected before making the bitter experience in practical use.

2.2 Runtime uses

The first, weak approach to operationalizing the learning process for the target user "at run time" is the configuration of the learning environment with available tools, resources, communication structure and so on. If this configuration is done once without dynamic addition and removal of elements we call this static configuration. "Compiling and instantiating" such an environment from the specification should be the minimal functionality of a system meant for "playing" the learning design.

While running a learning process model the system can monitor the activities performed by the students. Monitoring functionality could be used twofold: On the one hand the information can be used internally to adapt the process according to the exact specification, on the other hand the monitored information can be visualized to participants of the learning process and give them information on what they have done and produced. This additional feedback can be used to promote reflection about the process or the participants' own behavior, e.g., to stimulate meta-cognitive activities.

At the "informed end" of the spectrum of computer support we see the potential use of the system for scaffolding the learning process, especially when the "typical path" through the process was left by the participants (Koedinger et al., 2004). An enriched specification can give advise to and offer a scaffold to the learners on "what and when to do, how they can play their assigned role best" and so on. Depending on the strictness of the scaffolding the system's behavior can vary between an unrestraining advisor and an intervening tutor. Ideally a script could contain dynamic aspects for adaptive fading in and fading out of scaffolds for the learners.

3. INVESTIGATING THE CAPACITY OF IMS LD FOR FORMALISING COLLABORATIVE LEARNING SCRIPTS

A collaborative learning experience can be described by a collaboration script. Many collaboration scripts have been designed, tested, and even embedded in CSCL applications (e.g., Hoppe & Ploetzner, 1999; Guzdial & Turns, 2000; Miao, Holst, Haake, & Steinmetz, 2000; Pfister & Mühlpfordt, 2002). When using IMS LD to formalize collaboration scripts, we see several major difficulties and challenges:

Modeling groups: Modeling group work with IMS LD raises the problem how to model multiple groups with the same role and how to model the dynamic changes of groups. IMS LD allows for defining multiple roles. Each role can be played by multiple persons. When investigating, we found that in many cases the notational element of "role" can be used to model groups for CSCL scripts. However, by using IMS LD it is very difficult to specify how a group work pattern is assigned to several groups working in parallel and how sub groups can be defined within these groups. If each group or subgroup is defined as a role, the designer has to define a list of roles representing multiple groups. The problem of this solution is that the number of groups in a run is unpredictable during the modeling phase. If only one role is defined for all members of all subgroups, then the information about groups or subgroups will be missing and the run-time system cannot support inter-/intra-group collaboration appropriately. In addition, in IMS LD roles are assigned to persons before running a unit of learning and these assignments stay unchanged within the life cycle of the run. However, in some situations groups are formed and group members are assigned after the start of the process execution. Therefore, in some situations, the notational element of role cannot meet the requirement to model groups.

Modeling artifacts: A second major difficulty while modeling CSCL scripts with IMS LD is the modeling of artifacts. In learning processes, actors usually generate artifacts such as a vote, an answer, an argument, or a design. In IMS LD, an artifact can be modeled as a property, for example a property of a person or a role, that creates the artifact. This property can be used to maintain information such as the learning outcome of a person or a role and to support personalised learning. In collaborative learning processes, an artifact is usually created and shared by a group of people. It is normally used as an object of mediation to facilitate indirect interaction among group members. It may be created in an activity and used in other activities like in an information flow. In order to support group interaction, an artifact should have attributes such as artifact type, status, created_by, creation_activities, contributors, consume_activities, current_users, and so on. By using IMS LD to model an artifact as a property, one has to model all attributes of the artifact as properties as well. These properties should be defined as a property-group with specific constraints. Such a complex definition cannot be understood intuitively. It will be very difficult to model dynamic features even for technically experienced designers, because the limited data-types of properties and the number of references needed make it very complicated to handle artifacts. In addition, it is difficult to model a collective artifact, because IMS LD does not support array-like data-types for a property.

Modeling dynamic features: A third major difficulty while modeling CSCL scripts with IMS LD occurs when modeling dynamic process aspects. IMS LD provides two categories of operations on process elements: read-access operations ("getters") to get the state of process elements (e.g., users-in-role, datetime-activity-started) and write-access operations to change the state of process elements (e.g., change-property-value, hide/show elements, and send notification) to model dynamic features of learning processes. For modeling collaborative learning processes, more of these read and write operations are needed. At least, process element operations concerning our proposed extensions like group and artifact should be extended. In addition, some destructive or constructive operations (e.g., form a group with only male members) should be added. Furthermore, more complicated operations based on these elementary operations will be performed by run-time systems or by users (e.g., to do the configuration and logistics work such as distributing artifacts within a group). Adding such actions will empower learning designers to model complicated processes without being bothered by the technical complexity.

Modeling complicated control flow: A fourth major problem is how to model complex process structures. IMS LD provides play, act, role-part, and activity-structure to model structural relations at different levels. Primarily learning/teaching processes that are structured in a sequential way with concurrently executable activities can be modeled. However, as Caeiro et al. (2003) pointed out, the linear structure of a play with a series of acts introduces a great rigidity while modeling network structures. Although it is possible to model non-linear structural relations among activities by using conditions and notifications, the specification of a collaborative learning process might be very complicated and confusing.

Modeling various forms of social interaction: The last difficulty we want to stress in this chapter occurs when modeling various forms of social interaction. IMS LD uses a metaphor of a theatrical play to model learning/teaching processes. A play consists of a sequence of acts and within an act there is a set of role-parts. These role-parts can run in parallel. Role-parts enable multiple users, playing the same or different roles, to do the same thing or different things concurrently on the same act. For example, while each student reads the same article, the teacher prepares presentation slides. If a group of people performs a synchronous activity, IMS LD enables them to use a conference service and provides no means at the activity level to support collaboration. In collaborative learning processes, it is quite usual that people with the same or different roles perform a shared activity through direct or indirect interaction. While making the joint effort, people with different roles may have different rights to interact with other roles and the environment. In particular, it can not be clearly modeled by using IMS LD whether and how people collaborate, because people may work in a variety of social forms: Individually, in an informal group, in sub-groups, in a group as a whole, or in a community.

4. AN APPROACH TO REPRESENT CSCL SCRIPTS

In order to enhance effective collaboration designs, we have developed a CSCL scripting language to represent collaboration scripts. Because of the limited space of the chapter, we briefly present the CSCL scripting language by explaining the core concepts and their relations, rather than giving a systematic description. Then we focus on describing how the identified problems of IMS LD for CSCL scripts are solved in our scripting language by introducing the required constructs on the conceptual level. This does not necessarily imply that we want to provide a completely independent approach for formalizing learning processes, including providing our own interpreting machine or engine. At the moment we are still in the process of exploring if the existing standard can be extended according to the identified needs. Another possibility is to consider our approach as a higher-level one closer to the practitioner's and researcher's needs that can be "compiled", that is, semantically mapped to the existing description format. This question in its completeness is unresolved, but we will give some details on the aspects that we consider to be resolved at the moment.

4.1 A conceptual basis for CSCL scripting

In this subsection, we briefly present the core concepts and their relations of the CSCL scripting language.

A CSCL script is a specific learning design which emphasizes collaboration. A CSCL script contains contextual information that applies to other elements within the process. As shown in Figure 8-1, a CSCL script consists of a set of roles, activities, transitions, artifacts, and environments. A CSCL script has attributes such as learning objectives, prerequisites, design rationale, coercion degree, granularity, duration, target audience, learning context, script specific properties, and generic information (e.g., id, name, description, status, creation date, and so on). The attribute *design rationale* enables to express and communicate the design ideas and underlying pedagogic principles. The values of the attribute *coercion degree* represent different degrees of *informedness*. CSCL scripts with different coercion degrees have different usages, which will be discussed later in the chapter. If a CSCL script of fine granularity is embedded in a CSCL script of coarse granularity, the mappings between the roles, properties, and artifacts of two CSCL scripts should be specified. A role is used to distinguish users who have different privileges and obligations in the processes described in the CSCL script. Both persons and groups can take a role. A group can have subgroups and person members. An activity is a definition of one logical unit of a task performed individually or collaboratively. There are three types of activities: atomic activity, compound activity, and route activity. A compound activity is decomposable into a set of networked activities and even other scripts. A transition specifies a relation of temporal dependency between two activities. An artifact may be created and shared in and/or across activities as an intermediate product or a final outcome or both. An environment can contain sub-environments and may contain tools and contents. A tool may use artifacts as input parameters or output parameters or both. A content is a kind of learning object which exists and is accessible. An action is an operation and may be performed by users during an activity or by the system before or after an activity. A property may be atomic or may have internal structure. An expression may use properties and other expressions as operands. Like IMS LD, a condition refers to a condition clause which is defined as an if-then-else rule consisting of a logical expression and actions, transitions, and/or other conditions. Actions, properties, expressions, and conditions have very complicated relations with other process elements (e.g., scripts, roles, activities, artifacts, persons, groups, environments, and so on). For example, an action may use process elements as parameters and change the values of attributes of certain process elements. Such relations are not drawn in this diagram in order to keep the diagram simple and readable.

Using the scripting language to formalise a collaboration script means specifying how persons or groups or both, playing certain roles, work collaboratively towards certain outcomes (which can be artifacts) by performing temporally structured activities within environments, where needed tools and content are available. Actions, properties, expressions, and conditions are useful to model more complicated, dynamic control-flow and information flow in collaborative learning processes.

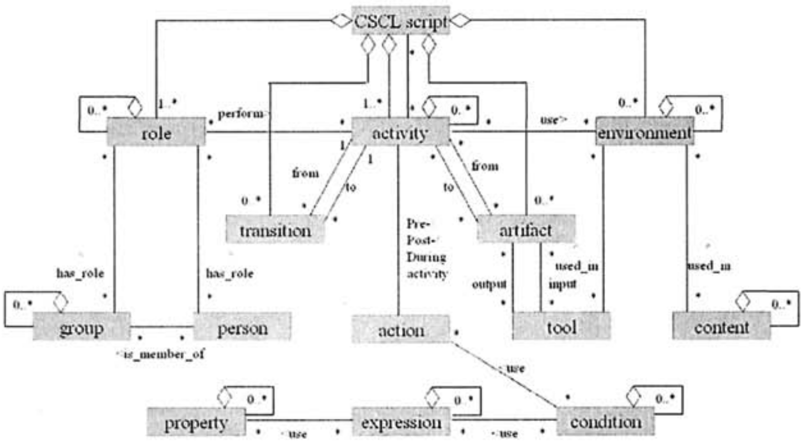


Figure 8-1: Core modeling elements and their interrelation

4.2 Solutions

In this subsection, we focus on presenting our solutions to the identified problems of IMS LD for CSCL scripts.

Explicitly introducing groups. The introduction of a group element enables us to model group based collaboration in a simpler and more intuitive way. In our CSCL scripting language, a group is modeled by using attributes such as name, max-size, min-size, person members, super-groups, subgroups, engaged roles, form-policy, disband-policy, dynamic/static, and runtime information. In addition, local-/global group properties are added for learning designers to define additional attributes of a group. One or more groups can play the same a role. Therefore, when a role is defined and is assigned to carry out an activity, it does not matter how many groups will play this role at runtime. On the one hand, a group can have subgroups and form a hierarchically structured organization (a directed-acyclic-graph). Any change in the organization has no effect on the definition of the role in scripts. On the other hand, re-definition of roles in scripts does not effect organization. This proposal raises the question when to model a group and when to use a role for a group. From our perspective, some roles are organization oriented definitions like students and staff. Others are behavior-oriented roles such as meeting chairman and tutor. It would be better to model an organization-oriented role as a group role and to model a behavior-oriented role as a role for assigning tasks.

Explicitly introducing artifacts. The artifact element does not exist in the IMS LD specification. As we explained already, the usage of artifact elements can enable to model CSCL contexts much more intuitive and easier than to model the same process within IMS LD, because some burdens on the designers to handle technical tasks are avoided by providing built-in mechanisms. In our language, an artifact is treated as a file which can be a MIME-type or user-defined type. The attributes of an artifact contain generic information (e.g., title, description, type, status, URL, sharable, and aggregated), association information (e.g., creation_activities, consume_activities, and default_tool), and run-time information (e.g., created_by, creation_time, contributors, last_modification_time, current_users, locked_status, and so on). An artifact and its status will be accessible in the environment of the creation-/consume- activities at run-time. The specification of the relations between artifacts and tools will help the run-time system to pass artifacts as input/output parameters to and from tools automatically at runtime. Some expressions and actions related to artifacts should be added for mediating group work such as get-current-users-of-artifact and change-artifact-status. The artifact-specific properties may be useful to model a specific feature of an artifact. As an aggregated artifact, it is possible to append collective information to the same file.

Extending actions and expressions. An action is a generic and powerful mechanism to model dynamic features of a collaborative learning process. We add some actions as components of the CSCL scripting language that can be executed directly by the runtime system. In addition, we add an action declaration mechanism for experts to define a procedure by using the CSCL scripting language. In order to support the definition of complicated procedures, we add a "collection" data type and a loop control structure. The defined procedure can be interpreted by the run-time system as process element operations, and in turn, as executable code. Therefore, complicated actions can be defined by using an action declaration and assigning the parameters needed. IMS LD provides a limited set of actions such as property operations, showing/hiding entity, and notification. The action notation we introduced provides a unified form of operations including not only actions defined in IMS LD but also commonly used operations concerning script, activity, artifact, role, group, person, transition, environment, and their relations. An expression is defined as it is in IMS LD: some read operations can be used as operands in expressions like "is-member-of-role", "datetime-activity-started", and "complete". However, it is necessary to add read operations to support collaboration such as "are-all-role-members-online" and "artifact-contributors". Furthermore, corresponding to the action declaration, we add an expression declaration mechanism for experts to define complicated expressions which could be used by normal teachers and students.

Introducing transitions and routing activities. We partially accept the suggestion of Caeiro et al. (2003) to introduce transitions and routing constructs recommended by the Workflow Management Coalition (WfMC Website). Because interactions of person-to-person, group-to-group, and role-to-role and splitting and synchronization of process threads are never restricted at higher levels, we have to use such a mechanism not only at play level but at all possible levels in order to model the arbitrarily complicated structural relations among activities.

Using activity-centered methods to assign roles. We give up the metaphor of a theatrical play and the role-part method. Instead, we use an activity centered role assignment method. In the CSCL scripting language, for modeling an activity, the attributes are defined to specify engaged roles, used environments, input/output artifacts, transitions and restrictions, pre-/post-/during activity actions, user-defined activity-specific properties, completion-mode, execution-time, completion-condition, mode of interaction, social plane, interaction rules, generic information, and simulation information. Some attributes are important for designers to model collaborative processes and some for the run-time system to configure collaborative learning environments appropriately for users. For example, the possible values of social planes are: separately with a certain role, individually with a certain role, collaboratively with one or multiple roles or both, collaboratively in subgroups with a certain role, and so on. If the choice is "separately", the run-time system will create an activity instance for each user starting the activity. If anyone completes his activity, all activity instances terminate. "Individually" means that the run-time system will create an activity instance for each user. The run-time system synchronizes access to the following activity by continuously checking whether all users have already completed the current activity. In comparison, the run-time system based on IMS LD typically handles this situation defined by using the role-part method. The choice of "collaboratively with one and/or multiple roles" makes the run-time system create only one activity instance and a session facilitating collaboration. The semantics of the value "collaboratively in subgroups with a certain role" is that the run-time system creates an activity instance and a session for each sub-group and the members of each sub-group can have a shared activity workspace. The run-time system synchronizes access to the next activity when all subgroups finish their work. Another example is the attribute *interaction rules*. An interaction rule specifies under which condition which role can (not) perform which actions. For example, the tutor can perform the actions to create (sub)groups and assign group members. Such information can be used by the run-time system to automatically provide corresponding awareness information in the user interface to help users to perform specified actions. In short, interaction rules explicitly specify different responsibilities of different roles in a collaborative learning activity.

5. MODELING A COLLABORATION SCRIPT WITH IMS LD AND THE CSCL SCRIPTING LANGUAGE

In this section, a collaboration script is used as an example. We discuss how this collaboration script can be modeled by using IMS LD and by using our CSCL scripting language. Our example will be the "Knowledge Convergence Script" (Weinberger, Fischer, & Mandl, 2004, and Weinberger, Stegmann, Fischer, & Mandl, this volume), that has been shown to be effective in improving the learners' convergence either on epistemic or on the social level.

In short this script consists of the following phases and interactions between the members of groups of three students:

- Phase 1 - case reporting: Each student gets information about a (educational) case and is writing a report about the case.

- Phase 2 - criticizing 1: Each student gets the case and the report of the student to his left and writes a comment about the report.
- Phase 3 - criticizing 2: Each student gets a case, the report and the comment the student to his left produced in phase 2 and writes a second comment about the report.
- Phase 4 - Finalizing the report: Each student gets back his own report together with the comments of the two other students and rewrites it taking the comments into account.

The flow of the artifacts produced by the students, specifically the artifacts in relation to Case 1, can be seen in the graphical schema in Figure 8-2.

5.1 How to model the script by using IMS LD

IMS LD is designed mainly for supporting web-based learning environments and the run-time environment will render the web pages for users according to the definition of the unit of learning. To give an impression of the design work we will abstract from generation of HTML and XML content pages, but focus on the major steps in the design process for the Knowledge Convergence Script:

1. Define three roles for the three group members, since IMS LD does not explicitly represent groups. Each role will be constrained to have at most 1 person playing the role.
2. Define 12 properties for the reports and comments produced by the students, because each student writes a report, two comments on the others' reports, and a final version of the report. Properties are the means of choice in IMS LD, because they can be flexibly used for person- or role-related aspects, thus also as a substitute for a missing "document/artifact" construct. For a better structuring it is advisable to compose sets of properties, such as all documents related to Case 1, in so called property-groups, that contain references to their constituents.
3. Define the 12 activities that the learners should perform in this script and their effects on the properties representing the documents (i.e., the products of student writing). These properties have to be set explicitly from the outside, that is, from an external service or from a learning object document.
4. Predefine the document flow (represented in the properties) for each step of the script explicitly, such as "Student 1 has to get report 3 from student 3, Student 2...". This is statically defined for a fixed number of documents and learners.

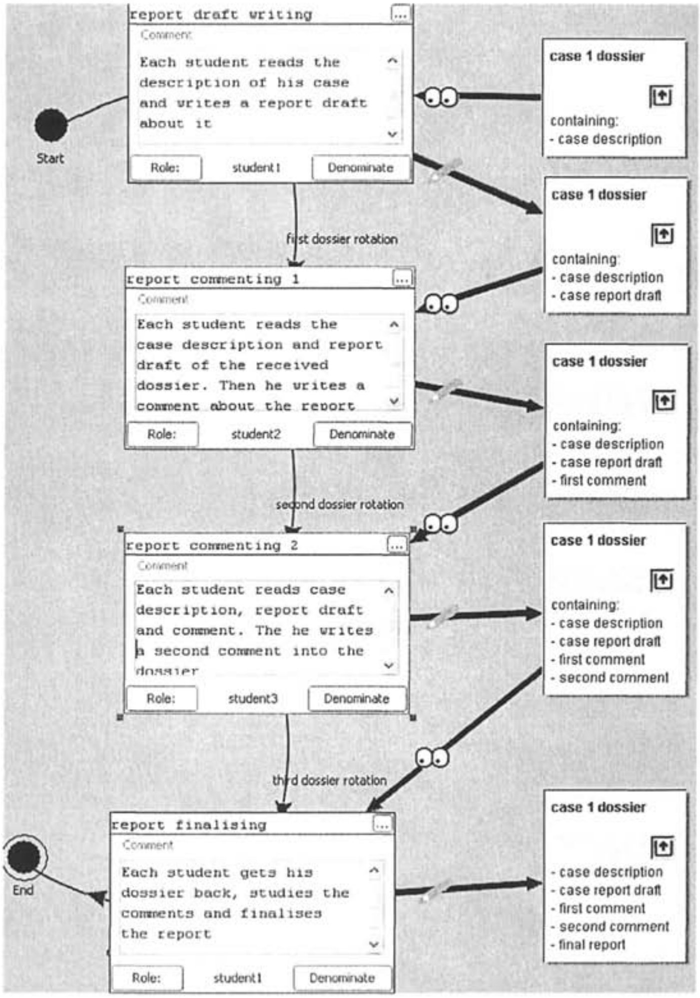


Figure 8-2: Diagram showing the flow of the dossier of Case 1 through activities

5.2 How to model the script by using the scripting language

The same process will now be sketched for the CSCL script representation presented in the previous sections. Our main focus is also on the general overview with some details about practical and technical issues of applying and implementing this notation:

1. Define a group of three members explicitly; there is still also the option of defining groups by roles, but the notation also offers a dedicated "group" construct to the designers.
2. Explicitly define three artifacts that represent the documents produced by the students. For better structuring these artifacts can be aggregated to "composite artifacts" (e.g., one dossier for all documents related to one case and even a collection of all dossiers) of complex structure.
3. Define actions for the initial distribution and the re-assignment/rotation of artifacts to the group members. The independence of concrete numbers for documents and persons is highly desirable, so that the action can be re-used in different situations or stages of the learning process. These actions can be freely defined by a learning designer, if he has some understanding of specifying procedures on an abstract level. In our example the two actions "DistributeArtifactCollection" and "RotateArtifactCyclic" would be very useful, especially the latter, because it is performed after every writing phase of the students, but with different actors getting the dossiers. To give an impression of the specification level of such a generic action we give some pseudo-code representation for "RotateArtifactCyclic" and a graphical schema for this procedure (see Figure 8-3), that gives the dossier to the next group member in sequence.

```
rotateArtifactCyclic(ArtifactCollection art, Group learners){
  while (art.hasMoreElements()){
    assign(art.currentElement(), group.nextMember());
  }
  assign(art.lastElement(), group.firstMember());
}
```

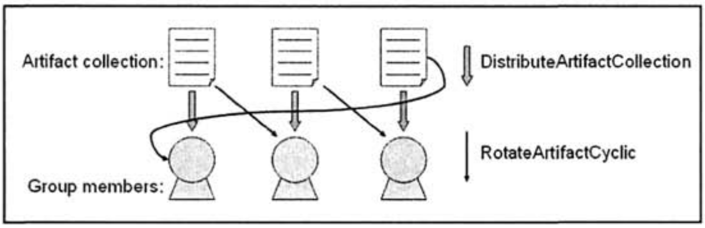


Figure 8-3: The graphical schema for explaining two actions: "DistributeArtifactCollection" and "RotateArtifactCyclic"

4. Use the action to rotate artifacts in the learning design for each step of the script; the advantage of having a generic action definition is now that the action can be re-used now by calling this action with different parameters (the current states of the dossiers and the group) without any manual assignment of the respective documents. This re-use can be done for self-defined actions and also for any library of pre-defined actions that other learning designers created. Thus, in case that a suitable pre-defined action is already available (such as the mentioned "RotateArtifactCyclic" that we defined for our own purposes), the Step 3 can be skipped, which is especially desirable for practitioners without programming skills. Pre-defined actions can be used conveniently in our tree-based editor tool, by choosing and parameterizing the appropriate actions from a list (see Figure 8-4).

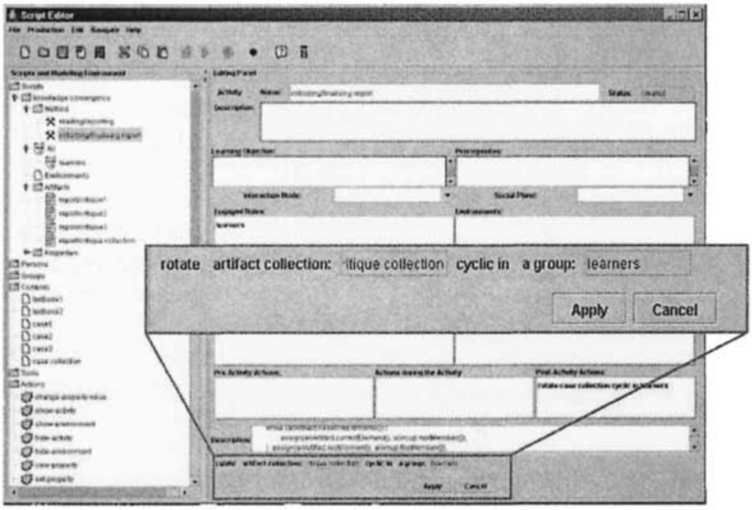


Figure 8-4: Define a post-activity action by assigning parameters

Figure 8-4 shows a screenshot of our tree-based authoring tool when defining the script. The left panel is used to define the script elements (two activities, a group, three artifacts and an artifact collection) and their structural relations. The right panel is used to create a detailed design for each process element, currently for the "criticizing/finalizing-report" activity. The enlarged part illustrates how a post-activity action can be defined in a user-friendly manner. Users can assign the parameters of the action by dragging an element node from the structural tree in the left panel and dropping into the parameter boxes of the action representation.

5.3 Comparison of the two approaches

Although the example script does not cover all features that we discussed in the chapter, we can see the differences when modeling the script with IMS LD and with the CSCL scripting language. We hope that we can provide added value in different respect:

First, the use of a conceptual level, that is closer to the concepts practitioners use, such as the availability of explicit group definitions and artifacts produced by the participants, enables a better understanding for the designer and also in the discussion between practitioners than the IMS LD constructs, such as properties and roles as substitute for groups, offer.

Second, the presented approach of defining own actions in a potentially very generalizable way (using parameters), makes these actions much more re-usable than the IMS LD solution where the definition has to be predefined in a static way; the activity "rotateArtifactsCyclic" in our example can be re-used flexibly within the same script or in a completely different one just by using different parameters for both artifact collection and group, while in the LD solution each step has to be edited again; this is especially useful with different numbers of artifacts to distribute to an arbitrary group (which would not be a problem for our generic activity definition).

Our approach has been prototypically implemented in different tools for editing of CSCL scripts. These tools have been presented in more detail in Miao, Hoeksema, Hoppe, and Harrer (2005).

6. CONCLUSIONS

In this chapter we have identified five major limitations of IMS LD when formalizing CSCL scripts. Based on this, we have suggested a scripting language for CSCL. The identified problems of IMS LD are solved in the language respectively by 1) explicitly introducing the group entity to facilitate modeling organizational role and behavior role; 2) explicitly introducing the artifact entity to enable designers to model artifact and information flow easily and intuitively; 3) extending process element operations and providing declaration mechanisms to capture dynamic features of collaborative learning processes; 4) exploiting WfMS routing technologies to enable the specification of complicated control flow; and 5) giving up the metaphor of theatrical play and the role-part and using an activity-centered definition method to model various forms of social interaction. In addition, we briefly discussed the potential usages of CSCL scripts and possibilities of system support.

Through comparing the two approaches of modeling the same collaboration script with IMS LD and with the CSCL scripting language, we see, at minimum, two advantages of our approach: First, at conceptual level, practitioners can use terms that are closer to the concepts they use in practice. It will be helpful for them to understand and design teaching/learning process models. Second, using actions in our approach makes it possible for practitioners to model complicated processes, because the burden of practitioners to handle technical complexities is reduced.

REFERENCES

Caeiro, M., Anido, L., & Llamas, M. (2003). A critical analysis of IMS learning design. In B. Wasson, R. Baggetun, U. Hoppe, & S. Ludvigsen (Eds.), *Proceedings of the International Conference on Computer Support for Collaborative Learning - CSCL 2003, COMMUNITY EVENTS - Communication and Interaction* (pp. 363-367). Bergen, NO: InterMedia.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen: Open Universiteit Nederland.

Guzdial, M., & Turns, J. (2000). Effective discussion through a computer-mediated anchored forum. *Journal of the Learning Sciences, 9*(4), 437-469.

Harrer, A., & Malzahn, N. (2006). Bridging the Gap - Towards a Graphical Modeling Language for Learning Designs and Collaboration Scripts of Various Granularities. *Proceedings of International Conference on Advanced Learning Technologies (ICALT 2006)*. Los Alamitos, CA., IEEE Press.

Hernandez, D., Asensio, J.I., & Dimitriadis, Y. (2004). IMS Learning Design Support for the Formalisation of Collaborative Learning Flow Patterns. *Proceedings of the 4th International Conference on Advanced Learning Technologies (Aug.30 - Sep. 1, 2004)*, (pp.350-354). Joensuu, Finland: IEEE Press.

Hoppe, U.H., & Ploetzner, R. (1999). Can analytic models support learning in groups. In P. Dillenbourg (Ed.), *Collaborative-learning: Cognitive and Computational Approaches* (pp.147-168). Oxford: Elsevier.

IMS LD Website. Retrieved August 05, 2005, from http://www.imsglobal.org/learningdesign/index.cfm

Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B. M., & Hockenberry, M. (2004) Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In the *Proceedings of the 7th International Conference on Intelligent Tutoring Systems (ITS-2004)*, Maceio, Brazil.

Kollar, I., Fischer, F., & Hesse, F. W. (in press). Collaboration scripts – a conceptual analysis. *Educational Psychology Review*.

Koper, E. J. R. (2001). *Modeling units of study from a pedagogical perspective: The pedagogical meta-model behind EML*. Document prepared for the IMS Learning Design Working Group. Heerlen: Open Universiteit Nederland.

Koper, E. J. R., & Olivier. B. (2004). Representing the learning design of units of learning. *Educational Technology & Society, 7*(3), 97-111.

Mäkitalo, K., Weinberger, A., Häkkinen, P., & Fischer, F. (2004). Uncertainty-reducing cooperation scripts in online learning environments. In P. Gerjets, P. A. Kirschner, J. Elen, & R. Joiner (Eds.), *Proceedings of first joint meeting of the EARLI SIGs "Instructional Design" and "Learning and Instruction with Computers"*. Tübingen: Knowledge Media Research Center.

Miao, Y., Hoeksema, K., Hoppe, U., & Harrer, A. (2005). CSCL Scripts: Modeling features and potential use. In T. Koschmann, D. Suthers, & T. -W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years* (pp. 423-432). Mahwah, NJ: Lawrence Erlbaum Associates.

Miao, Y., Holst, S.L., Haake, J.M., & Steinmetz, R. (2000). PBL-Protocols: Guiding and controlling problem based learning processes in virtual learning environments. In B. Fishman & S. O'Connor-Divelbiss (Eds.), *Fourth International Conference of the Learning Sciences* (pp. 232-237). Mahwah, NJ: Lawrence Erlbaum Associates.

O'Donnell, A. M., & Dansereau, D. F. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interaction in Cooperative Groups: The theoretical Anatomy of Group Learning* (pp. 120-141). London: Cambridge University Press.

Pfister, H.-R., & Mühlpfordt, M. (2002). Supporting discourse in a synchronous learning environment: The learning protocol approach. In G. Stahl (Ed.), *Proceedings of the CSCL2002 Conference on Computer Supported Collaborative Learning, Boulder, USA* (pp. 581-589). Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

WfMC Website. Retrieved August 05, 2005, from http://www.wfmc.org/index.html

Weinberger, A., Fischer, F., & Mandl, H. (2004, April). Knowledge convergence in computer-mediated learning environments: Effects of collaboration scripts. *85th Annual Meeting of the American Educational Research Association (AERA)*. San Diego, CA, USA.

Chapter 9

SCRIPTED ANCHORED DISCUSSION OF MULTIMEDIA LECTURE RECORDINGS

Tobias Lauer and Stephan Trahasch
Albert-Ludwigs-Universität, Freiburg

Abstract: Lecture recording has become a widespread way of producing e-learning contents. The resulting documents, however, are usually limited with respect to collaborative learning. This chapter introduces the concept of *scripted anchored discussion* as a means to facilitate net-based group collaboration around multimedia lectures. Discussion contributions are anchored at specific spatial and temporal positions within the document, allowing both document-centred and discourse-centred views of the discussion. Cooperation scripts are used to structure and sequence the discussion process. A formal model is proposed to represent discussion scripts. Furthermore, we explore ways to fade out scripting instructions in order to adapt to users' increasing internalization of the scripts.

1. INTRODUCTION

A substantive portion of the digital contents for computer-supported learning available today are multimedia recordings of live lectures and talks. An ever-growing number of institutions – both universities and corporations – have adopted *presentation recording* as a cost-effective way of producing contents out of those resources that have already been there: experts in a discipline and their explanations of a subject to an audience. As most presentations today make heavy use of computer technology anyway, it is a natural approach to also use the computer to preserve the experience of the talk for other audiences. Since the advent of multimedia PCs for the masses in the mid 1990s, a great deal of research and development has been put in methods to automatically record computer-based live presentations in order to produce multimedia learning contents, and today a wealth of different systems and approaches is available (for an overview, see Lauer & Ottmann, 2002; Müller & Ottmann, 2003).

Among the factors making this approach attractive as a content-production method are its cost-efficiency and the very quick availability of the contents. Instructors simply prepare the materials (such as slides, animations etc.) as usual and teach their regular face-to-face lectures, which are recorded automatically by the hardware and software in the lecture hall and turned into a multimedia document (see Figure 9-1). In the ideal case, no additional personnel are required and the instructors do not even notice that they are recorded. Depending on the time required for post-processing and format conversion, the resulting multimedia documents can be ready within minutes after the live class has ended. This is in stark contrast to other production methods for e-learning contents, which often involve a whole team consisting of content experts, pedagogues, media designers and others, resulting in a high cost and long time of production, which has made content creation the "bottleneck" of most e-learning applications. Hence, there are good reasons for using presentation recording, in spite of the shortcomings discussed below.

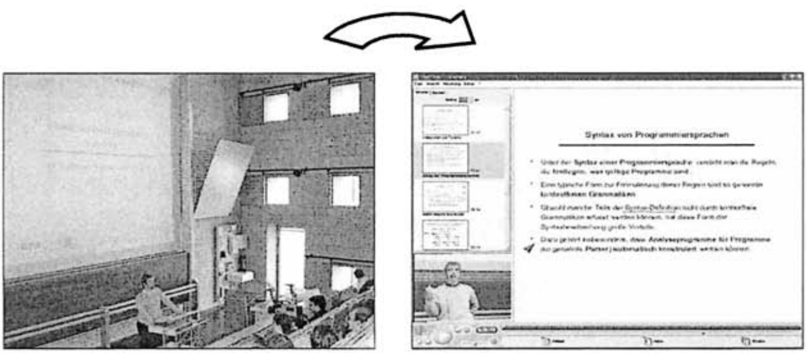


Figure 9-1: Presentation recording turns a live lecture (left) into an "e-lecture."

The resulting contents are far more than simple video recordings of lectures; modern systems produce integrated multimedia documents containing presentation slides and other materials shown, synchronized with the presenter's audio narration, video picture and their annotations, and augmented with rich structural information and navigational support for viewing the contents other than in a purely linear fashion. The tool that this work builds on is the "Authoring on the Fly" (AOF) system developed at the University of Freiburg (see Müller & Ottmann, 2000).

Learners access the resulting *e-lectures* via download or streaming over networks or get them on mass storage media like CD-ROM/DVD, as described by Lauer, Müller, and Trahasch (2004). Comfortable ways of navigation can be provided, such as a table of contents linked to each slide or chapter of the lecture. Real-time random access allows for visible scrolling, a feature enabling users to skim through a recording like flipping pages in a book, in order to locate relevant topics or skip parts of minor interest. Replay at variable speed is possible, which can save time. Full-text search on the slide text allows learners to find certain topics more easily.

Lecture recordings can be used as content modules for distance learning courses or as a supplement for on-campus lectures (blended learning). During the early 2000s, a lot of experience with offering distance and blended courses over the Internet based on lecture recordings was gained in several large-scale projects like VIROR (2006) and ULI (2006). Even students who attend the "real" live lecture make extensive use of lecture recordings when reviewing the content and preparing for examinations, as has been studied by Zupancic and Horz (2002).

A problem found in this context is that learner interaction is quite limited. First, interacting with the contents mainly consists of different forms of search and navigation. Second, user interaction is completely optional, as opposed to many CBTs and WBTs, where quizzes etc. force learners to interact frequently. And third, users can only interact with the computer, not with other co-learners. The reasons can partly be found in the nature of the documents themselves, which – by way of their production method – are rather linear and expository. Even though they do not have to be watched in a linear fashion, there is only limited room for active exploration. Other reasons are not restricted to lecture recordings but also apply to CBT and WBT. For example, it is usually not possible to annotate dynamic, time-dependent multimedia presentations similar to the way printed text and graphics can be annotated. In addition, opportunities for collaborative learning with such documents in real-life scenarios are rather sparse. If collaborative features are provided, they are usually separated from the contents. For example, most learning management systems (LMS) offer chat tools or discussion forums for user interaction and collaboration. However, they are not directly accessible from or linked to the learning contents such as WBTs, and learners have to switch back and forth between the two. Often, communication tools are provided without any instructions or guidance, and learners do not know what they are expected to do with them.

In this chapter, we propose the concept of *scripted anchored discussion* (cf. Trahasch & Lauer, 2005) as a means for structured collaborative learning with lecture recordings. Section 2 describes previous work on online discussion in collaborative learning scenarios and relates artifact-centered discussion to anchored annotation of documents. In section 3, we outline how the concept of cooperation scripts can be used to structure and sequence anchored discussion of time-dependent artifacts. We also investigate methods to fade out parts of a script in order to enhance the internalization by the learners. Section 4 describes a theoretical model, finite state machines, to describe scripted anchored discussions.

2. STRUCTURING AND SEQUENCING OF ONLINE DISCUSSION

Online discussion forums are a widespread means to integrate collaboration into computer-supported instruction. Usually, learning management systems provide the option to include such forums in courses. In their easiest and most common form, discussions are visualized as *threads*. Simply providing a space for discussion, however, does not necessarily lead to its use by students, since every representation system has its own strengths and weaknesses. Thus, it is necessary to consider the characteristics of the communication and collaboration tool and the representational system when combining them (for example, using lecture recordings together with discussion forums). In the following paragraphs we categorize discussion notes according to their relationship to the artifact under discussion and give an overview of visualization of notes and documents. Thereafter we describe some lecture recording systems with annotation features.

2.1 Artifact-centered discussion as exchange of digital annotations

Net-based discussion can be seen as sharing notes (or, annotations) which usually refer to a previous note or to some part of the object of discussion. While learning with traditional, analog media such as books or written notes, learners usually add personal notes or highlight certain parts of a resource or document. Such annotations augment the original document with additional information about structure, relevance, references etc. In recent times, annotation of digital documents has also become a major research area, and systems exist for annotating web pages or PDF documents.

Annotations can be characterized with respect to document type, anchor type and representation of the annotations. In general, one can distinguish between static – for example, text – documents and time-dependent artifacts such as audio, video, or integrated multimedia documents, for example, lecture recordings. Depending on the document type, an annotation can refer to the document in several ways. For example, it could be attached to the whole document (*holistic annotation*). However, in artifact-centered communication there is a strong focus on the document that is discussed. Understanding and following the discussion process can be facilitated by knowing which specific part of the artifact is referred to in a contribution. Therefore, annotations can also be anchored at a specific location (*anchored annotation*) within the artifact. Annotations can be anchored by spatial coordinates relative to the document. This mechanism is problematic for documents with no fixed layout, for example, HTML. In such cases, annotations can be anchored in specific content objects like keywords or phrases. The KOLUMBUS system described by Herrmann and Kienle (2003) allows the annotation of small units of a document, such as paragraphs. However, the document has to be split into meaningful units before. This approach can be seen as an annotation method somewhere in between holistic and anchored annotation. D3E is another example of a Web-based annotation system originally introduced for scholarly discussion (cf. Sumner & Shum, 1998). It allows anchored annotation of Web documents. However, non-textual and dynamic elements such as video clips embedded in HTML pages can only be annotated as a whole. The system is thus less suitable for documents consisting almost exclusively of dynamic media, such as lecture recordings.

Information in dynamic media is distributed not only over space but also over time (and over different perceptual channels). Therefore, annotations must be anchored temporally, since a pure spatial reference, as in static data, is not sufficient. There are different options for anchoring annotations in such documents. A pure temporal anchor at a point of time may be sufficient for annotation of video, if the notes do not refer to particular parts of the screen and are displayed in an extra frame. However, in learning scenarios it is often important to refer to a particular object that is only part of the currently displayed contents. For such cases, it is more useful to provide both temporal and spatial anchors. For notes that are displayed directly on the contents, the duration of visibility is also important. Thus, an interval of time rather than a single point in time is required as a temporal anchor. If a true object-based representation of the contents is available, it may be possible to anchor notes directly at content objects (which implicitly yields spatial and temporal coordinates). In the latter case, a note would be able to move along with the associated object (e.g., in an animation).

Annotations can be stored with the document itself, as is done, for example, in Adobe PDF, or separately from the artifact, for instance in a database. In the latter case it is necessary to save version information of the document with the annotations, because when the document is changed the annotations may not fit to the content anymore. A typical example would be an updated web page.

2.2 Representation of discussion notes

Suthers (2001) distinguishes between different implementations of arranging the discourse and the artifact serving as an anchor. In the *parallel* arrangement, the artifact and the discussion forum are physically separated in two different applications or windows. Typically this layout is useful when holistic annotations are given to static as well as continuous documents. However, the users have to switch back and forth between artifact and communication tool. An example would be a lecture recording watched in a special player and a web forum for discussion displayed in a web browser, where users have to constantly move from one medium to the other. Also, users have to make references to certain parts of the artifact explicit with linguistic means, unlike face-to-face scenarios, where deixis (pointing to something, using "this", "here", etc.) is frequently used. Similarly, other participants have to find the anchors before understanding the discussion note.

If it is intended to give more specific feedback by annotating specific positions in the artifact, this kind of layout is insufficient because references have to be described explicitly by the users, which can lead to misunderstandings. These disadvantages can be avoided by *embedding* the discussion in the artifact. Here, discussion items are directly inserted in the artifact at the appropriate places. Examples are Wiki web pages that can be edited by users. The main drawback of this representation style is that the discussion contributions are scattered over the document. For the users the coherence of the notes is difficult to comprehend and following the whole discussion is more difficult.

A third approach, which combines the advantages of parallel and embedded arrangement, leaves the discussion separate from the artifact but provides links to the respective reference points. Each note is linked with a specific position in the document. Viewing a note scrolls the document towards the referring location. In some versions of this arrangement, the notes can be shown or hidden directly in the document. That way, it is possible to follow the discussion thread in a separate window note by note, or to follow the discussion from the perspective of the document and view the notes anchored at specific regions of the artifact.

2.3 Systems for anchored discussion of lecture recordings

Anchored discussion of lecture recordings has been implemented in several systems. In the approach by Bargeron et al. (2002), the discussion notes are anchored at a time-stamp of the video and can only be displayed in separate frames next to the contents. Another shortcoming is a frequently occurring time delay of about 10 to 15 seconds between the relevant content and the notes, resulting from the fact that learners cannot comfortably navigate back to the exact time in the streamed video after they discover a relevant topic to contribute, and thus insert their notes at a later position in the document than the actual anchor. The eClass project extended their lecture recordings with CoWeb (Pimentel, Ishiguro, Kerimbaev, Abowd, & Guzdial, 2001), which is based on the Wiki principle. At the end of each slide of a recording, learners can add notes, which can be edited by every user. A slide as an anchor for the annotations is very imprecise and of rough granularity. With the system described by Chong and Sosakul (2003) it is possible to annotate video streams, but it is restricted to purely temporal anchors as well. The video sharing and annotation system VSA (Emond, Brooks, & Smith, 2001) allows users to attach notes to streaming video. They are anchored only with an inexact parameter of the timeline. Among other things, teachers using VSA demanded more precision for the point of attachment between video and annotation, context dependency of annotations on the active video segment and video skimming capabilities for previewing video. WebConstellations (Goldman-Segall & Rao, 1998) is a web-based tool for collaborative analysis and organization of media data on the WWW. It requires, however, that the media be in a format that is playable in a browser plugin. The system can thus not handle hybrid documents such as state-of the-art lecture recordings which integrate many different media types. None of the above systems supports spatial references, for example, to a certain part of a slide in a recording.

These disadvantages have been resolved in the Annotation Web Service, which was implemented as part of the Authoring on the Fly (AOF) system. While viewing a lecture document, learners can use a slider to visibly scroll to the exact point of the lecture where they want to add a note. By clicking directly on the associated contents, a new annotation can be created which is anchored both temporally and spatially in the document, as outlined by Fiehn et al. (2003).

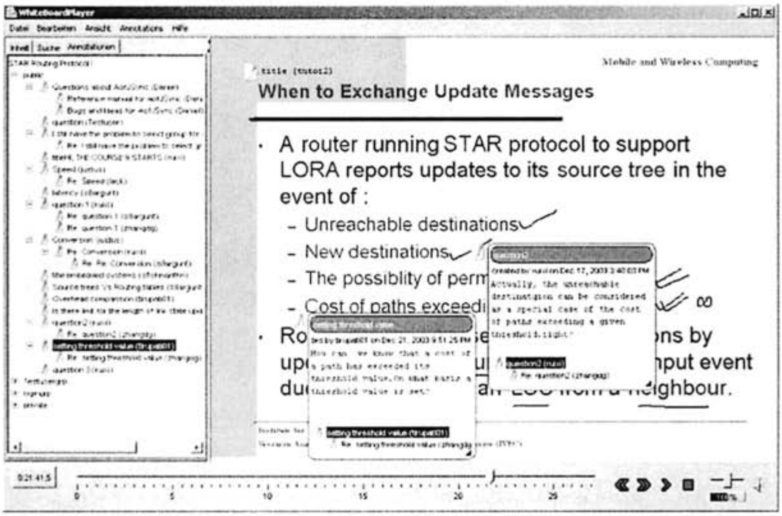


Figure 9-2: Annotation of a lecture recording. The left side contains a view of the notes as a threaded discussion; the right side displays the artifact under discussion with embedded annotations.

The duration of visibility for each note can be set by the user who created the note. By default, a note will disappear as soon as the slide on which it was created is no longer visible. Each note can be minimized to an icon and restored to its original size. Also, they can be moved via drag and drop if underlying parts of the documents are hidden. In addition, all notes can be shown or hidden upon request.

The scope of a note can be declared *public, private,* or *group*. While private notes are only visible to the person who authored them, public and group notes are shared with other learners, who see the notes both in a threaded overview and as virtual "sticky notes" attached directly to the relevant content (see Figure 9-2). These two views are linked such that clicking on a note in the overview will immediately navigate the multimedia document to the referenced position.

Users may reply to any note they see, very similar to standard discussion forums. It is possible to anchor a reply at a position different from that of the original note. This is useful if, for example, a question was anchored at a certain point and the reply refers to a different part of the lecture recording that explains the answer to this question.

3. SCRIPTED ANCHORED DISCUSSION OF LECTURE RECORDINGS

Anchoring discussions in content materials, as illustrated above, avoids the problem of switching between artifact and discussion forum and can thus overcome one obstacle that keeps learners from using such an option. However, it does not mean that learners will therefore engage in a meaningful, structured discussion. Plötzner, Philipp, and Oestermeier (2003) have claimed that especially in e-learning scenarios, where an instructor cannot intervene spontaneously in order to lead the learners back on the right track, additional and more careful pre-structuring of activities is required. Such pre-structuring can be realized by a script of an activity, or a cooperation script, if several learners are meant to collaborate. Using maps as a more open, self-regulated way to guide learners through the learning process is an alternative way to support learners, as has been suggested by Schmidt and Bannon (1992) as well as Herrmann and Kienle (2003). In the context of discussions around lecture recordings, a method using scripts seems to be an appropriate approach.

For online collaborative learning scenarios, we define *scripted anchored discussion* as an activity in which several learners – synchronously or asynchronously – exchange structured comments or notes that are anchored in digital documents. The activity is sequenced in the sense that a script defines the set of actions allowed for any user (or, more precisely, for any role taken by a user) at a given time. Scripted anchored discussion is therefore a subclass or special type of collaboration scripts.

Net-based discussion can be structured by categorizing notes according to their type of contribution (note typing) or by offering sentence openers (prompts), as described by Baker and Lund (1997). When making a contribution, users can choose whether their contribution is, for example, a question, comment, new topic, and so forth. Sentence openers go one step further and provide users with a typical beginning of an appropriate sentence. In this case, users get a list from which they can choose the sentence opener that fits their intended discussion note. These openers implicitly represent the type of contribution. Both facilitation features can also be mixed in a composed element consisting of a note typing and an according prompt.

All these kinds of structuring elements can be categorized according to the supported dimension of the collaborative learning process, like communication or coordination. Furthermore, instructions on how to use a specific structuring element (e.g., what is a good summary) help learners using these facilitation features.

In addition to the structural information regarding individual notes, a script usually describes the order in which certain types of contributions should come. For example, in a debate a claim is usually countered or agreed on before a new claim is made. This sequencing can be done both by restricting the users' choices and by encouraging (or forcing) them to act in a certain way. In synchronous net-based discussion, it is important to keep the flow of discussion going. A script could therefore prompt users to act and contribute according to their current role in order to keep the discussion going. For example, a user in the role of a critic may be prompted to give constructive criticism after another user has stated a claim. In asynchronous discussion, sequencing may be implemented by allowing only certain types of notes as responses to previous notes, or to restrict the number of new topics while existing topics are unresolved. The sequence and structure of discussions, together with the assignment of roles, has to be specified by the designer of the script.

Different scripts vary regarding their degree of coercion, that is, the extent to which they force users into specific actions (Dillenbourg, 2002). This degree of coercion is usually implemented in the user interface, more precisely, in the specific instructions or choices presented to the users. However, once learners begin to internalize a script, it is useful to reduce the level of coercion and allow the learners to make more decisions themselves. This could be done by *fading out* explicit instructions or choices, according to the level a user has reached. Such a script would of course need a mechanism to adapt to the users' progress, which can be done before starting a script or during the runtime of the script. Ways of adaptation can range from simply counting the number of times a learner has used a certain structure in the current session to complex learner profiles keeping track of the learning process over a long period of time.

There are several existing applications that support the structuring and/or sequencing of net-based discussions. The learning protocol approach by Pfister and Mühlpfordt (2002) supports synchronous communication in chats with typed contributions, explicit references to previous messages, role assignment, and message sequencing. In first experiments, positive effects of learning protocols on knowledge acquisition in two domains could be proven. ACT (Gogoulou, Gouli, Grigoriadou, & Samarakou, 2004) is a synchronous communication tool which supports chats with sentence openers and communication acts (note typing). These communication scaffolding tools are adapted to the specific roles of the users and the educational objective of the collaborative learning activity. With ACT it is not possible to make references to artifacts or to anchor contributions. Further systems include CaMILE (Guzdial & Turns, 2000) and DEGREE (Barros & Verdejo, 2000). Our approach can be seen as an extension and adaptation for time-dependent artifacts. A scripted anchored discussion supports message typing as well as message prompts, and references are made to parts of the time-dependent artifact under discussion. Also, adaptation (fading out/in) of structuring elements is supported. The set of structuring elements used by the students may depend on the learning content and also on the study level of the learners. In the extended *aofJSync* Player, the instructor can define the structure types in a XML document according to a XSD. Each structuring element in this file is determined by a title, a category like cognition, communication and so forth, and instruction hints how to use this element and a fading level. The latter means that the lecturer can create several fading levels containing a subset of structuring elements which belong together and can be faded out together by a student. Alternatively, a user can disable each structuring element separately (see Figure 9-3). When using the annotation feature of the AOF system, students can only choose from the enabled elements structuring their notes. At the moment only the manual fading has been implemented. We are currently investigating the possibilities of automatic adaptation of the scaffolding in combination with sequencing features.

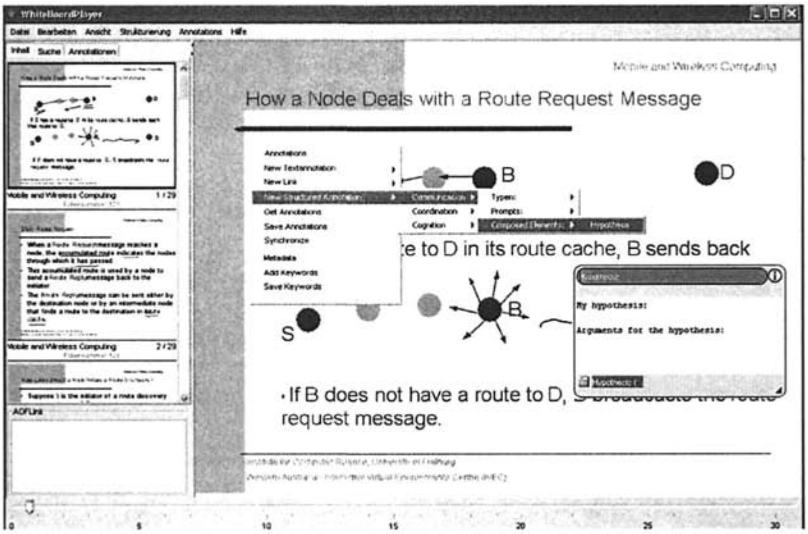


Figure 9-3: (1) Selection of a structuring element and (2) structured annotation with type and sentence opener.

4. A FORMAL MODEL FOR THE DESCRIPTION OF SCRIPTED ANCHORED DISCUSSION

A conceptual question on the technical level is how to formalize scripted anchored discussions for lecture recordings. Besides specific modelling languages such as IMS LD (cf. Miao, Harrer, Hoeksema, & Hoppe, this volume), there are several formalisms like Petri Nets and State Charts and modeling methods describing collaborative tasks. For instance, SeeMe (Herrmann, Hoffmann, Kunau, & Loser, 2004) is an approach for modeling socio-technical systems allowing the design of incompleteness and the self-regulation of systems. The practical uses of such a model include, among others, the design of software which can execute scripts, the possibility to exchange and re-use existing scripts, and a decoupling of scripts with their actual presentation to the learners.

For structuring and sequencing anchored discussions, nondeterministic finite state machines are sufficient (also see Haake & Pfister, this volume). Given a set Σ of roles of the participants, a set S of structuring elements like contribution types, prompts and composed elements as described before, and given a set of rules describing the sequential order of structuring elements depending on the roles and predecessor of contribution type, we can formalize a cooperation script for anchored discussions as a set of nondeterministic state machines. Each machine $M_k = (S, \Sigma, s_0, T, A)$ defines a part of the script in which the assignment of learners to script roles is consistent. A role switch for learners requires a new machine M_l and a mapping between the role assignment of M_k and M_l. For each machine, the alphabet Σ consists of the roles and the set S contains the structuring elements. The meaning of the machine being in a specific state s can be interpreted as "a new structuring element corresponding to s has just been produced." The transition function T: $S \times \Sigma \rightarrow P(S)$ is determined by the rules and sequences given by the script. The start state $s_0 \in S$ is a specific state corresponding to an "empty" discussion. We define a set A of accept states ($A \subset S$). If $A = S$ then M accepts every discussion process, that is, the script does not require any specific structure. The discussion can stop at every step. If $A \subset S$ then a discussion sequence is complete only if a certain sequential order of structuring elements is executed by the predefined roles in the discussion.

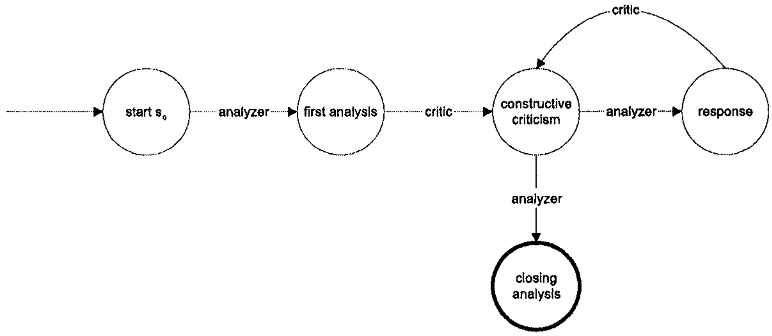


Figure 9-4: Representation of a scripted anchored discussion as a nondeterministic finite state machine.

Figure 9-4 illustrates the state machine representing a script which requires a participant with the *analyzer* role to do a first analysis, which is further described in Weinberger, Fischer, and Mandl (2002). Then, a learner with the *critic* role provides constructive criticism. From that point, the discussion can either be ended by a closing analysis by the analyzer, or continued by a response to the criticism. In that case, the critic is required to react again. The loop of constructive criticism and response may be repeated as often as necessary until a closing analysis ends the discussion. If the script demands a specific number of loops of a certain sequence of actions, it is necessary to expand the states according to the number of loops. For example, if the above script is modified in such a way that it requires at least one complete loop of constructive criticism and response, two additional states have to be introduced.

One drawback of current existing cooperation scripts is their fixed level of rigidity. Using cooperation scripts for a longer period of time can be strenuous and restrictive for learners. To overcome these obstacles the system should provide certain mechanisms to adapt the visibility of the cooperation script to the user needs and learning process. One strategy is the more a script is used by learners the more elements are faded out. Therefore the teacher defines several fading levels in the configuration of the scripted anchored discussion. A structuring element is faded out automatically if the learner has played the role as often as defined by the fading level. Alternatively, this process can be controlled by the learners themselves. They can hide certain structuring elements. One open issue is the coordination of fading between several users with the same role. This is the case if a learning group consists of more users than roles of the cooperation script.

The fading levels are not reflected in the formalization of the script, as the script itself essentially remains the same and only the way of its transparency to the learners is different. Thus, the environment handles the fading levels according to a specification made by the script designer.

In addition to the facilitation for learners, who can see the type of contribution immediately, the system can also collect some information about the discussion and may even react to it, for example by encouraging a learner who has only made one type of note to also contribute other types. Whether or not this categorization should be mandatory for each discussion note and whether keywords or sentence openers (or both) are used of course depends on the script and should be made configurable.

5. CONCLUSION AND FUTURE WORK

Collaborative learning around multimedia documents requires tools that take into account the special properties of time-dependent data. Examples of such documents are lecture recordings, which are used extensively at universities and other institutions. In order to facilitate online discussion of multimedia lectures, it is useful to allow learners to anchor each of their contributions in a fine-granular way at the relevant part of the original document and to provide different views to the discussion. Our Annotation Web Service integrated in the AOF system is one implementation of such a collaborative tool, allowing anchored discussion of time-dependent multimedia documents.

Anchored discussion can be combined with cooperation scripts. A script provides structuring and sequencing information such as message typing, valid sequences of types, and roles that are assigned to the participants. We have defined scripted anchored discussion and have demonstrated that a script can be modelled as a nondeterministic finite state machine, where the states correspond with structural elements and the alphabet consists of the roles.

In addition, it should be possible to fade out some of the explicit instructions or restrictions of a script in the course of the learning process, as learners internalize the script more and more.

Opportunities for further research and development are manifold. If scripts are combined with anchored discussion they can make explicit use of the anchoring facility provided by the system. For example, a script could not only allow but *demand* that learners back up their statements in an online discussion with a reference to the relevant part in the material or a previous comment. For instance, if one participant contradicts another participant's claim, he or she must anchor that contribution at the position of the original document that backs up the point. This is, of course, especially useful for scripts that are *meant* to teach debate and argumentation.

Because of the connection between discussion contribution and parts of the lecture recording, a scripted anchored discussion defines a personalized view of the lecture recording. Learners mark sections of the document which are important for the achievement of the learning objectives defined in the cooperation script. These sections as a whole can be replayed and, depending on the cooperation script, they constitute a *script view* of the document. It is an open issue if, for example, students learning with a script such as guided reciprocal peer questioning (King, 1990) or the MURDER script (O'Donnell & Dansereau, 1992) produce a coherent view of the document and if these views really contain the important and relevant parts of the lecture. If so, these script views of a lecture recording would represent a new "version" of the lecture, which is important for the specific learning outcome modeled in the script.

REFERENCES

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

Bargeron, D., Grudin, J., Gupta, A., Sanocki, E., Li, F., & Tiernan, S. L. (2002). Asynchronous collaboration around multimedia applied to on-demand education. *Journal of MIS, 18*(4), 117-145.

Barros, B., & Verdejo, M. F. (2000). Analysing student interaction processes in order to improve collaboration. The DEGREE approach. *International Journal of Artificial Intelligence in Education, 11*, 221-241.

Chong, N., & Sosakul, T. (2003). A framework for video-centered discussions on the web. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, (Vol. 1, pp. 620-626). Honolulu, Hawaii, USA: AACE Press.

Dillenbourg, P. (2002). Over-scripting CSCL: the risks of blending collaboprative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen, Open Universiteit Nederland.

Emond, B., Brooks, M., & Smith, A. (2001). A broadband web-based application for video sharing and annotation. *Proceedings of the 9th ACM International Conference on Multimedia* (pp. 603-604). Ottawa, Canada: ACM Press.

Fiehn, T., Lauer, T., Lienhard, J., Ottmann, T., Trahasch, S., & Zupancic, B. (2003). From lecture recording towards personalized collaborative learning. In B. Wasson, S. Ludigsen, & U. Hoppe (Eds.), *Designing for Change in Networked Learning Environments - Proceedings of the International Conference on Computer Support for Collaborative Learning 2003* (Vol. 2, pp. 471-475). Dordrecht, Boston, London: Kluwer Academic Publishers.

Goldman-Segall, R., Rao, S. V. (1998). A collaborative on-line digital data tool for creating living narratives in organizational knowledge systems. *Proceedings of HICSS 31 – International Conference on System Sciences* (Vol. 1, pp. 194-199), Kohala Coast, Hawaii, USA: IEEE Computer Society.

Gogoulou, A., Gouli, E., Grigoriadou, M., & Samarakou, M. (2004). Adapting the "communication-scaffolding" tools in a web-based collaborative learning environment. In L. Cantoni & C. McLoughlin (Eds.), *Proceedings of the World Conference on Educational Mul-*

timedia, Hypermedia and Telecommunications (pp. 1153-1161). Lugano, Switzerland: AACE Press.

Guzdial, M., & Turns, J. (2000). Effective discussion through a computer-mediated anchored forum. *The Journal of the Learning Sciences, 9*(4), 437-469.

Herrmann, T., & Kienle, A. (2003). KOLUMBUS: Context-oriented communication support in a collaborative learning environment. In T. van Weert & R. Munro (Eds.), *Informatics and the Digital Society. Social, Ethical and Cognitive Issues* (pp. 251-260). Boston: Kluwer Academic Publishers.

Herrmann, T., Hoffmann, M., Kunau, G., & Loser, K. (2004). A modeling method for the development of groupware applications as socio-technical systems. *Behaviour & Information Technology, 23*(2), 119-135.

King, A. (1990). Enhancing peer interaction and learning in the classroom through reciprocal questioning. *American Educational Research Journal, 27*, 664-687.

Lauer, T., Müller, R., & Trahasch, S. (2004). Web technologies and standards for the delivery of recorded presentations. *Proceedings of ITHET 04 – 5th IEEE International Conference on Information Technology Based Higher Education and Training (ITHET)*. Istanbul, Turkey: IEEE Press.

Lauer, T., & Ottmann, T. (2002). Means and methods in automatic courseware production: Experience and technical challenges. *Proceedings of the World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education* (pp. 553-560). Montréal, Canada: AACE Press.

Müller, R., & Ottmann, T. (2000). The "Authoring on the Fly" system for automated recording and replay of (tele)presentations. *ACM/Springer Multimedia Systems, 8*(3), 158-176.

Müller, R., & Ottmann, T. (2003). Content production and "the essence of simplicity". *Proceedings of ITHET 03 - 4th IEEE International Conference on Information Technology Based Higher Education and Training* (pp. 784-787). Marrakech, Morocco: IEEE Press.

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

Pimentel, M., Ishiguro, Y., Kerimbaev, B., Abowd, G., & Guzdial, M. (2001). Supporting educational activities through dynamic web interfaces. *Interacting with Computers, 13*(3), 353-374.

Pfister, H., & Mühlpfordt, M. (2002). Supporting discourse in a synchronous learning environment: The learning protocol approach. In G. Stahl (Ed.), *Proceedings of the Conference on Computer Supported Collaborative Learning* (pp. 581-589). Hillsdale, NJ: Lawrence Erlbaum Associates.

Plötzner, R., Philipp, J., & Oestermeier, U. (2003). Organizing and externalizing activity structures in distributed settings. In F. Hesse & Y. Tamura (Eds.), *Proceedings of the Joint Workshop of Cognition and Learning Through Media-Communication for Advanced E-Learning* (pp. 174-179). Berlin, Germany: Deutsch-Japanisches Zentrum.

Schmidt, K., & Bannon, L. (1992). Taking CSCW seriously. Supporting articulation work. *Computer Supported Cooperative Work (CSCW). An International Journal, 1*(1-2), 7-40.

Suthers, D. (2001). Collaborative representations: Supporting face to face and online knowledge-building discourse. *Proceedings of the 34th Hawaii International Conference on the System Sciences* (p. 4016). Maui, Hawaii, USA: IEEE Computer Society.

Trahasch, S., & Lauer, T. (2005). Scripted anchored discussions of e-lectures. *Proceedings of E-Learn 2005 – AACE World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 2436-2443), Vancouver, Canada: AACE Press

ULI (Collaborative University Education in Computer Science) Website. Retrieved May 30, 2006, from http://uli-campus.de/

VIROR (Virtual University in the Upper Rhine Valley) Website. Retrieved May 30, 2006, from http://www.viror.de/

Weinberger, A., Fischer, F., & Mandl, H. (2002). Fostering computer supported collaborative learning with cooperation scripts and scaffolds. In G. Stahl (Ed.), *Computer support for collaborative learning: foundations for a CSCL community. Proceedings of the International Conference on Computer Support for Collaborative Learning (CSCL) 2002* (pp. 573-574). Hillsdale, NJ: Lawrence Erlbaum Associates.

Zupancic, B., & Horz, H. (2002). Lecture recording and its use in a traditional university course. *ACM SIGCSE Bulletin (Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education), 34*(3), 24-28.

Chapter 10

FLEXIBLE SCRIPTING IN NET-BASED LEARNING GROUPS

Jörg M. Haake[1] and Hans-Rüdiger Pfister[2]
[1]*FernUniversität, Hagen;* [2]*Universität Lüneburg*

Abstract: CSCL scripts facilitate cooperative learning by constraining the activities of co-learners and thereby supporting coordination between distributed co-learners as well as guiding co-learners through the collaborative learning process. So far, such scripts have been encoded in CSCL environments and their tools. This made flexible adaptations of scripts an expensive task, which hinders experience-based improvements of CSCL scripts. In this chapter, we present a formal model of CSCL scripts and show how it can be used to help teachers and designers develop, adapt and experiment with CSCL scripts. In our approach, a script is represented as an extended finite state automaton, which is used to control the user interface and the possible activities in a web-based CSCL environment. We distinguish between atomic scripts, which support a specific collaborative learning activity, and composite scripts, which support a complex collaborative learning task through a sequence of atomic or composite scripts. Scripts can be created by a two-step process: defining atomic CSCL scripts, and linking existing scripts into a composite script for the overall learning activity. This approach enables the definition and reuse of CSCL scripts as well as their adaptation to learning groups and learning situations.

1. BACKGROUND

While in face-to-face (ftf for short in the following) cooperative learning the coordination among participants and their contributions is facilitated via non-verbal cues such as gestures, facial expression, prosody, and other kinds of informal communication, distributed net-based learning groups may benefit by explicit coordination efforts among learners, especially if they are working synchronously and are using a restricted form of communication such as text-based chat (Herring, 1999). Since learners and teachers are not accustomed to distributed collaboration, this adds considerable cognitive overhead to the task of learning. On the other hand, the advantages of cooperative or collaborative learning (we will use both terms synonymously) in general have been widely acknowledged (Fischer, 2002; Slavin, 1992; King, this volume).

One approach to this problem is to use computer-supported collaboration scripts (CSCL scripts), which control the user-interface and activities available to the learners according to the sequences of activities defined in the script. In general, the notion of a script serves as a highly generic term to conceptualize the regularities imposed on actions of individuals or groups during collaboration, and these regularities are usually based on some pedagogical or didactical rationale. In the context of cooperative learning, scripts are used to structure instructional situations in order to improve learning outcomes of individual participants as well as of the group as a whole (Hall, Dansereau, & Skaggs, 1990; O'Donnell, 1999; Pfister, Mühlpfordt, & Müller, 2003; King, this volume). For net-based cooperative learning scripted cooperation might be especially helpful, since virtual communication and collaboration quite often suffer from incoherent discourses and lack of coordination (Herring, 1999; Jucks, Paechter, & Tatar, 2003).

Current approaches to CSCL scripts can be categorized into informal and formal approaches. Informal approaches informally provide a collaboration procedure through instructions (which can be ignored or adapted by the learners). Formal approaches provide a rigid, predefined definition of the collaboration procedure used within the group (see also Hesse, this volume). Such a procedure is encoded formally into the CSCL environment, which enforces that the learners follow the prescribed procedure. The problem with the formal approach is that such CSCL scripts are difficult to define and to implement. Consequently, not much practical experience with formal CSCL scripts has been achieved.

This chapter addresses the above problem by introducing the concept of flexible scripting. Flexible scripting enables teachers to easily define complex CSCL scripts by reusing and linking already existing scripts, and to execute such a CSCL script in a collaborative learning environment. Here, basic scripts (which are either created by software developers or by teachers using an end-user programming environment) are taken as atomic building blocks. A framework for the definition and execution of flexible scripts is provided. We argue that such an approach facilitates quick, effortless, and inexpensive experimentation with and improvement of CSCL scripts, and may thereby lead to increased usability of distributed CSCL situations.

2. BASIC CONCEPTS AND ASSUMPTIONS

Scripted cooperation, originating in the work of Dansereau (1988), O'Donnell and Dansereau (1992), and O'Donnell and King (1999), can be defined as a prescriptive set of rules or instructions, independent of the knowledge domain, how a group of learners should proceed when learning together (Dillenbourg & Jermann, this volume; King, this volume). The idea of a script goes back to Schank and Abelson (1977) who used it as an explanatory cognitive structure which determines how people act in specific prototypical situations, such as in a restaurant; the notion of a script has later been extended to denote the central representational structures of human memory (Schank & Abelson, 1995). From this theory immediately follows the assumption that the process of learning, including its cognitive and social aspects, as well as the process of preserving and recollecting knowledge, should be enhanced if an appropriate script can be instantiated. Hence, on the most general level, a cooperation script for learning should define – externally or internally – the sequence of relevant activities to be performed in order to attain a learning goal (Gagné, Briggs, & Wager, 1992). In the framework of CSCL, this happens in the social context of a group of learners, possibly including a teacher or tutor, and thus needs to comprise rules for social interaction, especially rules for discourse (Clark & Brennan, 1991; Slavin, 1995).

From an applied perspective, CSCL scripts are primarily considered as support devices enabling learners to stick to an effective pattern of collaboration more easily and to obtain improved learning outcomes. There are, however, many variants how the support can be achieved on a practical level. In net-based scenarios, simple scripts can be carried out by the learners themselves by agreeing on a kind of social contract; this, however, is viable only for scripts of low complexity and for learning groups with high motivation and self-control. The most frequent kind of script implementation is via a human tutor or moderator, assisted by instructional guidance (O'Donnell, 1999). In this chapter we focus on technological implementation, that is, CSCL scripts are partly or completely implemented into the learning environment and guide the learning process implicitly by providing specific tools for and imposing particular constraints on the process of cooperative learning. From this technical point of view, Jermann, Soller, and Muehlenbrock (2001) distinguish three types of support systems: *mirrors*, reflecting the learners' activities in the interface, *monitors*, processing and presenting aggregated interaction data such as communication patterns, and *advisors*, giving explicit advice to the learners based on a theoretical model. We argue that we should further distinguish *enabling* systems, which merely provide opportunities for cooperation (e.g., NetMeeting), and genuine *supporting* systems (e.g., the learning protocols in Pfister & Mühlpfordt, 2000; Argue-Graph in Dillenbourg, 2002), which arrange the learning process according to some pre-defined structure and more or less oblige the participants to comply with this arrangements (Pfister, 2005).

Currently, supportive CSCL scripts are almost exclusively implemented within the collaborative learning environment as *hard-wired* tools or applications, that is, tailored according to one specified purpose, one didactical rationale, and one interface design (Haake & Schümmer, 2003a, 2003b; Hesse, Garsoffky, & Hron, 1997; Hesse & Hron, 1999; Hoppe, Gassner, Mühlenbrock, & Tewissen, 2000; Hron, Hesse, Cress, & Giovis, 2000; Wessner & Pfister, in press; Dillenbourg & Jermann, this volume). Hence, they can rarely be simply reused or adapted in a flexible way to different groups or new learning situations. This poses a severe limitation on the extensive use of CSCL scripts, because any special situation has its own affordances and requirements and needs at least some modification to suit the needs of the learning group at hand.

At present, there is some empirical confirmation on the effectiveness of special scripts in CSCL environments, though evidence is mixed. Pfister and Mühlpfordt (2002), Pfister et al. (2003), and Pfister, Wessner, Holmer, and Steinmetz (1999) provide evidence on an increase in learning performance when using a so-called *learning protocol* for explanations, that is, a structured text-based learning discourse which supports the exchange of questions and explanations among learners and a tutor and learners. It was shown that especially the requirement to indicate one's reference when delivering a contribution enhances performance on a knowledge test (Pfister et al., 2003). However, this result was restricted to groups of learners greater than three and to the knowledge domain of scientific facts; with respect to a philosophically oriented knowledge domain the learning protocol had no effect. Another instantiation of a learning protocol intended to support the co-construction of text summaries by a group of learners yielded mixed results; the reduced effectiveness probably being due to the increase in cognitive load by handling the complex user interface. The findings of Pfister et al. (2003) are in line with results from Weinberger, Fischer, and Mandl (2003), Weinberger and Mandl (2003), Weinberger et al. (this volume), and Ertl, Reiserer, and Mandl (2002), who implemented cooperation scripts with a video conferencing scenario (see also Ertl, Kopp, & Mandl, this volume). These studies also emphasize the specificity of scripted cooperation as employed in distributed net-based learning scenarios so far; what is needed is a framework to construct and implement effective CSCL scripts in a flexible way which can adapt to the requirements of a large class of learning scenarios without imposing unnecessary demands on the side of the users, that is, the learners.

3. SUPPORTING FLEXIBLE SCRIPTING

A CSCL script defines a collaboration procedure to be employed by a learning group when solving a learning task. It does so in terms of defining (or designing, or limiting) the possible interaction among learners. In distributed CSCL settings, such interaction is mediated through the collaborative learning environment. Therefore, the CSCL script expresses possible interactions between learners through defining possible interactions of the learners with their collaborative learning environment.

It is important to note that a CSCL script does not define or model how learners learn internally! Rather, the purpose of a CSCL script is to help coordinate group interaction by suggesting or limiting possible courses of interaction during a collaborative learning episode. As pointed out in the previous section, better coordinated collaboration should enhance performance by inducing cognitive processes that lead to better learning outcomes (King, this volume).

Support for coordination might be directed at two levels (Fischer et al., this volume): Firstly, coordination on the macro-level organizes larger group learning phases (individual, collective, collaborative) in a learnflow (Wessner, Dawabi, & Haake, 2002). For this, a CSCL script needs explicit knowledge about the learning process, its state, and its progress. Secondly, coordination on the micro-level orchestrates single learning activities or contributions within a single group learning phase. Such coordination may be facilitated through domain specific guidance (e.g., by providing instructions for formulating and organizing the content to be learned), social contracts (e.g., conventions for use of chat), or through adaptive user-interfaces (e.g., floor control, process control, adaptation of the user interface to facilitate only permitted activities or contributions).

The construction of CSCL scripts facilitating such coordination is a difficult task. It requires an integration of the didactical approach (which suggests how appropriate group behavior can be achieved) and the technical implementation in the CSCL environment (which provides the tools and affordances to carry out such group interactions). Usually, these aspects are meshed in the code of the CSCL environment, and thus are difficult to change. For example, Pfister et al. (2002) describe the implementation of two dedicated groupware tools offering support for exactly one CSCL script (cooperative text processing script, explanation script) each. Dillenbourg (2002) describes the ArgueGraph-Skript, which has been implemented as a Postnuke module (ArgueGraph 2003).

This leads to the problem of adapting CSCL scripts to the needs of individual groups and learning situations. Hard-coded implementations of scripts require extensive programming to achieve these adaptations. This makes reuse of scripts a costly approach, and consequently experimentation with and reuse of scripts is low.

Our approach to solve these problems is based on the idea that the collaborative learning environment should provide affordances so that a distributed learning group exhibits the intended learning behavior. Intended learning behavior can be expressed in a CSCL script (on the macro and micro level). This led to the concept of composite CSCL scripts (i.e., macro scripts containing other macro or micro scripts), just as structured programs contain procedures, which call other procedures. We distinguish between atomic CSCL scripts supporting a single collaborative learning activity (i.e., micro script), and a composite CSCL script, which consists of a sequence of CSCL scripts and which aims at supporting a complex collaborative learning task.

Since the learners interact with each other via executing operations provided by the CSCL environment, the availability of such operations must be controlled by the script. In addition, information about the status of the script (i.e., the state of the learning process) needs to be displayed. These two types of elements (available operations, and extent of status display) define what the script can change to provide those affordances in order to support the learners to follow the CSCL script.

An editor for composite CSCL scripts helps teachers to define sequences of scripts. This editor facilitates reuse of atomic and composite CSCL scripts and allows adaptation and improvement of CSCL scripts. An execution environment for composite CSCL scripts on top of a CSCL platform supports easy set up and execution of composite CSCL scripts. The execution environment converts the CSCL tools into tools providing the affordances required by learners following the CSCL script.

Benefits of this approach include the easy definition of a composite script from existing scripts, easy adaptation of composite scripts to new learning situations and groups, and easy set up and execution of scripts in the collaborative learning environment. However, this approach still requires means for defining simple (atomic) scripts, for example through software developers or through end-user programming scripting environments.

4. DEFINITION OF COMPOSITE CSCL SCRIPTS

We distinguish between atomic CSCL scripts aiming at coordination at the micro level (during a single collaborative learning phase or activity) and composite CSCL scripts aiming at coordination at the macro level (during the entire collaborative learning task composed of different phases or activities).

4.1 Atomic CSCL scripts

An *atomic CSCL script* defines the collaboration procedure employed by collaborative learners during a specific learning activity such as a question-answer game. The basic idea of defining an atomic CSCL script is to define types of actors and possible sequences of their contributions (in terms of operations that they may perform in the collaborative learning environment). Consider the example of a Question-Answer-Script (King, this volume). This CSCL script is characterized by two roles (Questioner, Responder) and four activities: The Questioner may *start* a question-answer round, *ask* a question or *quit* the script. The Responder may *answer* a given question. In addition, after one question-answer pair, the roles are switched to ensure equal exposure of the learners to both roles. In order to support teachers (or tutors or any other individual responsible for instructional design) in defining such a CSCL script for an existing cooperative application, we propose to define for every type of user (i.e., role) the permitted behaviors in terms of macros or operations (i.e., sequences of atomic operations belonging to one semantically meaningful contribution within the script). In the example of the Question-Answer-Script, we need to define three macros for the Questioner role (for the activities *start, ask,* and *quit*), and one macro for the Responder role (for the activity *answer*).

Our approach of defining a CSCL script consists of four steps:

1. *Definition of the roles* (i.e., types of actors in the script): For each role, the teacher defines every permitted semantic action (type of contribution) as a macro having a unique semantically meaningful name (such as *ask*). As a result of this step, we get a set of distinct roles and corresponding sets of macros for each role.
2. *Definition of the states and the transitions between the states* of the script: For each role, the teacher defines possible states (such as *Placing Question* and *Awaiting Answer* for the Questioner role in Figure 10-1) and connects these via transitions (arrows in Figure 10-1). Every transition that a learner having this role may cause must be labeled with the corresponding macro name, for example the transition of the Questioner from *Placing Question* to *Awaiting Answer* is caused by the execution of the *ask* macro. One can imagine this step as constructing UML activity lanes for each role, which define states and how users of that role may use operations (macros) to switch between states. As a result of this step, we get for each role a separate finite state automaton with separate sets of states and macros.

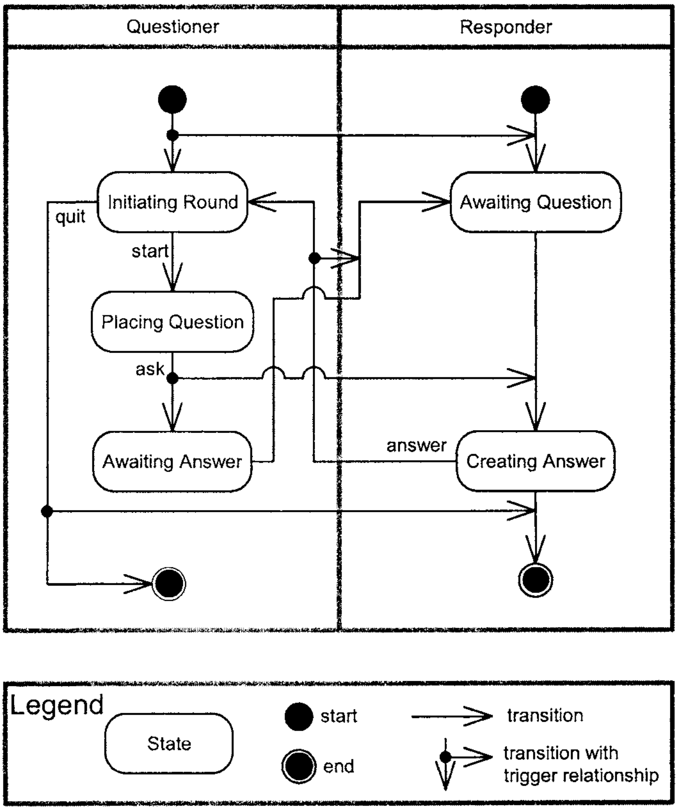


Figure 10-1. Finite state automaton for the Question-Answer-script.

In the case of our Question-Answer-Script, we would thus construct one part of the automaton for role Questioner (its transitions labeled with macros *start*, *ask*, or *quit*), and one part of the automaton for role Responder (its transitions labeled with the macro *answer*).The teacher must define the initial state of the script by identifying initial states of each role. The execution of the script may then start by assigning users to each role, marking the initial states as active, and then allowing only execution of those actions (macros), which are labels of transitions originating from the current states. All other operations (macros) are not enabled in this state. When users perform possible actions (macros) they cause transitions, which will change the current states and thereby potentially change the operations (macros) available to the users.

3. *Definition of collaboration behavior*: Here, the teacher needs to connect the individual behaviors of the roles into an orchestrated network. Thus, the teacher has to define two types of connections:

a) An activity of Role 1 may cause a state transition of Role 2. In the Question-Answer-Script a state transition in the Questioner part of the automaton happens when the *ask* macro is executed. The Questioner part of the automaton reaches a *Awaiting answer* state, and simultaneously the Responder part of the automaton executes a state transition from the *Awaiting Question* state to the *Creating Answer* state. Only in this state the responder may answer the previously created question. Thus, a transition in one part of the automaton may trigger a transition in another part of the automaton. We therefore need to add *trigger relations* to transitions. These relations point from the source transition to a destination transition (which is automatically executed after the source transition was executed). In Figure 10-1, the trigger relation is denoted by adding an arrow starting from a bold point in the source transition and pointing to the destination transition. The destination transition does not carry a macro name, as it cannot be triggered through executing a macro by the user having Role 2. Its sole purpose is to synchronize state transitions between parts of the automaton.

b) An activity of one role may lead to a role change (i.e., the user having Role 1 may now get assigned Role 2, and vice versa). For example, in the Question-Answer-Script we need to switch roles after one question-answer pair has been finished. For this purpose, the transition labeled *answer* from *Creating Answer* in the Responder part of the automaton to *Initiating Round* in the Questioner part of the automaton initiates the role change. It uses a trigger to enforce the corresponding transition from *Awaiting Answer* in the Questioner part of the automaton to *Awaiting Question* in the Responder part of the automaton. As a result, both users switched roles and a new question-answer-round can begin.

In computer science, a finite state automaton or finite state machine (Hopcroft & Ullman, 1979) consists of a set of states, a set of input events, a set of output events, and a state transition function. The state transition function takes the current state and an input event and returns the new set of output events and the next state. Some states may be designated as *terminal states*. In the following, we will use this concept to define our notion of a finite state automaton that is used to control the behavior of users of an atomic CSCL script (see also Lauer & Trahasch, this volume). Following the above approach, an atomic CSCL script is defined as an extended finite state automaton represented by a 8-tupel (Roles, Actions, States, Transitions, Start, End, InputDocument, OutputDocument) with

- Roles is a finite set of distinct role ids (i.e., representing the types of actors in the script).
- Actions is a finite set of distinct macro ids (i.e., representing the macros a user may perform using the collaborative learning tool under this script).
- States is a finite set of distinct state ids (i.e., representing the different states, which the script can take while being executed).
- Transitions defines state transitions from the current state (a state taken by a role) of the script to the next state (a state taken by a role) of the script. A transition may cause another transition to be executed. Thus, Transitions is defined as a subset of the Cartesian product of (Roles × States) × Actions × (Roles × States) × (Powerset(Transitions) ∪ nil). The reader should note that this definition allows transitions to cause a role change (e.g., in our example the transition caused by the answer macro of the user having the Responder role leads for this user to a state where the user has now the Questioner role, while the trigger relation (see 3(a) above) of this transition executes another transition putting the user having the Questioner Role into the Responder role. Since multiple roles may be affected by an action the trigger relation may activate multiple transitions). With this definition, a user can only have exactly one role. This is not a limitation, as we can model any combination of roles as new roles, with appropriate new states and transitions added to the automaton.
- Start, End are distinct subsets of (Roles x States) representing the starting and ending states of the script (i.e., what are the initial and terminal assignments of roles to states).
- InputDocument, OutputDocument are instances of documents used as initial input and final output of the script.

The semantics of an atomic CSCL script is defined by its transitions. The outgoing transitions from a state define which macros are available for use in this state. Users can only select a macro available in the current state. The execution of the macro causes the associated transition to be executed, and thus the automaton assumes a new state. The overall semantics of an atomic CSCL script is therefore defined by all possible sequences of state transitions starting from any state in Start and ending when all states in End are achieved.

4.2 Composite CSCL scripts

A composite CSCL script coordinates learner interaction throughout an entire collaborative learning task composed of different phases or activities (macro level). We propose to define a composite CSCL script as a sequence of (atomic or composite) CSCL scripts. This allows teachers to reuse CSCL scripts for specific learning activities or tasks in the context of a larger collaborative learning task. Figure 10-2 shows an example of a nested composite script for learning how to understand vulcanism. The script consists of an atomic Question-Answer script on answering questions about vulcanism and a subsequent composite script on analyzing vulcanism; this composite script itself consists of two atomic scripts (brainstorming and clustering). Rectangles indicate scripts, rounded rectangles start and end states within sub automatons of atomic scripts, black circles start and end states and small rectangles within atomic scripts indicate states. Arrows represent transitions.

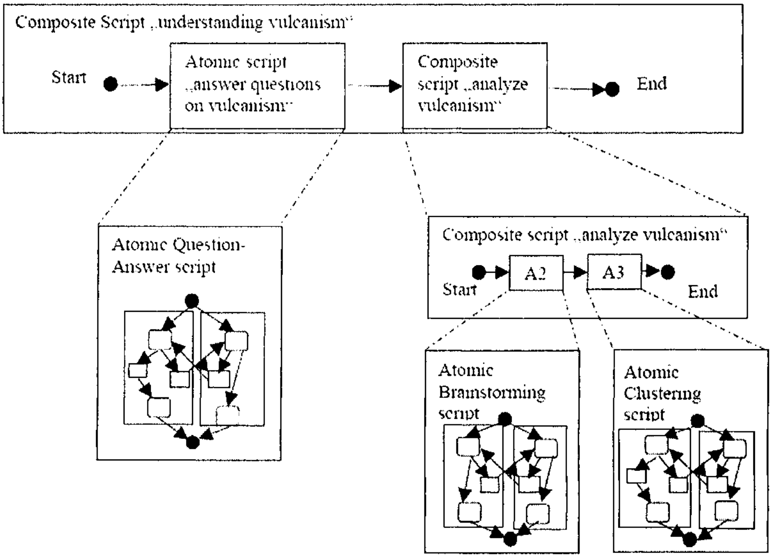


Figure 10-2. Example of a nested composite script for learning how to understand vulcanism.

As each atomic CSCL script has defined start and end states, the transition from a current script to a successor script can be achieved by activating the start states of the successor script when the predecessor script terminates. The start state of the composite CSCL script is defined by the start state of the first script, and the terminal state of the composite script is defined by the terminal state of the last script. Through this definition, the execution semantics of nested composite CSCL scripts is clearly defined. Thus, execution of a composite CSCL script begins with executing the first script. When this script terminates, the next script in the sequence is executed. The execution of the composite script terminates when the last script of the sequence terminates.

To allow reuse of results or artifacts created in a predecessor script, we use two variables in a script (namely, InputDocuments and OutputDocuments) to provide successor scripts with the resulting data from a predecessor script during the execution of a composite CSCL script.

One could argue that more powerful control statements are needed to support more complex scripts (e.g., including "if-then-else" or loops). In many cases, however, simple sequences are sufficient and can be supported by a simple user interface. If more complex cases are needed, more complex control structures can easily be added. The corresponding editor for composite scripts would then come close to a general purpose programming environment and consequently more skills would be required by the users.

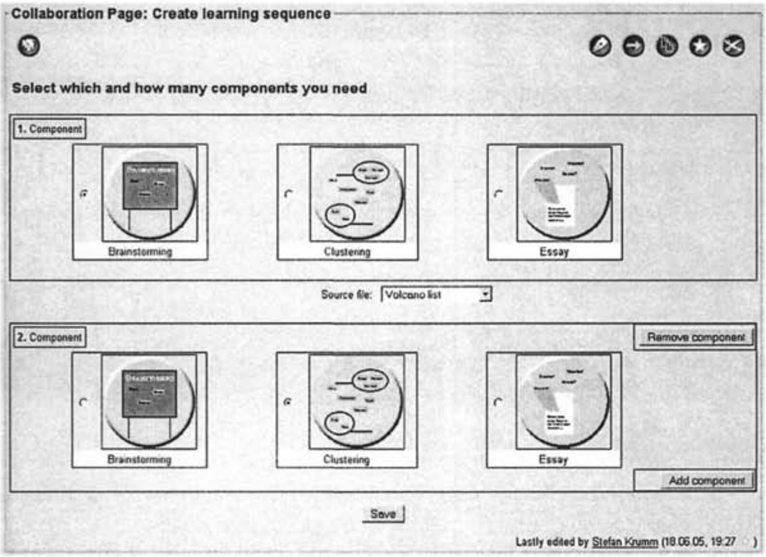


Figure 10-3. Creation dialogue of a new composite CSCL script for understanding vulcanism (see section 4.2).

5. EDITING FLEXIBLE CSCL SCRIPTS

5.1 Editing atomic scripts

Atomic CSCL scripts can be created in two ways:

Firstly, developers can implement atomic scripts within the collaborative learning environment. In this case, the software component implementing the script only needs to export functions used by the execution environment to start and initialize the script.

Secondly, designers or teachers could use an end-user programming environment for CSCL scripts to construct the finite state automaton representation themselves. In this case, they either need to define possible actions (macros) by using a macro recorder approach or they need access to the names of operations provided by the collaborative learning environment. The execution environment will then call such macros (operations) of the collaborative learning environment when the respective transition is executed.

5.2 Editing composite scripts

A new composite CSCL script can simply be created in the collaborative learning environment. When editing the new composite script, teachers select in a simple editor dialogue existing scripts and bring them into a fixed order (see Figure 10-3). Instances of this script can then be created and executed at run-time.

Figure 10-3 shows how the user can create composite scripts. The available component scripts are shown with an icon representing each script's type. The teacher starts with a script containing one component (in the upper part of Figure 10-3 labeled *1. Component*, the teacher selected a *Brainstorming* script). He connects this component with an input document called *Vulcano list* that contains required initial data (see input field called *Source file* in Figure 10-3). In the lower part of Figure 10-3, the teacher can add or remove additional components. The components are connected in the order in which they were added. The connection defines the temporal sequence in which the components will be executed as well as the data flow between the components. In the example, the result of the brainstorming script is used as the input document for the clustering script.

6. EXECUTION OF A CSCL SCRIPT

In the following we assume that each learner works with a local client of the CSCL environment. These clients communicate with a central server in order to coordinate their behavior (i.e., the automaton representing the script is maintained only at the server). Based on the above specification of a composite CSCL script, a run-time environment will use the corresponding graph of states and transitions to compose the user interface of the cooperative learning tool at each client in the following manner:

1. *Initialization*: For all roles (individual parts of the automaton with their own starting state) the environment computes the set of macros that are used as labels on outgoing transitions (such as *start* in Figure 10-1). These macros are executable in the current state of the individual part of the automaton. After assigning users to roles, the environment can provide a dedicated user interface to each user based on his or her role (i.e., part of the automaton). It will make those macros available in the local client as, for example, buttons with appropriate labels at the user interface.
2. *Execution*: When an operation is executed (e.g., the user of the local client pressed the button), the environment executes the macro at the local client and performs the state transition in the corresponding part of the automaton. Then, any triggered transitions in other parts of the automaton are executed. These refer to changes at the remote clients of other users having the corresponding roles. For every client that experiences a state change, the new user interface (i.e., the enabled operations) must be recomputed in the server and sent to the clients (see above).
3. *Termination*: If a local client reaches an end state of its part of the automaton, this should be indicated at the user interface, as the participation of this user in the script ended.

Then, the next script in the sequence of the composite script can be activated.

During execution, the automaton of the script defines the user interface of each learner according to his/her role in the script. Transitions lead to state changes, which are reflected at the user interface of the learner associated with the new state. Thereby, actions of one user may affect the user interface of other learners. Finally, when the composite CSCL script terminated the artifacts produced by the learners in the script are stored in the final document produced by the last script and can be examined by a teacher.

In the example in Figure 10-4, the users are currently in the clustering script of the composite script. The result from the first component (brainstorming) was automatically used as input for the clustering (see the words in the upper part of the figure). Users can add a new cluster by providing a new cluster name and selecting (ticking) those words belonging to the new cluster. Selecting an existing cluster allows to change the selection of its words.

Since learners interact concurrently with their user interfaces, several users may cause transitions at the same time. Since we use a central server with transaction management, only one transition is executed - the other users may have to retry (if their action is still possible in the new state). This might cause a problem as learners may loose some input. However, this applies only if the new state does not allow for the same action anymore (which, obviously, depends on the nature of the script). If the teacher wants to avoid such effects a script which enforces strict turn-taking should be defined instead.

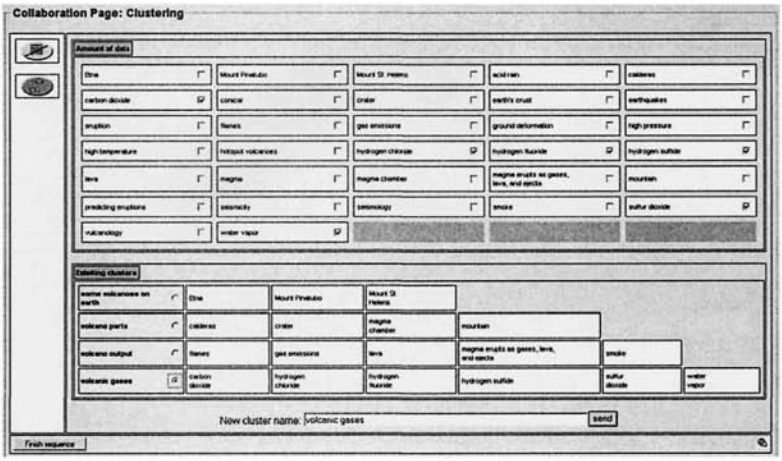


Figure 10-4. Sample screen dump of one user participating in the execution of the composite script defined in Figure 10-3

7. IMPLEMENTATION

We applied our approach to the CURE web-based CSCL environment (Bourimi, Haake, Landgraf, Schümmer, & Haake 2003; Haake, Haake, Schümmer, Bourimi, & Landgraf 2004; Haake, Schümmer, Bourimi, Landgraf, & Haake 2004; Haake, Schümmer, Haake, Bourimi, & Landgraf 2004). Users direct their web browser to a login page in the CURE server. CURE runs in a Servlet-Container (such as Apache Tomcat) and forwards web requests to the appropriate servlet (i.e., code that is executed in the web server). The servlet executes the appropriate code (possibly changing internal states of the web server) and returns an HTML page that is displayed by the browser. This way, servlets can dynamically change the user interface of the application by taking the internal state of the web server and the user input into account. Atomic scripts are implemented as objects in our server. Composite scripts are represented internally by a script object knowing its component scripts and their temporal order. In addition, each script object keeps track of the active state for each user.

The composite script editor is implemented as a servlet listing the existing component scripts and allowing users to compose a sequence of component scripts to a composite script. This includes that the user can select the desired components and link them in an execution sequence.

Composite scripts are executed by copying the script object and setting its active state to the start state of the top level script. User activities trigger servlet requests in our server, which are forwarded to the script object, which in turn modifies the script state according to the state transitions defined in the script object. As a result, a new user interface is computed and sent to the sender of the servlet requests (thus leading to a browser update). In addition, all transitions defined in a trigger relation are executed. Those will lead to state changes for other users, whose browsers will be notified about the change. Those browsers will then reload the current page from the server, which leads to an updated display.

If teachers notice any problems with the script, they can edit the composite script (or create a new version by copying and changing the current version of the script). Once the script definition has been changed, new users using the script will automatically get the new version. Thus, teachers can easily create, test and compare different alternatives for a composite script.

8. DISCUSSION

Our approach is based on modeling composite CSCL scripts as nested finite state automatons, which define for a user having a certain role and state what operations this user may execute and what user interface this user will see. State transitions with triggering relationships are used to define sequences of permitted operations.

Usually, CSCL scripts are encoded in the CSCL tool. Thus, they are developed by programmers and are usually hard to change. In our approach, the specification of a composite script can be changed at any time. Thus, experimentation with scripts becomes cheap and simple.

In the learning domain, IMS-LD (IMS-LD, 2003) allows the specification of teaching processes. An IMS-LD specification can then be executed by an execution engine such as CopperCore (CopperCore, 2004). However, IMS-LD describes collaboration in terms of activities, roles and the environment (incl. tools). The resulting specification expresses when certain resources are displayed for whom and when one can pass from one content to another. This coarse level of description does not allow more fine-grained specification of collaborative processes such as required in micro-level scripts. Furthermore, different synchronization behavior and awareness cannot be easily expressed. Here, the composite CSCL script approach is better suited.

Workflow management systems support the definition and execution of asynchronous workflows. Workflow designers specify in a workflow schema how a task is broken down into activities, which can be assigned to workers having a certain role, and what the data- and control-flow dependencies between activities are. At run-time, an instance of the workflow schema is created and users who may fill the needed roles are assigned to activities. The workflow management system ensures that the flow of activities follows the specified schema. The Flex-eL (Marjanovic & Orlowska, 2000) system applied workflow management to support individual learning. However, as workflow systems do not support synchronous group activities they can not be used to implement synchronous CSCL scripts.

Other computer science approaches to represent CSCL scripts include scripting languages and Petri-Nets. Scripting languages and scripting environments such as HyperCard (Apple, 1987) require programming capabilities from the users. The benefit of this approach is full control over the behavior of the application (to the extent of the functionality of the scripting language). However, such scripting languages and environments are usually focusing on single-user or multi-user/non-cooperative settings. Thus, synchronization between users may need to be explicitly programmed. This is a tedious and error prone task, requiring experts. Our approach limits the required capabilities of teachers and designers to the graphical language used for specifying automatons, and to the use of the specification environment for recording macros and specifying the scripts. This approach is feasible, since we assume that scripts limit available operations to meaningful sequences. Thus, our scripts only take functionality away – although the editing of macros allows the definition of composed operations (i.e., automatically executed sequences of operations). Furthermore, our use of the central web server ensures consistency in a concurrent run-time environment.

Alternative representations of automatons are Petri-Nets. Here, a graphical network of places and transitions with tokens assigned to places represent the states and state transitions. The benefit of Petri-Nets is their formal nature. The behavior of a Petri-Net can be mathematically analyzed and there are interpreters such as Trellis (Furuta & Stotts, 1994) that can execute a

Petri-Net based specification of a cooperative application. However, Petri-Nets are quite complex and difficult for non-experts to use. Our representation, in contrast, is sufficiently simple so that a team of teacher and designer can manage the complexity associated with the design of CSCL scripts.

9. CONCLUSIONS

CSCL scripts constrain and guide the behavior of distributed learners to potentially effective sequences of activities, and thereby facilitate cooperative learning. In our approach, we model permitted operations in a web-based CSCL environment by defining macros (e.g., through implementing, recording and editing). We define the roles and associated cooperative behavior by (1) creating parts of a finite state automaton for each role, (2) specifying in each part of the automaton the states and state transitions, which define what macros a user with that role in that state may execute, and (3) connecting the parts of the automaton via state transitions and triggering relationships, which model role switches and synchronization of state transitions. Furthermore, teachers may specify sequences of scripts for supporting more complex collaborative learning tasks. The resulting specification of atomic and composite CSCL scripts can then be executed in a CSCL environment. As a result of our approach, experimentation with variants from composite scripts becomes inexpensive and simple. Thus, more knowledge about efficient composite scripts can be collected. Furthermore, a composite script can now be adapted to the needs of specific groups and situations. This is the basis for supporting flexible scripting in net-based learning groups.

Cumulating experience is needed to determine these building blocks, and to develop a methodology facilitating reuse and transfer of knowledge among teachers and designers concerning the construction of efficient composite scripts. Further extensions also concern the integration of behavior logging, which would enable evaluation of learner behavior. An even higher degree of flexibility could be achieved by relaxing the deterministic control towards an adaptive control, induced by a dynamic evaluation of the ongoing learning process, the user behavior, and the preferences of the learners. Finally, the application of our approach to non web-based architectures is an open issue, as the distribution of functionality between clients and servers may affect the implementation of our approach in these systems.

REFERENCES

Apple (1987). Retrieved from http://www.apple.com/hypercard/

ArgueGraph (2003). Retrieved August, 2004, from http://tecfaseed.unige.ch/users/mourad/arguegraph/ArgueGraph.html

Bourimi, M., Haake, J. M., Landgraf, B., Schümmer, T., & Haake, A. (2003). Unterstützung für das 2-Ebenen-Tailoring bei CSCL. *Proceedings of DELFI 2003: 1. E-Learning Fachtagung Informatik* (pp. 280-289). Bonn: GI.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 127-149). Washington, DC: APA.

CopperCore (2004). Retrieved from http://CopperCore.org.

Dansereau, D. F. (1988). Cooperative learning strategies. In C. E. Weinstein, E. T. Goetz, & P. A. Alexander (Eds.), *Learning and study strategies: Issues in assessment, instruction, and evaluation* (pp. 103-120). Orlando, FL: Academic Press.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL?* Heerlen: Open Universiteit Nederland.

Ertl, B., Reiserer, M., & Mandl, H. (2002). Kooperatives Lernen in Videokonferenzen. *Unterrichtswissenschaft, 30,* 339-356.

Fischer, F. (2002). Gemeinsame Wissenskonstruktion - Theoretische und methodologische Aspekte. *Psychologische Rundschau, 53,* 119-134.

Furuta, R., & Stotts, P. D. (1994, October). Interpreted collaboration protocols and their use in groupware prototyping. In *Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94)* (pp. 121-131). New York: ACM Press.

Gagné, R., Briggs, L., & Wager, W. (1992). *Principles of instructional design* (4th ed.). Fort Worth: HBJ College Publishers.

Haake, J. M., Haake, A., Schümmer, T., Bourimi, M., & Landgraf, B. (2004). End-user controlled group formation and access rights management in a shared workspace system. In *Proc. of ACM CSCW 2004*, Chicago, November 6-10, (pp. 554-563). New York: ACM Press.

Haake, J. M., Schümmer, T., Bourimi, M., Landgraf, B., & Haake, A. (2004). CURE – Eine Umgebung für selbstorganisiertes Gruppenlernen. *i-com Zeitschrift für interaktive und kooperative Medien, 3*(2), 20-26.

Haake, J. M., Schümmer, T., Haake, A., Bourimi, M., & Landgraf, B. (2004). Supporting flexible collaborative distance learning in the CURE platform. *Proceedings of the Hawaii International Conference On System Sciences* (HICSS-37), IEEE Press.

Haake, J. M., & Schümmer, T. (2003a). Supporting collaborative exercises for distance learning. *Proceedings of the Hawaii International Conference on System Sciences, HICSS-36,* IEEE Press.

Haake, J. M., & Schümmer, T. (2003b). Supporting collaborative exercises for distance learning. In B. Wasson, S. Ludvigsen, & U. Hoppe (Eds.), *Designing for Change in Networked Learning Environments*, (pp. 125-134). Dordrecht: Kluwer.

Hall, R. H., Dansereau, D. F., & Skaggs, L. P. (1990). The cooperative learner. *Learning & Individual Differences, 2*(3), 327-336.

Herring, S. (1999). Interactional coherence in CMC. *Journal of Computer-Mediated Communication, 4.* Retrieved from http://www.ascusc.org/jcmc/vol4/issue4/herring.html

Hesse, F. W., Garsoffky, B., & Hron, A. (1997). Interface-Design für computerunterstütztes kooperatives Lernen. In L. J. Issing & P. Klimsa (Eds.), *Information und Lernen mit Multimedia* (pp. 252-267). Weinheim: Psychologie Verlags Union.

Hesse, F. W., & Hron, A. (1999). Dialogue structuring in computer supported synchronous discussion groups. *Advanced research in computers and communications in education. Proceedings of ICCE'99, 1*, 350-355.

Hopcroft, J. E., & Ullman, J. E. (1979). *Introduction to automata theory, languages and computations.* Reading, MA: Addison-Wesley.

Hoppe, H. U., Gassner, K., Mühlenbrock, M., & Tewissen, F. (2000). Distributed visual language environments for cooperation and learning - applications and intelligent support. *Group Decision & Negotiation, 9*, 205-220.

Hron, A., Hesse, F. W., Cress, U., & Giovis, C. (2000). Implicit and explicit dialogue structuring in virtual learning groups. *British Journal of Educational Psychology, 70*, 53-64.

IMS-LD (2003). IMS Learning Design Information Model. Final Specification, IMS Global Learning Consortium. Retrieved from http://www.imsproject.org/learningdesign/ldv1p0/imsld/_infov1p0.html

Jermann, P., Soller, A., & Muehlenbrock, M. (2001). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. In P. Dillenbourg, A. Eurelings, & K. Hakkarainen (Eds.), *Proceedings of the European Conference on Computer-Supported Collaborative Learning EuroCSCL-2001* (pp. 324-331). Maastricht: The Netherlands.

Jucks, R., Paechter, M., & Tatar, D. G. (2003). Learning and collaboration in online discourses. *International Journal of Educational Policy, Research, & Practice, 4*, 117-146.

Marjanovic, O., & Orlowska, M.E. (2000). *Making flexible learning more glexible.* IEEE International Workshop on Advanced Learning Technologies IWALT'2000, December 2000, New Zealand.

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

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

O'Donnell, A. M., & King, A. (Eds.). (1999). *Cognitive perspectives on peer learning.* Mahwah, NJ: Lawrence Erlbaum Associates.

Pfister, H. R. (2005). How to support synchronous net-based learning discourses: Principles and perspectives. In R. Bromme, F. W. Hesse, & H. Spada (Eds.), *Barriers and biases in computer-mediated knowledge construction* (pp. 39-57). New York: Springer.

Pfister, H.-R., & Mühlpfordt, M. (2002). Supporting discourse in a synchronous learning environment: The learning protocol approach. In G. Stahl (Ed.), *Computer Support for Collaborative learning: Foundations for a CSCL Community. Proceedings of CSCL2002 - Conference on Computer Supported Collaborative Learning, Boulder, Colorado* (pp. 581-589). Hillsdale, NJ: Lawrence Erlbaum Associates. Available at: http://www.cscl2002.org).

Pfister, H.-R., Mühlpfordt, M., & Müller, W. (2003). Lernprotokollunterstütztes Lernen - ein Vergleich zwischen unstrukturiertem und systemkontrolliertem diskursivem Lernen im Netz. *Zeitschrift für Psychologie, 211*, 98-109.

Pfister, H.-R., Wessner, M., Holmer, T., & Steinmetz, R. (1999). Evaluating distributed computer-supported cooperative learning (D-CSCL): A framework and some data. In *Proceedings of the 2nd International Conference on New Learning Technologies (NLT99)* (pp. 234-241).

Schank, R. C., & Abelson, R. P. (1977). Scripts, plans, goals, and understanding: An inquiry into human knowledge structures. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R. C., & Abelson, R. P. (1995). Knowledge and memory: The real story. In R. S. Wyer Jr. (Ed.), *Advances in Social Cognition* (Vol. VIII, pp. 1-85). Hillsdale, NJ: Lawrence Erlbaum Associates.

Slavin, R. E. (1992). When and why does cooperative learning increase achievement? Theoretical and empirical perspectives. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interaction in cooperative groups: The theoretical anatomy of group learning* (pp. 145-173). New York: Cambridge University Press.

Slavin, R. E. (1995). *Cooperative learning: Theory, research, and practice* (2nd ed.). Needham Heights, MA: Allyn and Bacon.

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

Weinberger, A., Fischer, F., & Mandl, H. (2003). Gemeinsame Wissenskonstruktion in computervermittelter Kommunikation: Wirkungen von Kooperationsskripts auf den Erwerb anwendungsorientierten Wissens. *Zeitschrift für Psychologie, 211*(2), 86-97.

Weinberger, A., & Mandl, H. (2003). Computer-mediated knowledge communication. *Studies in Communication Science, 3*, 81-105.

Wessner, M., Dawabi, P., & Haake, J. M. (2002). L3 - An infrastructure for collaborative learnflow. In G. Stahl (Ed.), *Proceedings of the Conference on Computer Supported Collaborative Learning (CSCL) 2002*, (pp. 698-699). Hillsdale, NJ: Lawrence Erlbaum Associates.

Wessner, M., & Pfister, H.-R. (in press). Points of cooperation: Integrating cooperative learning into web-based courses. In H. U. Hoppe, M. Ikeda, & H. Ogata (Eds.), *New technologies for collaborative learning*. Dordrecht: Kluwer.

Chapter 11 – Discussion

ROLES OF COMPUTATIONAL SCRIPTS

Daniel D. Suthers
University of Hawaii

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

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

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

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

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

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

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

1. OFFLOADING TASKS

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

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

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

2. FOSTERING PRODUCTIVE INTERACTIONS

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

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

2.1 Setting up the conditions for interaction

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

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

2.2 Anchored discussion as a context for scripting

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

2.3 Micro-managing interaction

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

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

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

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

3. COMMUNICATING DESIGNS

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

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

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

4. SCRIPTS AS RESOURCES

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

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

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

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

REFERENCES

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

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

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

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

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

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL.* (pp. 61-91). Heerlen: Open Universiteit Nederland.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

PART III

EDUCATIONAL PERSPECTIVES

Chapter 12

SCRIPTING ARGUMENTATIVE KNOWLEDGE CONSTRUCTION IN COMPUTER-SUPPORTED LEARNING ENVIRONMENTS

Armin Weinberger[1], Karsten Stegmann[2], Frank Fischer[1], and Heinz Mandl[3]
[1]Knowledge Media Research Center, Tübingen; [2]Eberhard-Karls-Universität, Tübingen; [3]Ludwig-Maximilians-Universität, Munich

Abstract: Computer-supported collaborative learning (CSCL) environments may encourage learners to engage in argumentative knowledge construction. Argumentative knowledge construction means that learners work together to elaborate on concepts by constructing arguments and counterarguments. This is achieved through discourse with the goal of acquiring knowledge within a specific domain. However, learners may encounter problems relating to one of three dimensions of argumentative knowledge construction. First, learners seem to have difficulties in constructing arguments that contribute to solving the task. Second, learners' arguments may lack important components such as data and warrants. Third, learners rarely build upon the arguments of their learning partners. Structuring argumentative knowledge construction with collaboration scripts is a promising instructional approach for facilitating specific process dimensions of argumentative knowledge construction. Little is known, however, about how to most effectively facilitate the acquisition of knowledge by directing collaboration scripts at specific dimensions of argumentative knowledge construction. This chapter will outline the theoretical background of argumentative knowledge construction and will then describe script components that target different dimensions of argumentative knowledge construction. The chapter will then discuss the empirical findings of two studies regarding the effects of these script components.

Collaborative learners are sometimes meant to construct and exchange arguments by collecting and balancing evidence and counterevidence through discourse with the goal of acquiring individual knowledge within a domain. Typically, however, learners do not work well on collaborative learning tasks in terms of constructing adequate arguments and interacting productively (Kuhn, Shaw, & Felton, 1997; Mandl, Gruber, & Renkl, 1996). Kuhn's work points out that even adult discussants rarely warrant or qualify their claims and thus rarely construct complete arguments. Furthermore, discussants are often unable to balance and integrate arguments and counterarguments critically. It has become clear that simply asking learners to collaborate is not sufficient for fostering argumentative knowledge construction (Mandl et al., 1996).

Asynchronous computer-supported collaborative learning has been regarded as a suitable context for facilitating argumentative knowledge construction (Andriessen, Baker, & Suthers, 2003; Marttunen & Laurinen, 2001). Learners communicate simultaneously or in an unspecified sequence via text-based interfaces and are thus able to type and read messages at their own individual pace. In this way, learners have more time than face-to-face learners to both compose their own messages and understand the messages of their learning partners. This time advantage may encourage learners with heterogeneous argumentation skills to take part in an argumentative debate. Besides the time advantage, learners who communicate asynchronously via computer can repeatedly access the arguments that have been already contributed and can easily revise the wording of their own arguments (see Pea, 1994). In text-based asynchronous communication, learners may compensate for individual deficits in learning prerequisites by investing more time in the reception and production of individual contributions. Learners may also take advantage of individual adaptive instructional support that is provided as part of the communication interface, such as computer-supported scripts. The question is, to what extent single script components could be directed at specific process dimensions of argumentative knowledge construction to improve individual knowledge acquisition. These could potentially be used to varying degrees, depending on how well the script components are able to compensate for the deficits of the student learning groups.

In this chapter, we first describe our approach to argumentative knowledge construction according to three process dimensions and their conceptual relationship to individual knowledge acquisition. Furthermore, we analyze how computer-supported script components may facilitate argumentative knowledge construction within these process dimensions. Finally, we summarize and discuss the results of two empirical studies that investigate the effects of computer-supported script components (with specific goal dimensions) on the processes and outcomes of argumentative knowledge construction. These studies have also been published in greater detail (see Weinberger, Ertl, Fischer, & Mandl, 2005 and Weinberger, Stegmann, & Fischer, 2005).

1. ARGUMENTATIVE KNOWLEDGE CONSTRUCTION

In argumentative knowledge construction, learners acquire knowledge through the elaboration of learning material by constructing arguments. Typically, argumentative knowledge construction scenarios are based on collaborative learning tasks (Leitão, 2000). For instance, learners may be provided with contrasting hypotheses about a problem and then argumentatively discuss, which hypothesis applies (Kollar, Fischer, & Slotta, 2005). Learners also engage in argumentative knowledge construction when working on open-ended and complex problem cases for which they have to create and balance multiple hypotheses (Means & Voss, 1996). Our approach to argumentative knowledge construction in collaborative, problem-based scenarios differentiates between three process dimensions, namely an *epistemic dimension* that describes arguments as steps towards solving the learning task, an *argument dimension* in which formal criteria for the composition of arguments are represented, and a *dimension of social modes of co-construction* that represents how learners interact with one another. This interaction is described in terms of how the learners relate their own arguments to the arguments of their learning partners (see Weinberger & Fischer, 2006).

1) The *epistemic dimension* refers to the question of how learners work on the tasks they are confronted with, for example, by constructing relationships between the conceptual space and the problem space (Fischer, Bruhn, Gräsel, & Mandl, 2002). Arguments may provide hypotheses on how to solve complex tasks. Learners relate the theoretical concepts of a given theory to the information in a problem. The epistemic dimension of arguments can indicate which concepts learners refer to and how learners connect concepts to solve a problem. This dimension can also show the extent to which learners are able to adequately apply knowledge. Collaborative learners do not always apply knowledge appropriately. When learners verbalize inappropriate applications of knowledge, there is a chance that their learning partners may adopt these misconceptions (e.g., Jeong & Chi, 1999). In this respect, the *epistemic dimension* of arguments is an important component of argumentative knowledge construction. Depending on how well learners are able to construct arguments relating to the epistemic dimension, they may acquire adequate knowledge or misconceptions (Weinberger, 2003).

2) The *argument dimension* comprises how learners construct arguments with regard to defining formal relationships between specific components of arguments, such as claims, data, and warrants. Toulmin (1958) has put forward a structural model of single arguments that is made up of several components. In this model, a single argument consists of a *claim*, which is a conclusion that is being presented and justified in the argument. The claim is

based on a *datum*, which is a fact that is supposed to support the claim. A *warrant* specifies the principle of how the datum supports the claim. Sometimes a warrant itself needs support, which is called *backing*. Thus, the backing indicates the principle upon which the warrant is based. Arguments may optionally also provide components that limit the validity of the claim and anticipate counterarguments. The *qualifier* indicates the extent to which the datum warrants the claim or may limit the validity of a claim. A *rebuttal* serves to anticipate parts of a counterargument that attack the data, the warrant, or the backing.

Toulmin's model poses an alternative to formal logic, which is closer to everyday reasoning in uncertain situations based on probabilities. However, Toulmin's model has been criticized for difficulties in distinguishing between the single components of the model in everyday argumentation, for example, distinguishing backing from data or differentiating between a qualifier and a rebuttal (Voss & van Dyke, 2001). We will therefore apply a condensed argument model using the components of *claim*, *datum with warrant* and *qualifier*.

How is formal argumentation structure related to individual knowledge construction? When learners construct arguments they elaborate and self-explain the learning material (Baker, 2003). These self-explanations help learners integrate new information into existing cognitive structures (Chi, Bassok, Lewis, Reimann, & Glaser, 1989). In terms of Toulmin's model (1958), self-explanation could be described as a process of composing an argument from several components. From this perspective, learners are supposed to seek data that supports or opposes a claim, make an inference through a warrant that indicates how the data supports a claim, and limit the validity of a claim by constructing qualifiers.

3) The *dimension of social modes of co-construction* indicates how learners interact with one another. In this dimension, a number of social modes of co-construction and their relationship to individual knowledge construction have been identified. These indicate the different degrees to which learners operate using the reasoning of their peers (Fischer et al., 2002; Teasley, 1997). For instance, when building consensus in a conflict-oriented manner, learners need to pinpoint specific aspects of their peers' contributions and either modify them or present alternatives. In these terms, learners need to build their reasoning more closely upon the reasoning of their peers when working to build consensus in a conflict-oriented manner. This is in contrast to quick consensus building, that is, when learners only appear to accept the contributions of their learning partners in order to continue with discourse (Weinberger, 2003). The extent to which learners operate on the reasoning of what has been said before in discourse has been termed *transactivity* of dis-

course, which is known to be positively related to individual knowledge acquisition (Teasley, 1997).

Table 12-1. Process dimensions of argumentative knowledge construction (see Weinberger & Fischer, in press)

Process dimension	answers the question
Epistemic dimension	How do learners' arguments contribute to solving the task?
Argument dimension	How do learners construct arguments formally?
Dimension of social modes of co-construction	To what extent do learners operate on the reasoning of their learning partners?

In summary, we propose to include three process dimensions for the analysis and facilitation of argumentative knowledge construction based on problem-oriented learning tasks (see Table 12-1). The processes of argumentative knowledge construction can be analyzed on a) an epistemic dimension (constructing arguments that contribute to solving a task), b) an argument dimension (building formally complete arguments), and c), a dimension of social modes of co-construction (operating on the reasoning of learning partners).

2. SCRIPT COMPONENTS FOR ARGUMENTATIVE KNOWLEDGE CONSTRUCTION

Argumentative knowledge construction is based on the assumption that learners need to construct arguments appropriately in order to benefit from collaborative learning environments. One approach for facilitating the outcomes of argumentative knowledge construction is to support learners in constructing arguments appropriately. Collaboration scripts provide an instructional approach that aims to facilitate the processes of argumentative knowledge construction.

Collaboration scripts were initially developed to encourage college students working in dyads to acquire knowledge from texts on the natural sciences (O'Donnell & Dansereau, 1992). Collaboration scripts provide more or less explicit and detailed instructions for small groups of learners on *what* activities need to be executed, *when* they need to be executed, and by *whom* they need to be executed in order to foster individual knowledge acquisition.

Prototypical scripts are instructed prior to collaborative learning. Participants are trained to engage in the scripted collaborative activities, which are in turn supposed to facilitate the individual acquisition of knowledge. In computer-supported collaborative learning, there is typically no opportunity for antecedent collaboration training. Thus, computer-supported collaborative learning (CSCL) is often facilitated by the design of the interface (Baker & Lund, 1997; Scardamalia & Bereiter, 1996). Learners communicating via these interfaces are, to varying degrees, implicitly guided to engage in activities, as the interface suggests or limits specific discourse activities (Runde, Jucks, & Bromme, this volume). Computer-supported scripts similarly aim to directly influence the interaction patterns of collaborative learners rather than train learners prior to actual collaboration.

When analyzing collaboration scripts in the context of argumentative knowledge construction, it can be noted that scripts typically aim to facilitate different process dimensions simultaneously. For instance, prototypical scripts may support epistemic activities, for example, summarizing a paragraph, as well as specific social modes, for example, criticizing the contributions of the learning partner (O'Donnell & Dansereau, 1992). Little is known about the effects of single script components that target specific process dimensions on knowledge construction. In research on collaboration scripts, Larson and colleagues (1985) compared the effects of script components with different, specific goal dimensions, namely an elaborative and a metacognitive script component. The elaborative script component supported elaborative activities by modeling the role of a *recaller*, who was given the task of personalizing information or of using imagery to help remember the learning material. The metacognitive script component modeled the role of a *listener* who was given metacognitive tasks, such as error detection. The metacognitive script component impeded individual knowledge construction, whereas the elaborative script component facilitated individual knowledge construction. This study thus indicates that differentiated effects can be expected from script components with specific goal dimensions. Larson and colleagues (1985) argued, for instance, that some script components may also distract learners from learning goals. It is important to expand research on script components with single goal dimensions in order to better understand why and what kind of script components facilitate argumentative knowledge construction. The research should also analyze possible side effects that single script components may have on the processes and outcomes of argumentative knowledge construction. We may thus accumulate knowledge on how multiple processes of argumentative knowledge construction interact and affect individual knowledge acquisition.

With reference to the framework of argumentative knowledge construction, we differentiate between epistemic script components, argumentative script components, and social script components (Weinberger & Fischer, 2006).

Epistemic script components aim to structure the discourse activities of collaborative learners with respect to the content of the discussion and with regard to the steps towards solving the task. Epistemic script components may support learners in finding adequate task strategies and may ask learners to elaborate on aspects of the task they would not normally consider (cf. Reiser, 2002). Approaches that we classify as epistemic script components may thus map expert-like strategies onto the interaction of learners (Dufresne, Gerace, Thibodeau Hardiman, & Mestre, 1992; Herrenkohl and Guerra, 1998). For instance, these components provided collaborative learners with task strategies such as predicting and theorizing, summarizing results, and relating predictions and theories to results. As these qualitative studies indicate, epistemic script components may need to be reinforced by social script components.

Argumentative script components aim to support the construction of arguments in terms of warranting and qualifying claims based on argument models such as Toulmin's (1958). Argumentative script components aim to help learners construct formally adequate arguments and thus better elaborate the learning material (Andriessen et al., 2003). As learners supplied with argumentative script components are supposed to formulate better arguments in discourse, learners may also acquire knowledge on how to argue within a scientific domain.

Social script components specify and sequence the interaction of learners in order to promote knowledge construction (King, 1999). Social script components may thus support learners to engage in adequate interaction strategies that they would not apply spontaneously. For instance, social script components may facilitate transactivity by asking learners to respond critically to the contributions of their learning partners. Social script components typically also have learners rotate to work on different activities (e.g., Herrenkohl & Guerra, 1998). The reciprocal teaching approach (Brown & Palincsar, 1989), for instance, assigns the roles of a teacher and a learner for various text comprehension tasks.

In summary, prior research on different script components found that not all components are equally effective for promoting knowledge construction. Some script components appear to distract learners from the actual task or replace central learning activities rather than support learners in engaging in the activities themselves. To date, there has been little systematic research on the effects that various script components have on argumentative knowledge construction. Furthermore, most research on scripts deals with trained face-to-face learning environments rather than computer-supported scripts. Therefore, there is little knowledge about how specific computer-supported script components with different goal dimensions can facilitate the processes and outcomes of argumentative knowledge construction.

3. GOALS OF THE EMPIRICAL STUDIES

The goal of the empirical studies is to investigate the effects of computer-supported epistemic, argumentative and social script components on argumentative knowledge construction. These single script components focus on different process dimensions of argumentative knowledge construction and may have differentiated effects on its outcomes. We conducted two studies in order to investigate how individual computer-supported script components can facilitate argumentative knowledge construction (for further details see Weinberger, Ertl, et al., 2005; Weinberger, Stegmann, et al., 2005).

4. METHOD

The results are based on a one factorial analysis and single group comparisons of the experimental groups with epistemic, social, and argumentative script components with the control group. Each experimental group in the two studies consisted of 24 students. Thus, 96 participants in 32 groups of three entered the statistical analyses. The methods that were used in each of the studies were identical.

4.1 Sample and setting

First-semester educational science students from the University of Munich participated in the studies. The students took an obligatory introduction course to educational science. One of the regular face-to-face sessions of the course was transformed into an online learning session. Participation in this session was required in order to receive a course credit at the end of the semester. The learning outcomes of the experimental session, however, did not count towards the students' overall performance. The introduction course sessions normally consist of a one hour lecture and a successive two hour seminar. Similarly, the collaborative online learning session took three hours. The students were randomly assigned to the experimental conditions in groups of three. Participants in each group of three were separated from each other in different laboratory rooms and communicated asynchronously with the help of web-based discussion boards in a computer-supported learning environment.

4.2 Learning task

The task of the participants was to apply the attribution theory of Weiner (1985) to three problem cases (see Table 12-2 for an example of a problem case) and reach agreement on a final analysis for each case.

Table 12-2. One of the three problem cases, namely the "math case", learners needed to analyze and discuss

As a student teacher in a high school, you participate in a school counseling session with Michael Peters, a pupil in the 10th grade.
"Recently I've started to realize that math is just not my thing. Last year I almost failed math. Ms. Weber, my math teacher, told me that I would really have to make an effort if I wanted to pass 10th grade. Actually, my parents stayed pretty calm when I told them this. First mom said that nobody in our family is a math whiz. My father just kept smiling and told the story about how he cheated on his final math exams by copying from other students and using cheat sheets. 'The Peters family,' he said, 'has always been a math teacher's nightmare'. Once when I was slightly tipsy at a school party, I told this story to Ms. Weber. She said that it was not a bad excuse, but not a good one either. She said it was just one of a number of excuses you could come up with to justify being lazy. Last year I barely made it through mathematics, so I am really nervous about the upcoming school year!"

The descriptions of the problem cases were embedded into the web-based learning environment, so that the participants could study the problem case while composing new messages on the web-based discussion boards.

4.3 Computer-supported learning environment

All groups collaborated in three web-based discussion boards – one for each case. The web-based discussion boards provided a main page with an overview of all message headers. In this overview, answers to original messages appeared in outline form. The learners could read the full text of all messages, reply to the messages, or compose and post new messages. In the replies, the original messages were quoted out with > as in standard newsreaders and e-mail programs.

4.4 Procedure

Prior to collaboration, the randomization of participants was successfully controlled using individual questionnaires and tests, for example, on prior knowledge, ambiguity tolerance, and computer experience. Subsequently, learners were able to study a three page summary of the attribution theory for 15 minutes. Learners were allowed to make notes and keep the text and their notes during the collaborative phase. The collaboration time, in which learners communicated with each other via asynchronous, text- and web-based discussion boards, was 80 minutes in all experimental groups. All discourse activities were recorded within the web-based discussion boards to collect data on the dimensions of argumentative knowledge construction. The experimental conditions differed only with respect to the computer-supported script components that were implemented using the interface of the computer-supported environment. After collaboration, learners were tested for individual domain-specific knowledge using individual post-tests similar to the pre-tests on prior knowledge.

4.5 Instruments

In order to analyze the extent to which the script components influenced the processes of argumentative knowledge construction, we segmented and analyzed each of the single arguments the learners put forward in the written discourse on the epistemic dimension, the argument dimension, and the dimension of social modes of co-construction. On the epistemic dimension we differentiated, for instance, between adequate or inadequate arguments in terms of the relationships the learners constructed between concepts and case information. With respect to the argument dimension, we coded the completeness of arguments according to a model of arguments consisting of claim, datum with warrant, and qualifier. We also coded the arguments with regard to their social mode. The interrater reliability was sufficiently high (for a detailed description of the process analyses see Weinberger & Fischer, 2006).

Pre- and post-tests consisted of problem cases comparable to the three cases learners were asked to analyze during the collaborative phase. The case analyses the learners needed to produce in the pre- and post-tests were segmented into units consisting of a theoretical concept applied to problem case information. In a manner similar to the process analysis, these units were coded with respect to their adequacy in terms of the relationships learners constructed between theoretical concepts and case information to indicate domain-specific knowledge. The adequacy of the individual learner's case analyses in the pre- and post-test was determined by their fit to expert solutions of the problem cases. These expert solutions particularly stressed the application of multiple perspectives to the cases.

More detailed information on different aspects of the quantitative analyses of the individual empirical studies has been provided in various publications (e.g., Weinberger, 2003; Weinberger, Ertl, et al., 2005; Weinberger & Fischer, 2006).

4.6 Treatments

The following experimental groups were examined in Study 1:

The *control groups* accessed the three distinct web-based discussion boards of the CSCL environment to read or contribute messages. When composing a new message, learners were free to choose to start a new discussion thread or to reply and contribute to an existing discussion thread.

The *epistemic script component group* could access and contribute to the web-based discussion boards in a manner identical to the control group, but whenever a new discussion thread was started, the text window was structured with prompts of the epistemic script component. These prompts asked learners to separate relevant from irrelevant case information, structured how learners applied theoretical concepts to the problem cases, and asked learners to suggest future developments of the case and pedagogical interventions (see Figure 12-1).

Case information, which can be explained with the attribution theory

Relevant terms of the attribution theory for this case:

- Does a success or a failure precede this attribution?
- Is the attribution located internally or externally?
- Is the cause for the attribution stable or variable?
- Does the concerned person attribute himself/herself or does another person attribute him/her?

Prognosis and consequences from the perspective of the attribution theory:

Case information which cannot be explained with the attribution theory:

Figure 12-1. Prompts of the epistemic script component to apply the concepts of Weiner's (1985) attribution theory to problem cases.

The *groups with the argumentative script component* were provided with three text windows named claim, datum with warrant and qualifier. Learners were supposed to collect at least one datum for their claim, explicit the warrant for how the datum supports the claim, and provide a qualifier for their claim by filling out all three text windows (see Figure 12-2). Subsequently, learners could click an "Add"-Button which displayed the three argument components in the actual text window of the web-based discussion board. Learners could add any number of arguments in the main text window.

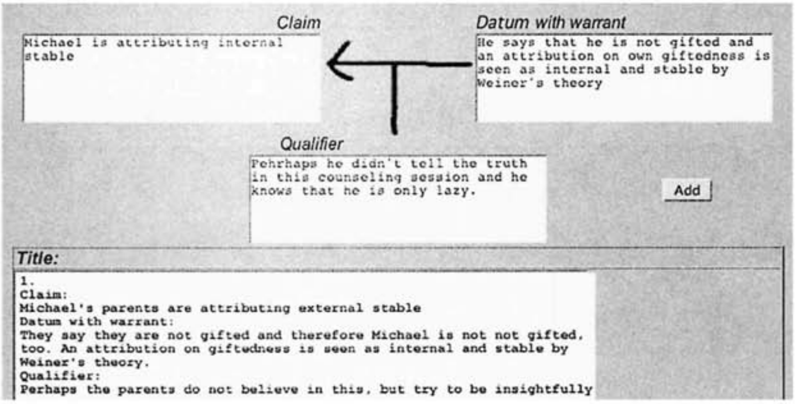


Figure 12-2. User interface realizing the argumentative script component with four text windows: claim, datum with warrant, qualifier, and message body.

The learners in the *social script component groups* were assigned two roles – a) analyst for one of the cases with the task of composing initial and concluding analyses of the case and responding to critiques, and b) constructive critic for the other two cases with the task of repeatedly criticizing the case analyses. The number of messages was determined by the social script component (one initial case analysis, two critiques, two replies to the critiques, two more critiques and one concluding analysis). These messages given by the social roles were automatically sequenced, that is, learners were led through each discussion board to submit eight messages in total (see Figure 12-3). Furthermore, the single messages were supported by prompts within the text windows such as "These aspects are not clear to me yet", "We have not reached consensus concerning these aspects" or "My proposal for an adjustment of the analysis is".

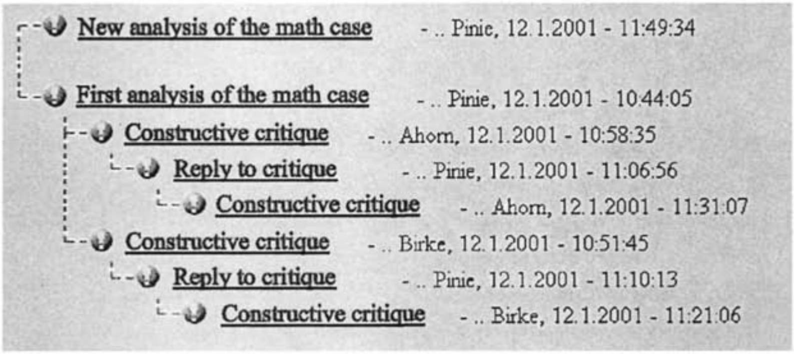


Figure 12-3. Structure of a discourse supported with the social script component with an initial analysis of the math case, two constructive critiques, two replies of the case analyst to the critiques, two more critiques and a new analysis of the case by the case analyst.

5. RESULTS

The *epistemic script component* reduced the amount of off-topic discourse and focused learners' discourse on just a few new and adequate knowledge concepts. The learners who were supported by the epistemic script component contributed more to solving the problem cases than learners without the script component (see Table 12-3; see also Mäkitalo, Weinberger, Häkkinen, Järvelä, & Fischer, 2005; Weinberger, 2003; Weinberger, Ertl, et al., 2005).

Table 12-3. Example of a learner's message with a case analysis supported with the epistemic script component.

Does success or failure precede this attribution?
- failures
Is the attribution located internally or externally?
- internal
Is the cause for the attribution stable or variable?
- Michael and his parents: stable causes (talent)
- Teacher: variable causes (effort)
Who is attributing? Self or other?
- Michael
- His parents
- The teacher

The epistemic script component, however, also affected the argument dimension and the dimension of social modes of co-construction. Learners with the epistemic script component constructed less formally complete and less transactive arguments than learners without the epistemic script component.

The *argumentative script component* reduced off-topic discourse and facilitated the formally adequate construction of single arguments. Learners with the argumentative script component warranted and qualified their claims substantially more frequently than learners without this script component (see Table 12-4).

Table 12-4. Example of a learner's message with a case analysis supported with the argumentative script component.

Claim:
in this case, it has to be an internal stable attribution
Datum with warrant:
parents say that 1.) the whole family was not "witty" in math and the father adds that he barely passed his math exam.
internal because talent is to be located within the person and stable because talent does not change.
Qualifier:
i can't think of any

However, this script component also impeded the content quality of the single arguments and reduced the adequate application of new knowledge concepts that were to be learned (see also Weinberger, Stegmann, et al., 2005).

The *social script component* reduced off-topic discourse and facilitated the dimension of social modes of co-construction of argumentative knowledge construction. The discourse of groups with the social script component was more critical and transactive than the discourse of groups of learners without social script component (see Table 12-5).

Learners supported with this script component operated more on the reasoning of their learning partners. Additionally, the social script component seemed to foster epistemic activities. Learners supported with this script component engaged more frequently in epistemic activities to solve the problem case than learners without this script component (see also Weinberger, 2003; Weinberger, Ertl, et al., 2005 for more detailed process analyses).

Table 12-5. Example of a learner's critical reply supported by the social script component.

These aspects are not clear to me yet:
What attribution according to attribution theory can be applied? If mother is not talented - so is the son?
We have not yet reached consensus concerning these aspects:
The teacher does mention that it is only his laziness; she doesn't explain it to him, however.
My proposals for an adjustment of your analysis:
Proposal for a solution: Parents should attend a re-attribution training!

Regarding individual knowledge acquisition, large improvements were observed between pre- and post-test for learners in all experimental groups, including the control group. The results further support the notion that the acquisition of individual domain-specific knowledge can be influenced by specific script components implemented within CSCL environments. The epistemic script component impeded the individual acquisition of knowledge compared to the control group. The argumentative script component did not facilitate knowledge acquisition beyond the levels of the control group. The social script component, however, proved to support the individual acquisition of domain-specific knowledge. After the collaborative learning phase, learners provided with the social script component were better able to individually apply different concepts to problem cases than learners without the social script component (see Weinberger, Ertl, et al., 2005; Weinberger, Stegmann, et al., 2005).

6. CONCLUSIONS

The studies investigated the effects of different script components on argumentative knowledge construction in computer-supported learning environments. The learning environments investigated differed only with respect to the script components, namely (1) the epistemic script component, which structured how learners handled the task and which concepts they used, (2) the argumentative script component that asked learners to warrant and qualify their claims, and (3) the social script component that aimed to facilitate how learners interacted with each other. All computer-supported script components substantially reduced off-topic discourse and facilitated the specific processes of argumentative knowledge construction that they were focusing on. Based on these findings, all script components seem to have the general effect of focusing learners on the task. Script components guide and inform learners of what to do next to solve the task in one way or another. Therefore, learners seem to have less opportunity to engage in off-topic discourse. Apart from this general effect, script components can be very specific. Script components with single goal dimensions can be implemented deliberately into CSCL environments to address specific shortcomings in the interaction of groups of learners rather than providing a "one-script-fits-all" model.

The results indicate that *epistemic script components* help learners to construct arguments that contribute to solving problem cases, but that learners do not necessarily benefit from this support with regard to individual knowledge acquisition. One explanation for this could be that epistemic script components might not sufficiently support joint elaboration of the learning material, but rather function as checklists. Thus, epistemic script components may enable learners to solve the tasks with a limited elaboration of the learning material. In order to avoid this elaboration-reduction-effect, epistemic script components may need to be faded out. It may also be necessary in some cases to make collaborative learning tasks harder instead of simplifying the collaborative learning task in order to facilitate the active elaboration of the learning material (Palincsar & Herrenkohl, 1999; Reiser, 2002). Furthermore, the degree to which epistemic script components demand the elaboration of learning material or "micromanage" the task may depend on the prior knowledge of the participants. It may prove unnecessary to provide epistemic script components to learners with above-average prior knowledge and skills. Advanced learners may already possess functional strategies for solving a task and additional epistemic scripting might simply distract learners from the actual task (see Larson et al., 1985). In order to avoid this over-scripting effect (Dillenbourg, 2002), epistemic script components may need to be carefully matched with the individual prior knowledge of the participants. Too much or too detailed epistemic scripting may impede the elaboration of the learning material and the interaction of learners; particularly when the script oversimplifies the task and divides it into subtasks that can be worked on by each learner individually (Cohen, 1994).

The *argumentative script component*, like the other script components, facilitated the process dimension that it targeted. This study showed that argumentative script components are able to support argumentative knowledge construction in both the formal argumentation process dimension during discourse and individual knowledge acquisition. Scripting the construction of arguments may support learners in elaborating the learning material. By constructing formally complete arguments with claim, datum, warrant, and qualifier, learners need to self-explain the learning material, which may facilitate the acquisition of knowledge (Baker, 2003). Learners supported with the argumentative script component may have elaborated the learning material better than learners without the script. Learners did not always use the appropriate concepts to solve the task, however, and may thus have elaborated prior knowledge using misconceptions rather than the knowledge that was to be learned. Argumentative script components in this way may function as a thinking tool to amplify elaboration, but fail to prompt learners to use the relevant knowledge concepts that are to be learned.

The results further indicate that *social script components* may facilitate social modes of co-construction, epistemic activities, and the individual acquisition of knowledge. Collaborative learners without support from a social script component often build a minimal consensus in order to hastily complete collaborative tasks or do not collaborate on the learning task at all. In contrast, social script components support learners in inquiring about the contributions of the learning partners more critically and thereby help them acquire more knowledge individually than learners without additional support in the dimension of social modes of co-construction (see King, 1999; Palincsar & Herrenkohl, 1999). This critical approach to the contributions of the learning partners has also appeared to facilitate the elaboration of the learning material. One explanation could be that socially scripted learners engaged in more transactive discussions and appeared to benefit to a greater extent from the contributions of their learning partners (Teasley, 1997). Another explanation is that learners with the social script component elaborated the learning material to a greater extent, because they anticipated critique from their learning partners. This explanation is in line with studies that indicate that only the expectation of externalizing knowledge facilitates learning (Renkl, 1997).

In summary, computer-supported script components can be designed to facilitate specific process dimensions of argumentative knowledge construction. Script components that "micromanage" discourse on an epistemic dimension may cause learners to focus on solving the task at hand without elaborating the learning material. In order to foster the elaboration of the learning material and individual knowledge acquisition, script components may need to target not only the epistemic activities, but also focus on social modes of co-construction in argumentative discourse (see Herrenkohl & Guerra, 1998; Palincsar & Herrenkohl, 1999). Conversely, script components that are aimed at formal aspects of argument construction without additionally fostering epistemic activities or social modes of co-construction may not be able to help learners achieve better results than without support from a script. Content-independent argumentative script components may aid elaboration, but hold the danger that learners may not be able to select the appropriate concepts that are supposed to be elaborated. The social script component of this study, in contrast, managed to not only facilitate transactive discourse, but also supported the epistemic activities of learners. This indicates that transactivity can be essential to argumentative knowledge construction and can be facilitated beyond the levels that collaborative learners would spontaneously achieve (Teasley, 1997).

7. FUTURE RESEARCH

CSCL environments offer a suitable context for scripting the interaction of learners. Clearly, there is further need to examine beneficial interactions of script components for CSCL, for example, investigating epistemic script components which do not micromanage interaction of learners in combination with social and argumentative script components (see Ertl, Kopp, & Mandl, this volume). Individual cognitive processes and their relationship to various process dimensions and outcomes of argumentative knowledge construction may explain to a greater extent how learners benefit from argumentative knowledge construction scenarios. Text-based CSCL may provide a unique opportunity for investigating the cognitive processes of learners. While engaging in written communication, learners may simultaneously provide information about their cognitive activities through think-aloud-techniques. We also need to better understand how scripts interact with learners' prior knowledge and skills, which may be represented as *internal scripts* in contrast to external, computer-supported scripts (Carmien, Kollar, G. Fischer, & F. Fischer, this volume). Therefore, an important question for future research of CSCL environments is how scripts can be designed not to substitute, but to facilitate discourse and cognitive activities related to individual acquisition of knowledge. In these terms, we need to further investigate the interaction of different script components that may be adapted to the already existing internal scripts. Design environments need to be developed in order to improve the impact of computer-supported script research in practice at schools and universities. These environments should facilitate the adaptive combination of script components with different representations that can be used relatively independent of the computer support available in classrooms.

ACKNOWLEDGEMENTS

The studies have been funded by the Deutsche Forschungsgemeinschaft (DFG) and the Kaleidoscope Network of Excellence within a European research team entitled "Computer-Supported Scripting of Interaction in Collaborative Learning Environments".

REFERENCES

Andriessen, J. E. B., Baker, M., & Suthers, D. (Eds.). (2003). *Arguing to learn. Confronting cognitions in computer-supported collaborative learning environments.* Dordrecht: Kluwer.

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

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

Brown, A. L., & Palincsar, A. S. (1989). Guided, cooperative learning and individual knowledge construction. In L. B. Resnick (Ed.), *Knowing, learning, and instruction. Essays in the honour of Robert Glaser* (pp. 393-451). Hillsdale, NJ: Lawrence Erlbaum Associates.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*, 145-181.

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

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL: Can we support CSCL?* (pp. 61-91). Heerlen: Open Universiteit Nederland.

Dufresne, R. J., Gerace, W. J., Thibodeau Hardiman, P., & Mestre, J. P. (1992). Constraining novices to perform expertlike problem analyses: Effects on schema acquisition. *The Journal of the Learning Sciences, 2*(3), 307-331.

Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. (2002). Fostering collaborative knowledge construction with visualization tools. *Learning and Instruction, 12*, 213-232.

Herrenkohl, L. R., & Guerra, M. R. (1998). Participant structures, scientific discourse, and student engagement in fourth grade. *Cognition and Instruction, 16*, 433-475.

Jeong, H., & Chi, M. T. H. (1999). *Constructing shared knowledge during collaboration and learning.* Paper presented at the AERA Annual Meeting, Montreal, Canada.

King, A. (1999). Discourse patterns for mediating peer learning. In A. M. O'Donnell & A. King (Eds.), *Cognitive perspectives on peer learning* (pp. 87-115). Mahwah, NJ: Lawrence Erlbaum Associates

Kollar, I., Fischer, F., & Slotta, J. D. (2005). Internal and external collaboration scripts in web-based science learning at schools. In T. Koschman, D. Suthers, & T.W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years!* (pp. 331-340). Mahwah, NJ: Lawrence Erlbaum Associates.

Kuhn, D., Shaw, V., & Felton, M. (1997). Effects of dyadic interaction on argumentative reasoning. *Cognition and Instruction, 15*(3), 287-315.

Larson, C. O., Dansereau, D. F., O'Donnell, A. M., Hytecker, V. I., Lambiotte, J. G., & Rocklin, T. R. (1985). Effects of metacognitive and elaborative activity on cooperative learning and transfer. *Contemporary Educational Psychology, 10*, 342-348.

Leitão, S. (2000). The potential of argument in knowledge building. *Human Development, 43*, 332-360.

Mäkitalo, K., Weinberger, A., Häkkinen, P., Järvelä, S., & Fischer, F. (2005). Epistemic cooperation scripts in online learning environments: Fostering learning by reducing uncertainty in discourse? *Computers in Human Behavior, 21*(4), 603-622.

Mandl, H., Gruber, H., & Renkl, A. (1996). Communities of practice toward expertise: Social foundation of university instruction. In P. B. Baltes & U. Staudinger (Eds.), *Interactive* *minds. Life-span perspectives on the social foundation of cognition* (pp. 394-411). Cambridge, MA: Cambridge University Press.

Marttunen, M., & Laurinen, L. (2001). Learning of argumentation skills in networked and face-to-face environments. *Instructional Science, 29*, 127-153.

Means, M. L., & Voss, J. F. (1996). Who reasons well? Two studies of informal reasoning among children of different grade, ability and knowledge levels. *Cognition and Instruction, 14*, 139-178.

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

Palincsar, A. S., & Herrenkohl, L. R. (1999). Designing collaborative contexts: Lessons from three research programs. In A. M. O'Donnell & A. King (Eds.), *Cognitive perspectives on peer learning* (pp. 151-177). Mahwah, NJ: Lawrence Erlbaum Associates.

Pea, R. D. (1994). Seeing what we build together: Distributed multimedia learning environments for transformative communications. Special Issue: Computer support for collaborative learning. *Journal of the Learning Sciences, 3*(3), 285-299.

Reiser, B. J. (2002). Why scaffolding should sometimes make tasks more difficult for learners. In G. Stahl (Ed.), *Proceedings of the Computer Support for Collaborative Learning: Foundations for a CSCL Community - CSCL 2002* (pp. 255-264). Boulder, CO: Lawrence Erlbaum Associates.

Renkl, A. (1997). *Lernen durch Lehren - Zentrale Wirkmechanismen beim kooperativen Lernen [Learning through teaching - central mechanisms in cooperative learning]*. Wiesbaden: Deutscher Universitäts-Verlag.

Scardamalia, M., & Bereiter, C. (1996). Computer support for knowledge-building communities. In T. Koschmann (Ed.), *CSCL: Theory and practice of an emerging paradigm* (pp. 249-268). Mahwah, NJ: Lawrence Erlbaum Associates.

Teasley, S. (1997). Talking about reasoning: How important is the peer in peer collaboration? In L. B. Resnick, R. Säljö, C. Pontecorvo, & B. Burge (Eds.), *Discourse, tools and reasoning: Essays on situated cognition* (pp. 361-384). Berlin: Springer.

Toulmin, S. (1958). *The uses of argument*. Cambridge, MA: Cambridge University Press.

Voss, J. F., & van Dyke, J. A. (2001). Argumentation in psychology: Background comments. *Discourse Processes, 32*(2&3), 89-111.

Weinberger, A. (2003). *Scripts for computer-supported collaborative learning. Effects of social and epistemic collaboration scripts on collaborative knowledge construction.* Doctoral thesis. Ludwig-Maximilian University, Munich. Available at: http://edoc.ub.uni-muenchen.de/archive/00001120/01/Weinberger_Armin.pdf.

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., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education, 46*, 71-95.

Weinberger, A., Reiserer, M., Ertl, B., Fischer, F., & Mandl, H. (2005). Facilitating collaborative knowledge construction in computer-mediated learning with cooperation scripts. In R. Bromme, F. Hesse, & H. Spada (Eds.), *Barriers and biases in computer-mediated knowledge communication - and how they may be overcome* (pp. 15-37). Boston: Kluwer.

Weinberger, A., Stegmann, K., & Fischer, F. (2005). Computer-supported collaborative learning in higher education: Scripts for argumentative knowledge construction in distributed groups. In T. Koschmann, D. Suthers, & T. W. Chan (Eds.), *Computer Supported*

Collaborative Learning 2005: The Next 10 Years! (pp. 717-726). Mahwah, NJ: Lawrence Erlbaum Associates.

Weiner, B. (1985). An attributional theory of achievement motivation and emotion. *Psychological Review, 92*, 548-573.

Chapter 13

SUPPORTING COLLABORATIVE LEARNING IN VIDEOCONFERENCING USING COLLABORATION SCRIPTS AND CONTENT SCHEMES

Bernhard Ertl[1], Birgitta Kopp[2], and Heinz Mandl[2]
[1]Bundeswehr Universität, Munich; [2]Ludwig-Maximilians-Universität, Munich

Abstract: Studies have shown that videoconferences are an effective medium for facilitating communication between parties who are separated by distance. Furthermore, studies reveal that videoconferences are effective when used for distance learning, particularly due to their ability to facilitate complex collaborative learning tasks. However, as in face-to-face communication, learners benefit further when they receive additional support for such learning tasks. This chapter provides an overview of two empirical studies that offer general insights regarding some effective and less effective ways to support collaborative learning with videoconferencing. The focus is on collaboration scripts that serve to provide task-specific support and content schemes that serve to provide content-specific support. Based on the results of the two studies, conclusions can be drawn about the support measures for promoting learning. Conclusions can also be reached about the need for employing both content schemes and collaboration scripts to provide learners with the most benefit.

Videoconferences are regarded as highly beneficial mechanisms for facilitating collaborative distance learning. In contrast to text-based communication, videoconferencing enables learners to interact more frequently and thus supports learners in solving complex tasks (see Anderson et al., 1997; Pächter, 2003). Furthermore, the use of shared applications allows learners to work collaboratively on a written solution to a problem *while* discussing important aspects of that solution.

However, as collaborative problem solving is a complex task in itself, learners need support when performing such tasks. When employing support measures that are widely used in face-to-face and text-based learning scenarios, the following question arises: To what degree are these measures applicable for learning in videoconferencing? For example, trainings (see O'Donnell & Dansereau, 2000; Rummel & Spada, this volume) that are widely used in face-to-face situations may be difficult to realize when learners are separated by distance. Furthermore, cues, such as sentence openers or prompts that are often used in text-based learning environments (see Weinberger, Fischer, & Mandl, this volume), may prove ineffective in spoken communication as they may be ignored in the natural flow of spoken communication.

Thus, it seems necessary to develop new methods of support for collaborative learning in videoconferencing. A key support feature may be the shared application, which is central to computer-mediated communication (see Baker & Lund, 1997; Dillenbourg & Traum, 1999). Videoconference participants can access shared applications on their computer screens and can easily manipulate the contents of these shared applications. Furthermore, the shared application may be pre-structured to provide instructional support and thereby function as a representational context for the learners (see Baker & Lund, 1997; Fischer, Bruhn, Gräsel, & Mandl, 2000; Suthers & Hundhausen, 2001). This context may change the learners' perception of the task and thus guide them to a better solution.

This chapter discusses ways to support collaborative learning in videoconferencing and compares two different support measures. The first support measure is a collaboration script that pre-structures the learning task. This method is widely used in scripted collaboration research. The second support measure is a content scheme that focuses learners' attention on important aspects of the subject matter and is realized through a pre-structured shared application. Through two empirical studies, the effects of both support measures are compared with respect to collaborative learning outcomes and individual learning outcomes.

1. COLLABORATIVE LEARNING

Collaborative learning in small groups means that groups act relatively independent of a teacher with the goal of acquiring knowledge or skills (see

Cohen, 1994; Dillenbourg, 1999). One major goal of collaborative learning is to support social interaction and encourage the learner's cognitive processes. In this context, learners' elaborations are seen to play a crucial role (see Webb, 1989; Webb & Palincsar, 1996) for expressing their knowledge, ideas and beliefs to their partners (see O'Donnell & King, 1999; Palincsar & Brown, 1984; Rosenshine & Meister, 1994). There are three specific mechanisms in collaborative learning that should be emphasized: the tendency for cognitive conflicts to arise (Doise & Mugny, 1984; Nastasi & Clements, 1992; Piaget, 1932), the need for elaborated explanations and negotiations (Webb, 1989; Webb & Palincsar, 1996) and the co-construction of knowledge (Bruhn, 2000; Fischer et al., 2000; Roschelle & Teasley, 1995). These socially mediated learning processes should foster individual cognitive engagement with the learning material and consequently benefit learning outcomes.

In the context of collaborative learning, it is also necessary to consider the conceptualization of learning outcomes. There are two main ways to assess the benefits of a collaborative learning scenario: either collaboratively on a group level or individually on the learner level.

The *collaborative learning outcome* is the success all learning partners achieve together. Due to the interdependent nature of the group task that requires the various contributions of every group member to solve it, learning success can be measured through the quality of the group product. This can be recorded through a case solution that learners develop during collaboration (see Hertz-Lazarowitz, Kirkus, & Miller, 1992) or through a test the learners complete collaboratively after collaboration (Salomon, 1998).

The *individual learning outcome* is based on the knowledge or skills which the individual learns through interaction with others. The main objective is to discover how much of the knowledge that is co-constructed in the collaboration can be transferred to the individual situation. There are many different potential learning measurements: tests (Jeong & Chi, 1999; Lambiotte et al., 1988) to measure factual knowledge acquisition as well as case solutions to measure applied knowledge acquisition (Bruhn, 2000).

However, there are differences in the interpretation of such learning outcomes (see Anderson, Reder, & Simon, 1996; Greeno, 1997; Hertz-Lazarowitz et al., 1992; Salomon & Perkins, 1998; Slavin, 1995; Webb, 1989). These involve the degree to which individual knowledge assessment is able to evaluate effects of collaborative knowledge construction or the degree to which a group assessment can provide indications about an individual's learning progress.

Characteristics of collaborative learning in videoconferencing. In videoconferencing, collaboration processes may change depending on the various media being used. Such changes can also affect learning outcomes. During videoconferencing, the synchronous interaction of the learners can be guided by the transmission of audio, video, and data. Studies on the usage of videoconference systems for small groups highlight the importance of the quality of the audio transmission (O'Connaill, Whittaker, & Wilbur, 1993): The collaboration scenario will only be successful if the audio transmission is reliable, specifically if sound bytes are not lost and audio delays are not more than 500ms (see Finn, Sellen, & Wilbur, 1997; O'Connaill et al., 1993). Another important component is video. A connection via video can modify the perception of the learning partners. During videoconferencing, some communication cues such as facial expressions and gestures may not fully be transmitted (see Bruce, 1996). It is also not possible for participants to make eye contact. Eye contact is particularly important for controlling the communication in groups (see Anderson et al., 1997; Isaacs & Tang, 1997; Joiner, O'Shea, Smith, & Blake, 2002). In these ways, communication in videoconferencing scenarios can differ from face-to-face settings. In spite of these differences in learning discourse, results from research conducted to date suggest that learning outcomes are not affected by videoconferencing (see Anderson et al., 1997; Bruhn, 2000; Fischer et al., 2000; Pächter, 2003).

Moreover, videoconferencing offers new methods for supporting collaboration and learning processes. One key feature of videoconferencing systems that provides this assistance is application sharing when it is used to transmit data. Using application sharing, learning partners in videoconference settings are able to access and modify the same content on their individual screens (Dillenbourg & Traum, 1999). When sharing an application, learners in different locations have the ability to work on the same document simultaneously and to collaborate to find a written problem solution. Learners are able to disseminate their knowledge with the help of the shared application. In short, shared applications can support the interaction and the exchange of knowledge through discourse. Furthermore, the interaction between the learners is simplified because they can refer to the shared information without having to provide further explanations to their collaboration partners (Roschelle & Teasley, 1995). Another important aspect of shared applications is the *salience* of their contents (Suthers, 2001). This salience of this permanent joint knowledge representation influences the co-construction of knowledge (Dillenbourg & Traum, 1999) and allows for modifications and improvements. Furthermore, the concept of salience can be useful and important for specifically supporting collaborative learning in videoconferencing. Important aspects of the collaborative task can be made salient and therefore foster collaborative learning through videoconferencing.

2. FOSTERING COLLABORATIVE LEARNING IN VIDEOCONFERENCING

As collaborative learning can be deficient in certain areas, e.g., due to differences in expertise (Slavin, 1995), differences in status (Cohen & Lotan, 1995) or dysfunctional group phenomena (see Salomon & Globerson, 1989), collaborative learning may impede the results of collaborative and individual learning outcomes. Therefore, it is necessary to provide support for the learners. Such support strategies focus mainly on improving collaborative learning processes, either by offering collaboration strategies in the form of collaboration scripts or by presenting content processing strategies in the form of content schemes. Both approaches will be characterized in the following sections.

2.1 Collaboration scripts

There has already been a large amount of research on strategies for improving the collaboration process that are widely used in scripted cooperation (see O'Donnell & King, 1999) or cooperative teaching (O'Donnell & Dansereau, 2000) research. These scripts mainly structure collaborative learning by assigning specific activities to the learners. Such activities are mainly content-independent; however, they are tailored to the task at hand, e.g., learning a theory or solving a problem. In many studies, beneficial effects of collaboration scripts were found in face-to-face scenarios (see O'Donnell & Dansereau, 1992; O'Donnell & King, 1999; Palincsar & Brown, 1984; Rosenshine & Meister, 1994). Recently, collaboration scripts have been increasingly used in text-based learning environments (see Weinberger, Fischer, & Mandl, this volume) and in videoconferencing (see Rummel & Spada, this volume). The main results of these studies show that collaboration scripts can have a beneficial influence on learning processes. However, there are no consistent results regarding learning outcomes.

Scripts may vary in many aspects. In the context of this chapter, we will focus on the aspects of *sequencing collaboration* and *strategy application.*

The aspect of sequencing collaboration is based on the script definition of Schank and Abelson (1977) stating that, when performing such procedures, it is advantageous to internalize routine procedures as a fixed script in memory. In the context of collaborative learning, this implies that once learners have internalized the script for performing a particular learning task (e.g., problem solving), they will be able to perform this task better in future situations. Furthermore, this kind of sequencing can serve as a model for learners to perform the task like an expert (see Collins, Brown, & Newman, 1989). The second aspect of strategy application means that learners are encouraged to apply beneficial learning activities when working collaboratively. In a meta-review, Rosenshine, Meister and Chapman (1996) found that, in particular, the strategies of summarizing and questioning provide beneficial learning activities for the learners.

2.2 Content schemes

While collaboration scripts encourage learners to focus their attention on specific activities, there are also strategies for encouraging learners to focus on specific contents. Brooks and Dansereau (1983) call them content schemes. Such schemes provide the representational context for a task by providing placeholders for important dimensions of content. Providing external schemes can modify the representational context of a task. According to Zhang and Norman (1994), modifying the representational context of a task may also change learners' subjective representation of this task and thereby influence their ability to solve the task. The modified representational context of a task may not only affect the learners' task solution when using this external scheme, but may also have an effect without the scheme.

Using content schemes in collaboration means that no specific activities are assigned to the learners, but that learners gain an increased awareness of important concepts and categories of the subject matter. The awareness of particular contents focuses the learning process on these contents and ensures that these contents receive increased attention. Usually, the specific contents are displayed permanently during the learning session, either on a style sheet (see Brooks & Dansereau, 1983) or within the user interface on the computer (see Fischer, Bruhn, Gräsel, & Mandl, 2002; Slotta & Linn, 2000; Suthers & Hundhausen, 2001). In this context, Suthers and Hundhausen (2001) refer to the concept of salience: Through permanent display, these contents remain salient during collaboration. Due to this salience, Suthers and Hundhausen

(2001) postulate a "representational guidance" that states that the representation of these concepts can guide and focus learners. This representational guidance can be an important mechanism for supporting the collaborative learning process. However, representational guidance also implies that the representation must be present to have an effect. For this reason, representational guidance and salience may be adequate support mechanisms *during* collaboration, but their effects *after* collaboration remain unclear (see also Salomon, 1992). For making predictions about long-term effects, one could rely on the concept of the representational guidance or look to the concept of the modified representational context. Considering the concept of representational guidance, content schemes may have effects similar to textual cues, prompts or scaffolds, which are only temporary support for the learners (see Rosenshine & Meister, 1992). In contrast, assuming that content schemes modify the representational context of a task, learners would perceive this task differently *every time* they work on this task – even if the content scheme is not present.

Until recently, the effects of content schemes were mainly studied within the context of individual learning (see Brooks & Dansereau, 1983; Kotovsky & Fallside, 1989; Kotovsky, Hayes, & Simon, 1985; Larkin, 1989; Zhang & Norman, 1994; Zhang, 1997) and only little was known about the effects of such mechanisms on collaborative problem solving (see Fischer et al., 2002; Suthers & Hundhausen, 2001). Results of the Fischer et al. (2002) study that investigates the effects of structural visualization similar to mapping, indicate that the content scheme is beneficial to the learning process and to collaborative learning outcomes. Suthers (2001) also reported similar results with respect to tabular schemes. However, there is little information regarding the influence of content schemes on collaborative learning in videoconferencing. Fischer et al. (2000) and Bruhn (2000) discovered that content schemes modified the collaborative learning processes in videoconferencing; however, content schemes did not seem to affect collaborative or individual learning outcomes. Since shared applications play a very prominent role in videoconferencing, the assumption can be made that interventions implemented through a shared application could be quite beneficial for collaborative learning scenarios.

3. RESEARCH QUESTIONS

Our basic research question examines the degree to which support measures implemented in the shared application – such as content schemes – affect collaborative and individual learning outcomes during collaborative learning in videoconferencing. The next objective is to discover the degree to which these effects differ from the effects of well-known support measures such as collaboration scripts. For this reason, we will compare two collaboration scripts with two content schemes that have been specifically designed for learning in videoconferencing. A further consideration focuses on the potential interactions between the collaboration script and the content scheme. Finally, we present our results concerning individual and collaborative learning outcomes.

The studies were conducted during the last few years. Study 1 analyzed the effects of a collaboration script and a content scheme on collaborative teaching (see table 13-1; Ertl, Reiserer, & Mandl, 2005). Study 2 was centered around the effects of a collaboration script and a content scheme on a collaborative problem-solving scenario (Kopp, Ertl, & Mandl, 2004).

Table 13-1. Participants, task and subject matter of the two studies.

	Participants	Task	Subject matter
Study 1	86 (43 Dyads)	Collaborative Teaching	Theory of Genotype Environment Effects
Study 2	159 (53 Triads)	Problem-Solving	Attribution Theory

In the following sections, we will first describe each individual study answering the following research questions:

- To what extent do collaboration scripts affect collaborative and individual learning outcomes in videoconferencing?
- To what extent do content schemes affect collaborative and individual learning outcomes in videoconferencing?

Then we will compare the results of the two studies regarding the influence of the different types of support within the two studies.

4. STUDY 1

The particular aim of Study 1 was to discover the degree to which a collaboration script and a content scheme, when used within a dyadic collaborative teaching scenario, could foster learners' collaborative and individual knowledge acquisition in videoconferencing.

4.1 Method of study 1

In study 1, learners worked within a collaborative teaching scenario. To understand the contents of a theory text, one learner took the role of a teacher while the other learner assumed the role of a learner. In preparation for the collaboration, the learner in the teacher role worked on a text individually. This text dealt with the theory of Genotype Environment Effects (see Scarr, 1984), which contained both theoretical concepts and evidence. After this individual preparation, the collaboration started. During collaboration, the person learning from the text functioned as the teacher, while the second person assumed the role of the learner. Both learners had the task (1) to study the most important contents of the theory text, both theoretical concepts and evidence and (2) to discuss their own reflections, ideas and comments on the subject. To achieve this, the learner in the teacher role was asked to explain the contents of the theory text to the learner in the learner role. Using a shared application (text editor), both learners had the opportunity to collaboratively create shared representations of theoretical concepts, evidence and personal elaborations, such as the consequences of the theory or their personal opinion.

Dyads were randomly assigned to one of four conditions. The first two conditions involved either the factor of collaboration script or content scheme. In a further condition, collaboration script and content scheme were combined, while learners in a control condition had no additional support.

Following the collaborative learning unit, domain-specific knowledge was assessed on an individual basis, without any of the support measures.

4.1.1 Collaboration script for collaborative teaching

The collaboration script structured the collaborative learning unit in two different respects. Firstly, it provided the learner with different phases in which to communicate the contents of the text. Furthermore, it provided specific activities for each phase to be undertaken by the learners in both the teacher and learner role. The *first* phase of the script served to facilitate the communication of the text by the teacher. The task of the learner in the teacher role was to explain the contents of the text. The partner in the learner role was asked to listen and to query the information as soon as anything was unclear. In the *second* phase, the learners deepened their comprehension of the text. In both phases, they worked together on a shared representation of the main contents of the text in the shared application. The partner in the learner role had the task of summarizing the contents and important points in the text editor; the teacher was given the task of supporting the learners in this activity. In the *third* phase of the script, both learning partners reflected individually on the contents of the text and considered any unanswered questions. In the *fourth* phase, the learners discussed the text and were given time for individual reflection. Then the partner in the learner role was tasked with capturing important notes from the discussion as a shared representation.

4.1.2 Content scheme for collaborative teaching

In the condition where learners used the content scheme, the scheme structured the shared application during the collaborative learning unit. Using this scheme, both partners were asked to consider the following categories: *theoretical concepts*, *evidence*, *consequences* and *personal opinion*. However, the condition did not explicitly sequence these topics or specify which of the partners should fill in the scheme. Both partners were asked to describe basic theoretical concepts in the category entitled *theoretical concepts* and present studies that supported the theory in the category entitled *evidence*. They used the category entitled *consequences* to capture personal elaborations on the applicability and limitations of the theory. The category entitled *personal opinion* allowed learners to present a personal evaluation of the theory and to provide an assessment. The scheme helped both partners to differentiate between theoretical concepts and evidence and supported them in their personal elaborations. The abstract categories were made more concrete by the questions contained in each category (see table 13-2).

Table 13-2. Structure of the content scheme

Theoretical concepts	*Evidence*
What are the core concepts of the theory? What are the most important statements of the theory?	How was the theory examined? Which findings support the thveory?
Consequences	*Personal opinion*
Which pedagogical interventions can be derived from the theory? Which limitations of pedagogical interventions are set by the theory?	What do we like about the theory? What do we not like? Which of our own experiences confirm the theory? Which of our own experiences contradict the theory?

4.1.3 Instruments

In order to measure the collaborative learning outcome, the concepts that were written down in the shared application were analyzed with respect to the areas of theoretical concepts, evidence and personal elaborations. These units of meaning were either summed together into a score for theoretical concepts or for evidence. For the evaluation of the personal elaborations, a similar method was employed. The sum was made of all comprehensibly elaborated units of meaning in the document. The individual learning outcome was measured by free recall; learners were asked to write down the most important contents of the theory text from memory. This test was also utilized for theoretical concepts and evidence.

4.2 Results of study 1

The *collaborative learning outcome* comprised the areas of theoretical concepts, evidence and personal elaborations. There were significant effects of both interventions in the area of theoretical concepts. On the one hand, the use of the collaboration script for collaborative teaching resulted in learners capturing significantly more units of meaning in this area (see figure 13-1). On the other hand, the use of the content scheme for collaborative teaching led learners to capture significantly less units of meaning in the area of theoretical concepts. There were no significant differences regarding evidence. With respect to personal elaborations, results indicate a clear effect of the content scheme: learners with the scheme externalized significantly more elaborations than learners without the scheme.

In addition, there was a significant interaction effect between scheme and script. The interaction indicates that the combination of both support methods resulted in the most adequate solution of the task by drawing attention to theoretical concepts, evidence and personal elaborations. Regarding absolute values of all categories, learners in the control group captured the least number of units of meaning while learners with content scheme and collaboration script performed best.

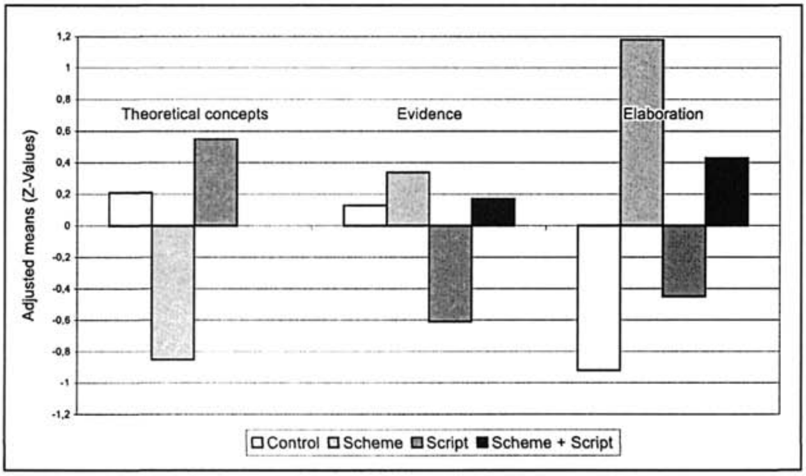


Figure 13-1. Collaborative learning outcome of study 1.

In summary, the collaboration script and content scheme had different effects on the collaborative learning outcome, that is, on the group product both learners built together. Learners with the script worked more on theoretical concepts than other learners. In contrast, learners with the content scheme worked less on theory concepts, but much more on personal elaborations. When both support measures were combined, these effects counterbalanced one another and learners with both support measures achieved the most adequate task solution.

With respect to *individual learning outcomes*, the results of the learners in the learner role are described, because only these results reflect the effects of the collaborative learning unit. In the area of theoretical concepts, the collaboration script or the content scheme had no significant effects. The factors of collaboration script or content scheme also had no significant effect on evidence.

4.3 Discussion of study 1

The results of Study 1 show the effects of both the collaboration script and the content scheme with respect to the collaborative learning outcome. The collaboration script was shown to be especially effective in the area of theoretical concepts. This result may be attributed to the structure of the collaboration script, which encouraged learners to deal with the core of the theory twice: once in the first phase of the script, when the learner in the learner role was explained the text material, and again in the second phase, when the learner in the learner role had to recall and record them. This learning by teaching (see Renkl, Mandl, & Gruber, 1996) may have led to a higher activity level of the learner in the learner role (see Reiserer, 2003) who had to document the collaborative learning outcome. The content scheme mainly influenced the area of personal elaborations. Learners with content scheme benefited from the mechanisms of representational guidance (see Suthers & Hundhausen, 2003) and focused on elaborations. For this reason, they tended to neglect theoretical concepts.

There were no apparent effects on individual learning outcomes. Thus, the question arises as to why the clear effects of the interventions on the collaborative learning outcomes were not evident in the individual learning outcomes. This may be, on one hand, related to the support of collaborative teaching and differences in the concepts of collaborative and individual learning outcomes (see Hertz-Lazarowitz et al., 1992; Salomon & Perkins, 1998). As the support measure was aimed at the teaching process, it may not have been helpful for individual learning transfer (see Ertl, Fischer, & Mandl, 2006). On the other hand, the results relating to the content scheme can be explained with the mechanisms of representational guidance (Suthers & Hundhausen, 2001). Important concepts were salient during the process but not during the individual posttest. In a manner similar to cues, the content scheme of the first study was a temporary support measure (see Rosenshine & Meister, 1992), which was only effective during the time the learners got it presented. For this reason, the content scheme was not able to change the learners' perception of the task according to Zhang and Norman (1994).

Consequently, for the learners in Study 2, we chose a different task: collaborative problem solving. We expected that strategies which were seen as helpful during collaborative problem solving would also be applied during the individual problem solving process after collaboration. Furthermore, we anticipated that we could design a content scheme that would change learners' perception of the task and thus have a lasting effect. For the problem solving activity, we decided to work with triads for a more stimulating discussion process.

5. STUDY 2

The aim of Study 2 was to investigate the effects of a collaboration script and a content scheme on collaborative problem solving in videoconferencing triads.

5.1 Method of study 2

The learning environment consisted of an individual and a collaborative learning unit. At the beginning of the exercise, learners worked individually on a text about the Attribution Theory with core concepts according to Heider (1958) and Kelley (1973). In the collaborative learning unit, all three learners worked together on the solution of a learning case. They were given case material, which contained somewhat different information for each learner. The learners' task was to discuss the case according to the Attribution Theory and to find evidence from the case material and relate it to theoretical concepts. Doing this, they had to name causes for success or failing, to find case information about the consensus and about consistency of the cause and to determine the attribution accordingly. At the end of discussion, learners were asked to document a case solution in the shared application (text editor). A collaboration script and a content scheme were used as support measures during the collaborative learning unit. In a further condition, the collaboration script and the content scheme were combined, while learners in the control condition had no additional support. After collaboration, each learner worked on a case individually and without any further support.

5.1.1 Collaboration script for collaborative problem solving

The collaboration script structured the collaborative unit in four phases. In the *first* phase, learners were asked to read case material and to work individually to extract information that was important for the case solution. In the *second* phase, the learners had to exchange information and to collaboratively resolve comprehension questions. They used a shared application for documenting concepts that were important for the case solution. In the *third* phase, learners had to reflect individually about the comprehensiveness of the information they had collected. During the *fourth* phase, the learners had to collaboratively develop the case solution.

5.1.2 Content scheme for collaborative problem solving

The participants using the content scheme received a pre-structured shared application (text editor). Learners followed the structure of the table, which was divided into three main categories (see table 13-3): *Cause,* for collecting the causes for the problem described in the case, *Information* for finding case information providing evidence for the cause and *Attribution* for making the correct attribution of the cause. The categories *Information* and *Attribution* each contained two subcategories: *Information* was divided into columns for *Consensus* and *Consistency* for making these two aspects of the attribution theory salient. *Attribution* was split into *Kelley* and *Heider* for guiding learners to attribute according to both theories. Using this scheme, learners would be able to record complete attributions according to Kelley and Heider with causes and case information about consensus and consistency.

Table 13-3. Structure of the content scheme

Cause	*Information about*		*Attribution according to*	
	Consensus	*Consistency*	*Kelley*	*Heider*

5.1.3 Instruments

In order to measure the collaborative learning outcome, the contents of the shared application were analyzed. A coding system was developed in accordance with the different categories of the attribution theory in which all causes, information and attributions were listed in an identifiable way without any overlap. Case information and theory concepts were assessed and each category summed up into a single score. A short case after collaboration was used to measure individual learning outcome. The analysis of this case was similar to the collaboratively solved case: scores were given for case information and theoretical concepts. Points for each category were then summed up as a single score. During the individual post-test case, learners received no support.

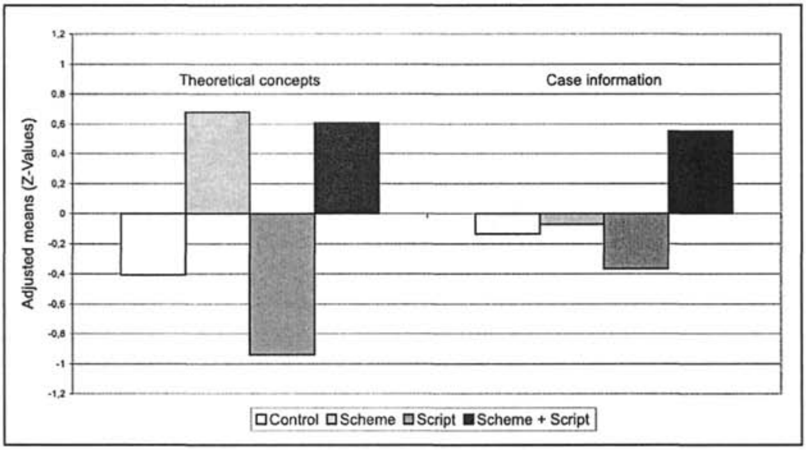


Figure 13-2. Collaborative learning outcome of study 2

5.2 Results of study 2

The content scheme had a strong and significant effect on the *collaborative learning outcome*. Learners with the scheme externalized nearly double the amount of theoretical concepts than learners without the scheme. Regarding the collaboration script and regarding case information, there were no significant effects. Descriptively, learners with content scheme and collaboration script scored the highest, while learners using only the collaboration script scored the lowest (see figure 13-2).

Regarding *individual learning outcomes*, the content scheme also proved to be highly influential. In the category of theoretical concepts, learners with the content scheme achieved a significantly higher score. The collaboration script had no significant effect. However, regarding all outcome measures, learners who were given both the collaboration script and the content scheme descriptively scored the highest.

5.3 Discussion of study 2

In summary, we can conclude that the content scheme greatly influenced collaborative and individual knowledge acquisition, particularly in the category of theoretical concepts. The effect during the process may be attributed to the salience of the relevant categories. Learners who used the content scheme may have experienced representational guidance (see Suthers & Hundhausen, 2001). However, even without the scheme, learners internalized these categories and applied them individually. Therefore, one can assume that the content scheme was able to modify the learners' perception of the task (see Zhang & Norman, 1994). Learners perceived the task of performing an attribution differently; in particular to find causes, evaluate consensus and consistency of the causes and decide on an attribution based on these evaluations (see Ertl et al., 2006).

The collaboration script had no significant effect. The reason for this may lie in the quite general structure of the collaboration script. These general activities do not appear to be very helpful when used without content-specific context. According to the script conceptualization of Schank and Abelson (1977), learners might also have to use this script more often to internalize it and benefit from it. However, results revealed that the collaboration script was able to further improve the effects of the content scheme. Thus, both support measures should be implemented together to foster the acquisition of knowledge.

6. GENERAL DISCUSSION

The aim of this chapter was to analyze the effects of collaboration scripts and content schemes on collaborative learning in videoconferencing. Furthermore, the collaboration script was compared with the content scheme to examine which would best support collaborative learning in videoconferencing.

The *collaboration scripts* used in both studies were of a similar structure to the scripts known from Scripted Cooperation (see O'Donnell & Dansereau, 1992) or Reciprocal Teaching (see O'Donnell & Dansereau, 2000). However, due to the nature of videoconferencing, these scripts were used only as a guideline without the specific training that is given when these scripts are utilized in face-to-face scenarios. Therefore, the collaboration scripts as they were used in videoconferencing may have not been as beneficial as would be expected in face-to-face scenarios (see O'Donnell & Dansereau, 1992, 2000; O'Donnell & King, 1999; Palincsar & Brown, 1984; Rosenshine & Meister, 1994). On the other hand, this may also be one limitation of videoconferencing. When using videoconferencing, it is not possible to conduct training in the same manner as in face-to-face settings. This raises the question of how relevant scripts are for videoconferencing. However, using the same guideline several times in videoconferencing scenarios may encourage learners to internalize it as a script. After a period of time, this internalized script may be able to support learners (see Schank & Abelson, 1977).

The *content schemes* proved to be highly effective in Study 1 as well as in Study 2. In Study 1, the scheme worked rather temporarily, mainly to benefit collaborative learning outcomes. In Study 2, the scheme had a positive effect on both collaborative and individual learning outcomes. These differences may be attributed to the different tasks in the studies (see Ertl et al., 2006). In Study 2, learners received a scheme for problem solving. Using the scheme, they were provided with a strategy for performing a complete attribution, which was independent of whether they used it individually or collaboratively. The scheme used in Study 1, however, was aimed directly at the collaborative teaching process and particularly focused learners' attention on empirical studies and elaborations. Learners may not have found this focus to be helpful for individual theory recall. When analyzing the mechanisms of content schemes, we could assume that some kind of "representational guidance" (Suthers, 2001) took place during the collaboration. Due to this representational guidance, learners were able to successfully focus on particular aspects of the task while using the content scheme. In Study 2, we also can assume that, for the learners using the content scheme, the modified representational context also modified their perception of the problem-solving task (see Zhang & Norman, 1994). Thus, the different perception of the task affected the learners working collaboratively as well as individually in the post-test, when no scheme was present. In conclusion, one can assert that content schemes that use the mechanisms of representational guidance (see Suthers & Hundhausen, 2003) have a significant effect when learners are working *with* them (see Salomon, 1992). After collaboration, content schemes may also have an effect, but only if the scheme had the ability to change learners' perception of the task (see Zhang & Norman, 1994).

In both studies, we gathered evidence that supported the notion that the collaboration script could enhance the effects of the content scheme. In Study 1, we observed an interaction that helped learners to achieve a more adequate collaborative learning outcome. In Study 2, learners who were using both support measures scored highest according to all outcome measures. Our conclusion is that, to be effective, collaboration-specific support needs a content-specific basis.

Further analyses showed that both interventions influenced the spoken discourse of the learners (see Ertl, Kopp, & Mandl, in press; Kopp, 2005; Reiserer, 2003). However, further investigation is required to examine how this modified learning discourse is related to learning processes and outcomes.

7. CONCLUSIONS

Based on these results, three main conclusions can be drawn. First of all, interventions which are very effective during the collaboration process may fail to benefit individual learning outcomes. This may occur when it is difficult to assess the goals of the collaboration following the collaboration. An improved teaching process, for example, may simply improve the teaching process itself, but not result in better outcomes concerning the material taught (see Ertl et al., 2006; Salomon, 1992). Thus, it is necessary to develop an approach for process evaluation which describes helpful activities in the learning process that are independent of learning outcomes. Such helpful skills may be, for example, that learners are better able to differentiate between theoretical concepts and empirical evidence (see Ertl, 2003; Kuhn, Weinstock, & Flaton, 1994; Sodian, Zaitchik, & Carey, 1991), that they use better, more scientific argumentation (see Leitão, 2000) or that they collaborate with less social conflicts (see Bales, 1950). These effects, which can be viewed as "side effects" of collaborative learning, should be considered for future investigations.

Secondly, combined support measures centered on collaboration and content-specific strategies should be applied to support learners. For the purpose of empirical research, it may be necessary to split these factors to avoid confounding them. However, focusing solely on content or on collaboration strategies may not appropriately meet the support needs of computer-supported collaborative learning environments.

Thirdly, further research should be conducted on the degree to which interventions applied in collaborative learning can prompt sustainable learning strategies. Fostering learning outcomes *once* is certainly important for collaborative learning. However, thoughtfully designed interventions should also have a lasting effect to benefit learners when solving similar tasks in the future.

ACKNOWLEDGEMENTS

This research was funded by DFG (Deutsche Forschungsgemeinschaft), project numbers MA 978/13-1, MA 978/13-3, MA 978/13-4.

REFERENCES

Anderson, A. H., O' Malley, C., Doherty Sneddon, G., Langton, S., Newlands, A., Mullin, J., Fleming, A. M., & Van der Velden, J. (1997). The impact of VMC on collaborative problem solving: An analysis of task performance, communicative process, and user satisfaction. In K. E. Finn, A. J. Sellen, & S. Wilbur (Eds.), *Video mediated communication* (pp. 133-155). Mahwah, NJ: Erlbaum.

Anderson, J. R., Reder, L. M., & Simon, H. A. (1996). Situated learning and education. *Educational Researcher, 25*, 5-11.

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

Bales, R. F. (1950). *Interaction process analysis: A method for the study of small groups.* Cambridge, MA: Addison-Wesley.

Brooks, L. W., & Dansereau, D. F. (1983). Effects of structural schema training and text organization on expository prose processing. *Journal of Educational Psychology, 75*, 811-820.

Bruce, V. (1996). The role of the face in communication: Implications for videophone design. *Interacting with Computers, 8*, 166-176.

Bruhn, J. (2000). *Förderung des kooperativen Lernens über Computernetze. Prozess und Lernerfolg beim dyadischen Lernen mit Desktop-Videokonferenzen [Fostering cooperative learning in computer networks: Process and learning outcome in dyadic learning in desk-top-videoconferencing]*. Frankfurt am Main: Peter Lang.

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

Cohen, E. G., & Lotan, R. A. (1995). Producing equal-status interaction in the heterogeneous classroom. *American Educational Research Journal, 32*, 99-120.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Erlbaum.

Dillenbourg, P. (1999). What do you mean by 'collaborative learning'? In P. Dillenbourg (Ed.), *Collaborative-learning: Cognitive and computational approaches* (pp. 1-19). Oxford: Elsevier.

Dillenbourg, P., & Traum, D. (1999, January). *Does a shared screen make a shared solution?* Paper presented at the Conference on Computer Support for Collaborative Learning (CSCL), Stanford.

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

Ertl, B. (2003). *Kooperatives Lernen in Videokonferenzen. Förderung von individuellem und gemeinsamem Lernerfolg durch external repräsentierte Strukturangebote [Cooperative learning in videoconferencing. Support of individual and cooperative learning outcomes by representational aids]*. Doctoral thesis. Ludwig-Maximilian-University, Munich. Available at: http://edoc.ub.uni-muenchen.de/archive/00001227/01/Ertl_Bernhard_M.pdf.

Ertl, B., Fischer, F., & Mandl, H. (2006). Conceptual and socio-cognitive support for collaborative learning in videoconferencing environments. *Computers & Education, 47*(3), 289-315

Ertl, B., Kopp, B., & Mandl, H. (in press). Fostering case based learning in videoconferencing. *Journal of Educational Computing Research.*

Ertl, B., Reiserer, M., & Mandl, H. (2005). Fostering collaborative learning in videoconferencing: The influence of content schemes and cooperation scripts on shared external representations and individual learning outcomes. *Education, Communication & Information, 5*, 147-166.

Finn, K. E., Sellen, A. J., & Wilbur, S. B. (Eds.). (1997). *Video-mediated communication.* Mahwah, NJ: Lawrence Erlbaum Associates.

Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. (2000). Kooperatives Lernen mit Videokonferenzen: Gemeinsame Wissenskonstruktion und individueller Lernerfolg [Collaborative knowledge construction. Analysis and facilitation in computer-supported collaborative scenarios]. *Kognitionswissenschaft, 9*, 5-16.

Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. (2002). Fostering collaborative knowledge construction with visualization tools. *Learning and Instruction, 12*, 213-232.

Greeno, J. G. (1997). On claims that answer the wrong questions. *Educational Researcher, 26*, 5-17.

Heider, F. (1958). *The psychology of interpersonal relations.* New York: Wiley.

Hertz-Lazarowitz, R., Kirkus, V. B., & Miller, N. (1992). Implications of current research on cooperative interaction for classroom application. In R. Hertz-Lazarowitz (Ed.), *Interac-*

tion in cooperative groups: The theoretical anatomy of group learning (pp. 253-280). New York, NY: Cambridge University Press.

Isaacs, E. A., & Tang, J. C. (1997). Studying video-based collaboration in context: From small workgroups to large organizations. In K. E. Finn, A. J. Sellen, & S. Wilbur (Eds.), *Video mediated communication* (pp. 173-197). Mahwah, NJ: Erlbaum.

Jeong, H., & Chi, M. T. H. (1999). *Construction of shared knowledge during collaborative Learning*. Paper presented at the AERA Annual Meeting, Montreal.

Joiner, R., Scanlon, E., O'Shea, T., Smith, R. B., & Blake, C. (2002). Synchronous collaboration support for adults evidence from a series of experiments on videomediated collaboration: Does eye contact matter? In G. Stahl (Ed.), *Computer support for collaborative learning: Foundations for a CSCL community (pp. 371-378). Proceedings of the conference on Computer Support for Collaborative Learning (CSCL) 2002, Boulder, CO.* Hillsdale: Erlbaum.

Kelley, H. H. (1973). The processes of causal attribution. *American Psychologist, 28*, 107-128.

Kopp, B. (2005). *Effekte schematheoretischer Unterstützung auf Argumentation und Lernerfolg beim kooperativen Lernen in Videokonferenzen [Effects of schemata on argumentation and learning outcomes in videoconferencing]*. Berlin: Logos.

Kopp, B., Ertl, B., & Mandl, H. (2004). Fostering cooperative case-based learning in videoconferencing: Effects of content schemes and cooperation scripts. In P. Gerjets, P. Kirschner, J. Elen, & R. Joiner (Eds.), *Instructional design for effective and enjoyable computer- supported learning. Proceedings of the first joint meeting of the EARLI SIGs Instructional Design and Learning and Instruction with Computers (CD-ROM)*. Tuebingen: Knowledge Media Research Center.

Kotovsky, K., & Fallside, D. (1989). Representation and transfer in problem solving. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon* (pp. 69-108). Hillsdale, NJ: Erlbaum.

Kotovsky, K., Hayes, J. R., & Simon, H. A. (1985). Why are some problems hard? Evidence from Tower of Hanoi. *Cognitive Psychology, 17*, 248-294.

Kuhn, D., Weinstock, M., & Flaton, R. (1994). Historical reasoning as theory-evidence coordination. In M. Carretero & J. F. Voss (Eds.), *Cognitive and Instructional Processes in History and the Social Sciences* (pp. 377-401). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lambiotte, J. G., Dansereau, D. F., O'Donnell, A. M., Young, M. D., Skaggs, L. P., & Hall, R. H. (1988). Effects of cooperative script manipulations on initial learning and transfer. *Cognition and Instruction, 5*, 103-121.

Larkin, J. H. (1989). Display-based problem solving. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon* (pp. 319-341). Hillsdale, NJ: Erlbaum.

Leitão, S. (2000). The potential of argument in knowledge building. *Human Development, 43*, 332-360.

Nastasi, B. K., & Clements, D. H. (1992). Social-cognitive behaviours and higher-order thinking in educational computer environments. *Learning and Instruction, 2*, 215-238.

O'Connaill, B., Whittaker, S., & Wilbur, S. (1993). Conversations over video conferences: An evaluation of the spoken aspects of video-mediated communication. *Human Computer Interaction, 8*, 389-428.

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

O'Donnell, A. M., & Dansereau, D. F. (2000). Interactive effects of prior knowledge and material format on cooperative teaching. *Journal of Experimental Education, 68*, 101-118.

O'Donnell, A. M., & King, A. (Eds.). (1999). *Cognitive perspectives on peer learning*. Mahwah, NJ: Erlbaum.

Pächter, M. (2003). *Wissenskommunikation, Kooperation und Lernen in virtuellen Gruppen [Knowledge communication, cooperation and learning in virtual groups]*. Lengerich: Pabst.

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 1*, 117-175.

Piaget, J. (1932). *The moral judgment of the child*. London: Routledge & Kegan Paul.

Reiserer, M. (2003). *Peer-Teaching in Videokonferenzen. Effekte niedrig- und hochstrukturierter Kooperationsskripte auf Lernprozess und Lernerfolg [Peer-teaching in videoconferencing. Effects of low- and high structured cooperation scripts on learning process and learning outcomes]*. Berlin: Logos.

Renkl, A., Mandl, H., & Gruber, H. (1996). Inert knowledge: Analyses and remedies. *Educational Psychologist, 31*, 115-121.

Roschelle, J., & Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem solving. In C. O'Malley (Ed.), *Computer Supported Collaborative Learning*. (pp. 69-97). Berlin: Springer.

Rosenshine, B., & Meister, C. (1992). The Use of Scaffolds for Teaching Higher-Level Cognitive Strategies. *Educational Leadership*, 26-33.

Rosenshine, B., & Meister, C. (1994). Reciprocal teaching: A review of the research. *Review of Educational Research, 64*, 479-530.

Rosenshine, B., Meister, C., & Chapman, S. (1996). Teaching students to generate questions: A review of the intervention studies. *Review of Educational Research, 66*, 181-221.

Salomon, G. (1992). Effects with and of computers and the study of computer-based learning environments. In E. De Corte, M. C. Linn, H. Mandl, & L. Verschaffel (Eds.), *Computer-based learning environments and problem solving* (pp. 249-263). Berlin: Springer.

Salomon, G. (1998). Novel constructivist learning environments: Some issues to be concerned with. *Research Dialogue in Learning and Instruction, 1*, 2-12.

Salomon, G., & Globerson, T. (1989). When teams do not function the way they ought to. *International Journal of Educational Research, 13*, 89 - 99.

Salomon, G., & Perkins, D. N. (1998). Individual and social aspects of learning. *Review of Research in Education, 23*, 1-24.

Scarr, S. (1984). Individuelle Erfahrungsunterschiede: Anlage- und Umwelteinwirkungen auf die Entwicklung des Kindes. In W. E. Fthenakis (Ed.), *Tendenzen der Frühpädagogik* (pp. 233-225). Düsseldorf: Schwann.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding*. Hillsdale, NJ: Erlbaum.

Slavin, R. E. (1995). *Cooperative learning: Theory, research and practice*. Needham Heights, MA: Allyn & Bacon.

Slotta, J. D., & Linn, M. C. (2000). The knowledge integration environment: Helping students use the internet effectively. In M. J. Jacobson & R. B. Kozma (Eds.), *Innovations in science and mathematics education: Advanced designs, for technologies of learning* (pp. 193-226). Mahwah, NJ: Erlbaum.

Sodian, B., Zaitchik, D., & Carey, S. (1991). Young children's differentiation of hypothetical beliefs from evidence. *Child Development, 62*, 753-766.

Suthers, D. (2001). Towards a systematic study of representational guidance for collaborative learning discourse. *Journal of Universal Computer Science, 7*, 254-277.

Suthers, D. D., & Hundhausen, C., D. (2001). Learning by constructing collaborative representations: An empirical comparison of three alternatives. In P. Dillenbourg, A. Eurelings, & K. Hakkarainen (Eds.), *Procedings of the First European Conference on Computer-Supported Collaborative Learning (euroCSCL)* (pp. 577-584). Maastricht: McLuhan Institute.

Suthers, D. D., & Hundhausen, C., D. (2003). An experimental study of the effects of representational guidance on collaborative learning processes. *The Journal of the Learning Sciences, 12*, 183-218.

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

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

Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive Science, 21*, 179-217.

Zhang, J., & Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science, 18*, 87-122.

Chapter 14

THE ROLES OF SCRIPTS IN PROMOTING COLLABORATIVE DISCOURSE IN LEARNING BY DESIGN

Janet L. Kolodner
College of Computing/Georgia Institute of Technology, Atlanta

Abstract: Using the design of Learning by Design (LBD) for illustration and results of its enactments as evidence, I make an argument about the roles Schank and Abelson's (1977) kind of scripts can play in promoting collaborative discourse and present a way of promoting the kind of script learning that results in productive collaborative discourse. LBD's way of promoting script learning has three parts to it: (i) a set of scripted activity structures and sequences (classroom scripts) that promote productive and appropriate participation in classroom practices (including collaborative discourse), (ii) an approach to instruction that focuses on repeated, deliberative practice of each of these classroom scripts, and (iii) an approach to getting started through *launcher units* that introduce the scripted activity structures, their sequencing, and how to participate in each. I argue that scripts students learn for participating in classroom practices can play three roles in promoting collaboration and collaborative learning: (i) they help students participate in whole-class discussions and in discursive practices by proposing sequencing for their discourse, (ii) they help students participate in whole-class discussions and discursive practices by proposing content for their discourse, and (iii) they provide focus for small group discourse as students aim their discussion toward fulfilling a script's expectations in order to be able to participate in the script later. Learning by Design is a design-based approach to science learning.

Example 1: Children in an 8th grade class are presenting their experimental investigations in a poster session – investigations aimed at determining the effects of different characteristics of a balloon engine on the distance a vehicle will go. Their balloon engines are made by gluing a drinking straw to a balloon. They attach the engine to the vehicle by securing the drinking straw to a tower, and they use the straw to blow up the balloon that powers the vehicle. In their experiments, they compare the distance the vehicle travels under different conditions. During the poster session, they present their research question, their procedure, their results, a rule of thumb describing the trends in their results, a set of four force diagrams, each representing a different stage in the vehicle's motion, and the best explanation they can make of their results using what they've learned about combining forces. Group 1 (a group of 3 students) reports on an experiment investigating the effects of the length of the straw on how far a vehicle will go. After the presentation, a child in the class notices that the arrow representing friction in the last stage, when the car has stopped, is larger than the one representing friction earlier and asks why. The child who drew the arrow responds that it's long only because he was in a hurry. But he continues thinking about how long the gravity arrow should be and goes on:

> CF: But really uh, it should be shorter, because there (gestures to poster) wasn't any like, air mass in the balloon.

Other students in the class talk among themselves about this, and others continue to explain and clarify, spontaneously participating in a sense-making discussion. Sense making continues, moving on to consideration of the force of friction at stage 1, before the vehicle begins moving, with argument about whether there is a friction force at all if the vehicle isn't yet moving. Discussion continues. A child asks for clarification of why the shorter straw results in the vehicle going a longer distance. A member of the group describes it as a traffic jam. But another child in the class is worried about that explanation. After all, the engine with the shorter straw also had a smaller mass. She raises her hand, and one of the presenting students calls on her.

> JG: Don't you think when you're cutting the straw it's changing the mass?

The class spends some time trying to make sense of what she is trying to say. Discussion continues considering if there's a different way to run the experiment without changing two variables at the same time. Someone notices a way to do it, but then someone else notices that that procedure would be answering a different question. Someone in the presenting group notices that the problem with their experiment might be a problem in other experiments too, e.g., when comparing engines with different diameter straws. Nine students in all take part in this discussion, and the teacher doesn't contribute a word.

Example 2: A group of boys who had not had a chance to participate in a presentation of their ideas to the class (a pin-up session) nonetheless prepared a plan and discussed it thoroughly before moving forward to construct their best parachute. As they would have done had they participated with the class in a pin-up session, they drew a chart of their ideas, taped it to the wall, and stood around it discussing with each other the pros and cons of constructing their parachute a certain way. Our researcher asked the teacher what role she had played in the boys' decision to utilize the pin-up. The teacher responded, "Oh, they did that on their own. I didn't assign them to work on anything. ... They did the pin-up because they know that is part of the design process.... Now, it wasn't pretty or anything, but they did the sketches of their ideas." (Fasse, field notes, fall, 1999)

Example 3: The group we're looking in on here was getting ready for the poster session in Example 1. They had investigated the effect of the circumference of the balloon on the distance a vehicle would go, decided that their rule of thumb was that the larger the circumference of the balloon, the farther the car would go, and that they needed a scientific explanation. At the same time, they were drawing the force arrows that show the forces on their vehicles, and they were worrying about whether they would finish on time. Their dialogue isn't fluid, but it interweaves discussion of the three things they need to consider before presentation – their rule of thumb, the forces on their vehicle, and their scientific explanation of their results. Each member of the group is assigned a different task – coming up with the rule of thumb, drawing the force diagrams, or developing the scientific explanation – but they think out loud and all help each other with their tasks, sometimes completing each other's sentences, and often stopping to ask each other clarification questions and to grapple with a hard concept. For example:

AB: Well, there's... Well the force coming out of the end of the balloon.

...

KK: When you have a larger circumference, there, is more air...

...

KK: ...Ok, so does this make sense? When you have a larger circumference, there is more, um, force of the air being pushed out of the balloon?

...

JG: (at same time as CW) When you have a larger circumference, there is more in the... Engine...

These are examples typical of the kinds of discussions heard among children in Learning by Design (LBD) classrooms (Kolodner, Camp, et al., 2003; Kolodner, Gray, & Fasse, 2003). They aren't exactly the discussions learned adults would have; they aren't as fluid, and they get off track, but they represent quite sophisticated discourse for eighth graders (14 years old). The first example is of a typical *poster session*, a formal presentation session in which each small group in the class reports about its investigation. Each group prepares a poster showing the question they were trying to answer, their investigative procedure, their data, their interpretation of the data, and if they can, a "rule of thumb" representing trends they see in their data. They take turns making presentations to the class, pointing to the data on their poster. After each presentation, students in the class ask questions – about the validity of the procedure, about validity of the data, about trends extracted, and so on. Then the next group presents. And so on. At the end, the class summarizes the trends in the data together, in a whole-class discussion, and then attempts to apply science they've learned to explain those trends. This poster session happened in mid-November, approximately 3 months into the school year, and it was the children's fourth or fifth poster session. The second example happened approximately 2 months into a school year, after students had engaged in several of these presentation forums, and it is a description of an informal discussion between two students. I've used it before to show that students are indeed learning skills and practices of science and project work in transferable ways (Kolodner, Camp, et al., 2003), but it also shows an example of LBD students using learned scripts (Schank & Abelson, 1977) to guide their own conversation. The third example is of several students in a small group getting ready for the poster session in Example 1. Here, they are focusing their discussion on the points they will have to present to the class.

Students engage in several kinds of scripted presentation activities during their project-based inquiry work in LBD – in poster sessions, they present investigations and their findings; in pin-up sessions, they present ideas about how they will solve the challenge they've been given and justifications for those ideas; and in gallery walks, they present solutions in progress and talk about how well they work, why they might not be working as well as expected, and how they might make them better. In each, groups take turns presenting to the class, and after each presentation, class members question their peers and provide advice. In each, the sequence of events, the purpose of each event in the sequence, and some of the how-to's of carrying it out are told to the students before the first time it is enacted. Then students enact the sequences several times with coaching, sometimes reflecting on what they did well and what could be improved, almost always reflecting in some way on what they learned from the session. The content of the discourse during these sessions can be quite sophisticated, as illustrated in the first example. In addition, as they prepare for these sessions, they spend time discussing the things that they need to present – the science content and processes they are learning – as shown in the third example. Finally, they are able to engage in similar sessions on their own, and in these sessions, their discourse is guided directly by the scripts they learned, as shown in the second example.

My goal in this chapter is twofold: to make an argument about the roles Schank and Abelson's (1977) kind of scripts can play in promoting this kind of collaboration, and to present a way of promoting the kind of script learning that results in productive collaborative discourse. Our way of promoting script learning has three parts to it:

1. A set of scripted activity structures and sequences (classroom scripts) that promote productive and appropriate participation in classroom practices (including collaborative discourse)
2. An approach to instruction that focuses on repeated, deliberative practice of each of these classroom scripts
3. An approach to getting started through *launcher units* that introduce the scripted activity structures, their sequencing, and how to participate in each (in some sense, providing training at participating in the classroom scripts)

I will argue that scripts students learn for participating in classroom practices can play three roles in promoting collaboration and collaborative learning: (i) they help students participate in whole-class discussions and in discursive practices by proposing sequencing for their discourse (as in Example 2), (ii) they help students participate in whole-class discussions and discursive practices by proposing content for their discourse (as in Example 1), and (iii) they provide focus for small group discourse as students aim their discussion toward fulfilling a script's expectations in order to be able to participate in the script later (as in Example 3).

I begin with the definition of script that I am using; it is one that is consistent with Schank and Abelson's (1977) original script definition but more fully incorporates the notions of participation and practice from the sociocultural tradition. I follow that with a description of Learning by Design, some of the scripted activity structures (classroom scripts) that comprise its practice, and our intentions with respect to the roles of several of those activity structures in promoting productive collaborative discourse. I move on to a description of how LBD tries to promote the learning of the scripts (instructional strategies, including launcher units) that enact these activity structures and sequences and then show examples that illustrate the roles LBD's scripted activity structures might play in promoting collaboration and collaborative learning. I end with discussion of lessons learned about how to design and enact classroom scripts to promote collaborative discourse.

1. CONCEPTUAL BACKGROUND ON SCRIPTS

1.1 Scripts as cognitive structures that promote productive participation

Schank and Abelson's (1977) theory of scripts and other knowledge structures proposes that people learn the sequences of events in common activities through participating in those activities. According to Schank and Abelson, one way people naturally learn how to participate in commonly occurring situations is to experience those situations repeatedly, generalizing a routine sequence of events and roles they and others play in those situations. We might observe, try out simple roles ourselves, get instruction or help from others in playing those roles, have sequences explained to us, and so on, gradually becoming more expert and better participants over time. We learn about the sequencing of events in a restaurant, for example, by going to restaurants and experiencing that sequencing. We learn how to participate in social events, such as going to a restaurant, by going with others who know the sequencing and roles, observing their actions, and eventually playing roles in the same ways we've observed, sometimes with some instruction. As we participate repeatedly in the same scripted events and experience the variations, we become expert at the common sequence of events and some of their variations, and we learn connections between events in the script, the purposes of some, what to expect from others, and the roles we should play and how to play them. In this way, we become fluent at being restaurant customers or at buying things in stores, getting up and dressed in the morning, going to birthday parties, entertaining guests, and so on, constructing in our memories cognitive structures that we call scripts (Schank & Abelson, 1977), each associated with a common kind of activity we participate in.

Schank's (1982, 1999) focus on dynamic memory attempts to explain how that learning happens; he, in essence, suggests a computational account of Piaget's accommodation and assimilation, proposing scripts as specific types of schemas that play a role in learning about activities in the context of participating in, observing, and hearing about those activities. According to his account, we pull out the regularities and make them into scripts while we notice the differences (things we were not expecting) and use explanations of those differences to index events that are different from the script. In this way, we create knowledge structures that specialize scripts to particular situations, combine scripts together to describe more complex situations, and provide access to experiences that violated those conventions. We can thus use scripts, cognitive artifacts derived from participation in culturally-common events, to get around in the world, anticipating what comes next, playing our roles appropriately, and anticipating and knowing how to deal with common script violations and variations (even scripting those). Activities we are familiar with become easy to participate in through creation of these knowledge structures.

It is not hard to wed Schank and Abelson's cognitive notion of scripts to Lave and Wenger's (1991) accounts, from a socio-cultural viewpoint, of learning through participation. Lave and Wenger's (1991) accounts of apprenticeship and legitimate peripheral participation focus on the social interactions and environmental factors that allow new members of a community to learn community practices through participation (directly and through observation). Apprentices participate first peripherally but observing the whole process and gradually taking on more responsibilities and adding to their repertoire and expertise, within the bigger context of the "shop". According to this viewpoint, and similar to Schank and Abelson's conception of scripts, we learn as a consequence of engaging with others in carrying out cognitive and social practices. Schank and Abelson concentrate on the cognitive structures that allow individuals to do the reasoning they need to do to participate; Lave and Wenger (1991) concentrate on the social interactions and ways of participating that would lead to such learning.[1] We have taken lessons from both cognitive and socio-cultural accounts to design ways for children in LBD classes to learn to participate in and prepare for participating in collaborative classroom and small-group activities.

Both the cognitive and socio-cultural views of learning tell us that an individual's conception of that repeated sequence of events is necessarily incomplete to begin with, but over time, and with participation and/or observation, especially when that participation is reflected upon or informed by others, the scripts an individual comes to know become fleshed out with more specifics – about, e.g., variations in events and scenes in the sequence and the effects of those variations; the purposes of different events in the sequence; the actors in the script, the roles they play, and the effects of their actions; causal connections between events and scenes – to the extent that the individual can figure out or is informed by others. According to this notion, one could help participants learn a script for a targeted collaborative activity by having them observe and participate in its enactment repeatedly and, to speed the learning process, helping them identify the specifics of events, scenes, sequencing, and roles they might play, variations on those things, and purposes of each.

1.2 Scripts as classroom practices

Another notion of scripts is used in most of the other chapters of this book, and we also refer to that notion when we discuss ways of designing the learning environment to promote script learning as described above. Under this second notion, a script is a designed activity structure or sequence used for an instructive purpose; a *classroom script* is a designed event sequence for the classroom that learners engage in repeatedly; an *instructional script* represents strategies and tactics for sequencing classroom scripts and speci-

[1] Our intention here is not to argue about what exactly is in the head (Lave and Wenger would certainly argue against the full representation of the script residing in an individual's head); rather, our aim is to present a notion of learning about how to participate in commonly repeated activities.

fying teacher roles (and sometimes student roles) so as to promote student learning of the classroom scripts. The hope is that through repeated participation in a classroom script guided by means of enactment suggested in an instructional script, students will internalize expected behaviors and construct cognitive structures (scripts) that will allow them to productively participate in learning activities.

Learning by Design, too, has the equivalent of classroom scripts and instruction scripts in its enactment. In each of the scripted activity structures and sequences (classroom scripts) designed for LBD classrooms, students play certain roles; the teacher plays other roles; software may play other roles, and there is a sequence of events that defines the activity structure. Some, which we've called *rituals* (Kolodner, Camp, et al., 2003; Kolodner et al., 2003; Kolodner & Gray, 2002) and *scripted activity structures* (Kolodner & Gray, 2002) in the past, are quite detailed (see Table 14-2), while some, which we've called *scripted activity sequences* in the past (Kolodner & Gray, 2002) specify the sequencing of activity structures (see Figure 14-1) at a more macro level. Scripted activity structures and sequences provide structure to the classroom and are designed, like other classroom scripts, to afford student learning of the ins and outs of classroom practices that will allow them to participate productively in discourse and other activities.

LBD also has the equivalent of what others have referred to as instructional scripts – the how-to's of making things work in the classroom, though I prefer to call these *instructional strategies* rather than instructional scripts. In particular, Learning by Design promotes a cycle of activities (as seen in Figure 14-1) that sequences classroom scripts with respect to each other in the context of attempting to achieve a design challenge. Each project-based inquiry unit includes several embedded go-throughs of that cycle. For example, designing a vehicle that can navigate several hills on its own begins with activities that help learners understand what the challenge entails and identify some of the science they will have to learn; then, for each science topic, includes at least one go-through of investigating in order to be able to complete some aspect of the design and then several go-throughs of redesign in order to both make the design solution work better and identify and revise science understandings; and then a set of design iterations that bring together what's been learned to perfect the solution to the challenge. The full *Vehicles in Motion* unit includes at least three poster-sessions, three pin-up sessions, and several gallery walks and often includes many more.

Two other instructional strategies are important in LBD. One is adopted from cognitive apprenticeship (Collins, Brown, & Newman, 1989) and specifies teacher roles over time. As each classroom script is being learned, the teacher helps students learn their roles by modeling those roles, coaching them through and providing scaffolding as they do it, and afterwards, guiding the kind of reflection on the activity and articulation of the reasoning that will allow the reasoning to become visible and conscious so as to allow revision over time. The second is the launcher unit, a set of activities designed to introduce learners to the classroom scripts that are so important for their science learning, successful design, and productive discourse. Launcher units (one for each science discipline studied in middle school) are done at the beginning of the school year, and each has a sequence of activities that engages learners in classroom scripts in ways that afford their construction of a cognitive framework representing their understanding of each script.

2. SETTING THE CONTEXT: MORE ON LEARNING BY DESIGN

Learning by Design (LBD; Hmelo, Holton, & Kolodner, 2000; Kolodner, Crismond, Gray, Holbrook, & Puntambekar, 1998; Kolodner, Camp, et al., 2003; Kolodner et al., 2003; Kolodner & Gray, 2002) is a project-based inquiry approach to middle-school science (grades 6 to 8; ages 12 to 14) that focuses on learning science and scientific reasoning in the context of attempting to achieve design challenges. For example, students learn about motion and forces (and about designing and running experiments, justifying with evidence, explaining scientifically, collaborating, and so on) by spending eight weeks iteratively designing, building, and testing a miniature vehicle and its propulsion system. They learn about mechanical advantage by designing and building machines for lifting heavy objects. Each design challenge provides reason for learning some targeted science content, and attempting to achieve the challenge provides a natural and meaningful venue for engaging in both science and design thinking. The need to make one's design ideas work provides opportunities and reasons for students to identify their incomplete and poor conceptions of the science content and to debug those conceptions; the iterative nature of design provides them opportunities to apply and test their new conceptions; and the collaborative nature of design provides learners the need to communicate ideas and results well and opportunities for team work, public practice and presentation of their scientific reasoning.

Figure 14-1 shows LBD's macro level[2]. Activities in the design/redesign cycle (on the left) afford achieving a design challenge, while successful engagement in those activities often requires engaging the investigative cycle (on the right) and its activities. Results of investigations, in turn, provide

[2] Kolodner et al. (2003b) provides the rationale for the different aspects of the design of LBD.

content for application to the design in progress. Individual activities in each cycle are designed to move learners towards successful achievement of a challenge and integrate a variety of science, design, collaboration, and communication practices. Within this framework, students learn the concepts and skills that are needed for success through identifying a need to learn them, carrying out investigations, trying out those conceptions by applying them to the design challenge, questioning their accuracy when the design doesn't work exactly as predicted, and revising.

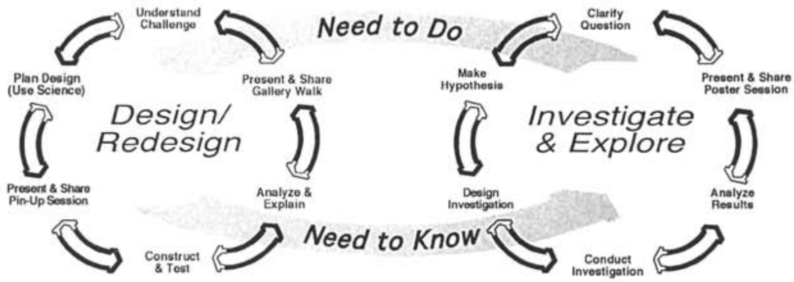


Figure 14-1. Learning by Design's Cycles. From "Promoting Transfer Through Case-Based Reasoning Rituals and Practices in Learning by Design Classrooms," by J. Kolodner, J. Gray and B. Fasse, 2003, *Cognitive Science Quarterly*, 3. Reprinted with permission.

Enactment of LBD's cycles of activities involves participation in a variety of carefully constructed scripted activity structures and sequences (classroom scripts) designed to contextualize important skills with respect to each other and with respect to their usefulness in a project's success. Table 14-1 shows a representative set. These classroom scripts are designed so that they allow success at carrying out the tasks in the cycles in Figure 14-1 at the same time that they provide practice at scientific reasoning and use of newly-learned science concepts.

There are two types of classroom scripts represented in the cycles: action and discourse. Action-based activities, such as *messing about* and *designing an experiment*, are associated with skills and practices of science and design and promote methodological habit and rigor. Students carry out action activities in small groups, dividing up responsibilities for investigations across teams when much needs to be investigated in order to achieve a successful design solution. The focus in action-based scripted activities is on the actions themselves, but as I shall discuss later, these classroom scripts provide context for discourse. Discourse activities have discourse as their major activity, and they sequence who has the floor and specify the content of discussions.

Table 14-1. A selection of LBD's scripted activity structures.

Function(s) in cycle	LBD scripted activity structure	Type and venue	Description
Design investigation	Design an experiment	Action: small group	Given a question to investigate (in the form of discovering the effect of a variable), design an experiment where variables are controlled well, with appropriate number of trails, etc.
Analyze results; analyze and explain, present and share	Creating and refining design rules of thumb	Action, discourse: small group	Identify trends in data and behaviors of devices; connect scientific explanations so as to know when the trends apply (small groups suggest new rules of thumb and the need for changes in existing ones)
Analyze results; analyze and explain, present and share	Creating and refining design rules of thumb	Action, discourse: whole class discussion	Identify trends in data and behaviors of devices; connect scientific explanations so as to know when the trends apply (whole class discusses the suggestions of small groups and chooses new ones and modifies existing ones based on commonalities across small-group experiences)
Present and share (investigate cycle)	Poster session	Discourse, present and share: whole class	Present procedures, results, and analysis of investigations for peer review; followed by rules of thumb
Plan design	Plan design	Action: small group	Choose and integrate design components to achieve the design challenge, basing choices on evidence
Present and share (design / redesign)	Pin-up session	Discourse, present and share: whole class	Present design ideas and design decisions and their justifications for peer review; followed by plan design or by construction and test of design
Construct and test	Test design	Action: small group	Run trials of constructed device, gathering data about behavior, attempt to explain; followed by gallery walk
Present and share (design / redesign)	Gallery walk	Discourse, present and share: whole class	Present design experiences and explain design's behavior for peer review and advice; followed by whiteboarding and rules of thumb

Each discourse activity is inserted into LBD's sequencing at a time when listening to others might help in achieving the project challenge. By sequencing them this way, the need to make a presentation encourages students to reflect on and interpret important aspects of their experiences during action activities, e.g., what they are doing, how successful they are at that, what science content they are using, what they know about that science content, how the science connects to their project goals, how their reasoning connects to their project goals, and so on. The ultimate purpose of this sequencing is to promote the kinds of deliberation that will result in students recognizing and debugging their understanding, skills, and practices. Many discourse activities are done in whole-class configurations; some are done as small groups. Usually, small groups perform actions and make a first pass at reflecting, while whole-group activities provide a venue for presentations from small groups, sharing advice and concerns, struggling together to understand some phenomena, pulling out abstractions and generalizations across what small groups have presented, and discussing the how-to's of next actions.

Each scripted activity structure includes a sequence of events, and each is sequenced with respect to the others. For example, designing an experiment, running an experiment, analyzing results, and presenting them to the class form a scripted sequence of activities, with scripted activity structures associated with experiment design (*design an experiment*), analysis of results (*creating rules of thumb*), presentation of results (*poster session*), and the discussion afterwards (*creating and refining rules of thumb*). Designing an experiment, done in a small group, involves identifying what values to give the variable that is being tested, which variables need to be controlled, how many trials to run, what needs to be measured and how, and variables that might be hard to control, and then generating a procedure. In a poster session, students present their procedures and results to the class and query each other about those results, followed by a full-class discussion of investigative and analysis procedures, implications of what was discovered, and so on.

3. DESIGN OF SCRIPTED ACTIVITY STRUCTURES (CLASSROOM SCRIPTS) AND THEIR SEQUENCING TO PROMOTE COLLABORATION AND DISCOURSE IN LBD

Recall from the discussion about scripts that our notion of script learning is that individuals will learn scripts through observation and participation in commonly-repeated scripted activity sequences (classroom scripts) and that learning can be promoted and sped up by helping them identify the specifics of events, scenes, sequencing, and roles they might play, variations on those things, and purposes of each (instruction strategies). LBD's classroom scripts were designed to be those commonly-repeated activity sequences we wanted students to learn and participate in as scripts. For purposes of discussion about promoting collaboration and learning through scripts, there are three things it is important to notice about the design of LBD's classroom scripts.

1. The classroom scripts that frame discourse were designed specifically to promote the kinds of discourse important to learning science and scientific reasoning.
2. Placement of these classroom scripts in the sequencing matches the design and investigative needs of learners. For example, whole-class discourse activities are inserted into the sequencing at points where there is authentic reason for public discourse – students have had experiences that it is worth sharing with their peers, and they can learn something from their peers' presentations.
3. LBD's iterative approach to achieving design challenges ensures that learners get repeated chances to engage in each scripted activity structure and sequence. During the four to eight weeks working on each project challenge, they have multiple opportunities to refine their understandings, capabilities, and design solutions as they work in small groups and then participate in each kind of public discourse forum.

Our claim is that the sequencing of these sessions and the expectations set about participating in them helps learners engage in interesting discourse as well as learn scripts for doing science and for participating with each other in scientific discussion. In this section, I provide additional detail on the design and sequencing of LBD's activity structures, and in the next section, I move on to how script learning is promoted in LBD.

I focus here on three particular scripted activity structures – LBD's three *public discourse forums* – poster sessions, pin-up sessions, and gallery walks – all designed not only to promote public discourse at times when it is helpful for achieving a project challenge, but also to encourage students to actively reflect on what they've been doing and why they did it the way they did, and to make their thinking transparent. Table 14-2 shows the purposes and sequencing in each.

Table 14-2. LBD's 3 (Scripted) Public Discourse Forums.

	Poster-Session	Pin-up Session	Gallery Walk
Purpose (with respect to achieving the project challenge)	Present and discuss investigative procedures and findings; attempt to draw out trends from the data	Present and discuss alternative solutions to a challenge, along with the strengths and weaknesses of each and what might be expected	Present solution in progress, to what extent it fulfills the challenge, why it might not be working as well as it should, and what might be done to fix it
Purpose (with respect to scientific reasoning and discourse)	Make reasoning and practices associated with designing an investigation and interpreting data visible.	Make reasoning and practices associated with making evidence-based decisions and predictions visible.	Make reasoning and practices associated with testing solutions and explaining scientifically why things behave as they do visible.
When?	After designing and running experiments or other investigations	After trends have been identified from investigations run previously and groups have spent time making sense of what the data, trends, and science they understand implies about achieving the challenge	Solutions have been constructed and tested; they might be complete or need revision
Artifacts/Props used in the presentation	Poster showing research question, procedure, data, trends	Poster showing proposed solution, and for each piece of it, why it was chosen (with references to previous investigations, trends identified across investigations, and science understanding)	Solution artifact

(continued)

	Poster-Session	Pin-up Session	Gallery Walk
Step 1: Presentations: Each group makes its presentation and opens up the floor for questions and advice from peers and teacher			
Group Presentation	Of procedures used, data collected, and trends in data	Of solution ideas, evidence that justifies each	Of solution in progress, what happened when it was tested, explanations of why, ideas about moving forward, areas where the group wants help from the rest of the class
Discussion	Clarification of procedures, appropriateness of procedure for answering posed question, trustworthiness of data, trustworthiness of analysis, …	Pros and cons of each solution idea, why particular evidence is the right evidence to use, validity of evidence, …	Possible explanations for the behavior of a tested solution, shortcomings of those explanations, ideas about moving forward, pros and cons of different ideas
Step 2: Making sense together: Teacher led discussion across presentations, first focusing on the content of what was presented to make visible and debug science conceptions, then focusing on the reasoning and practices of groups to make successful reasoning and practices visible and articulate their how-to's			
Content focus	Extract trends from across data, begin to try to explain those trends, identifying science content that needs to be read, discussed, and/or investigated; beginning of those discussions, including demos, short lectures, reading	Particular data trends and targeted science and what they imply with respect to achieving the project challenge	Using targeted science to explain behavior of solutions in progress; identification of confusions/misconceptions, further discussion of each
Reasoning and practice focus	How-to's of managing variables, getting to trustworthy data, drawing out trends, measurement procedures, …	Using evidence to justify claims and inform decisions	Which explanations are better ones and why; ins and outs of good testing procedures and fair tests

Notice that LBD doesn't simply have a *present and share* activity; it has three such activities. Poster sessions come after carrying out and attempting to explain results of an investigation; pin-up sessions come after planning a design and attempting to justify design decisions; gallery walks come after testing a design and trying to explain its behavior. Each presentation type shares its activity sequence, but in each, the discourse is different. That is, a different kind of presentation is required and different practices and content are targeted in discussions afterward. When presenting experimental results, it is important to report on procedures used and trends in the data; when presenting ideas, it is important to justify them with evidence; when presenting solutions in progress, it is important to report on procedures, what happened, and to explain why things didn't work as planned. By separating out these three kinds of presentations and calling them by different names, LBD calls attention to the fact that each requires different discourse. Discussions after presentations of the three types are quite different from each other. Separating present and share activities into three different discourse structures with different expectations about the content of that discourse has been particularly useful in helping students and teachers focus on scientific reasoning and discourse appropriate to what they are doing at the time of a presentation (Kolodner, Camp, et al., 2003).

As discussed, each scripted public discourse activity structure in LBD is placed in the sequencing of class activities at points where there is authentic reason for engaging in it – students have had small-group experiences that it is worth sharing with their peers, and/or they have a need to learn something from the presentations or discussions. These scripted discourse activities in LBD were designed to provide a public venue for participating in scientific reasoning, in this way promoting repeated deliberative practice needed for deep learning of science content and scientific reasoning (Kolodner, Camp, et al., 2003; Kolodner et al., 2003; Kolodner & Gray, 2002). But their placement also appears to play two other essential roles in promoting productive collaborative discourse. First, students prepare for discourse activities in their small groups, reflecting on their activities in ways that allow them to present to the class. The need to present certain specifics about their work and their reasoning causes them to have discussions about that work and reasoning. We see that in Example 3 at the beginning of this chapter. That is, while there is no script for participating in this preparation, the script for participating in a pin-up session, poster session, or other whole-class discourse activity provides focus for discussion during preparation. Second, the movement from small-group to whole-class and back again promotes observing the discourse of others and participating in discourse in multiple ways. During presentation in discourse forums, the reflection students have attempted to do as a small group can be discussed and scaffolded and taken to the next level through interaction with teacher and peers. Sequencing tends to move students from small social configurations to big ones and back again, so that groups can learn from each other and bring each other up to pace.

4. PROMOTING SCRIPT LEARNING IN LBD – INSTRUCTIONAL STRATEGIES

There are two parts to LBD's instructional strategy:

1. Its scripted activity structures and their sequencing (classroom scripts) are enacted over and over again in the context of new situations, within single units and across units, their enactments are scaffolded, and their enactments include reflection on how to participate in them productively. This repeated deliberative scaffolded practice implements a kind of cognitive apprenticeship (Collins et al., 1989).
2. Its launcher units, enacted at the beginning of the school year, introduce each of the important scripted activity structures – individually and in the context of its sequencing with other classroom scripts and in full design challenges (Holbrook & Kolodner, 2000).

4.1 Repeated deliberative scaffolded practice of scripted activity structures (classroom scripts)

Repeated practice works in two ways – within and across projects (curriculum units). Each designed activity structure and sequence is repeated several times in the context of each project, providing opportunities for participating in each close enough in time to other enactments that previous opportunities are remembered. This way, students not only experience small variations in sequencing but also remember enough about previous enactments to be able to draw parallels between enactments. The same activity structures and sequences are then used across projects, providing ongoing opportunities for repetition and for experiencing broader variation. A student might participate in three different poster sessions during the Vehicles unit – one focusing on what effects the distance a vehicle will travel on its own and culminating in a discussion of friction and gravity and how forces interact with each other; another focusing on what effects the force produced by a balloon engine, culminating in discussion of propulsion force and continuous and one-shot forces; and another focusing on what effects the functioning of a rubber-band engine, culminating in additional discussion about forces in pairs. In the next unit, focusing on mechanical advantage, a poster session will focus on the effects of increased mass on different kinds of simple machines, culminating in discussion of the relationship between force and mass in creating mechanical advantage. And so on. In each, they use what they already know about managing variables and obtaining trustworthy results to help each other continue to be able to design experiments and obtain trustworthy results as the relationships between variables become more complex.

Deliberation, i.e., thinking about their reasoning in a way that will allow them to learn how to reason better, is achieved in LBD classrooms by tactics used before and after enactment of each scripted activity structure. When an activity structure is introduced to students, they read text about the sequencing and purpose of the activity and how best to participate. They again read text about sequencing and participation when new variations are introduced. Then, after each experience with each activity structure, the teacher initiates reflective discussion about it, encouraging students to articulate the sequencing, the roles of each part of the sequencing, how they participated, what they gained from it, how it built on activities that came before, how it prepares them for activities to come, and so on. Additionally, as is typical of cognitive apprenticeship (Collins et al., 1989), the teacher has a role of modeling what is expected of students during early enactments and then later when more is expected of students in their enactments.

Figure 14-2. A Design Diary page: "My Experiment." Notice that it prompts learners for some of the important issues they need to discuss and/or plan for (See Puntambekar & Kolodner, 1998, 2005, for more detail)

Scaffolding during small-group work in LBD takes the form of design diary pages (Kolodner et al., 2003; Puntambekar & Kolodner, 2005) and/or software prompting (Kolodner et al., 2004) made available during each important activity. A design diary page is a kind of worksheet with prompts about what to focus on while doing the activity and questions to answer while engaging (see, e.g., Figure 14-2). Software pages provide more detailed prompting and hinting and sometimes templates that suggest what needs to be done and/or discussed. Figure 14-3 shows filled-in templates for the *plan* and *procedure* parts of the software-based design diary page (implemented as part of SMILE (Kolodner et al., 2004)). The posters students make for use during poster and pin-up sessions serve as scaffolding during whole-class activities – they remind them of what they need to be presenting and then discussing.

Plan

Describe your plans for investigating your problem.

Plan Summary

What variable will you change?	Length of whirly gig wing
What values will you give it?	Original (template) Original + 1 inch Original + 2 inches
What conditions (variables) will you control?	Length of stem; Number of paperclips
How many trials will you run?	5
What will you measure and how?	The time the whirly gig is in the air

Procedure

Include step-by-step instructions so another team could run the same procedure. Be sure to include steps for measuring and recording data in your procedure.

Step-by-step Procedure

Procedure step description	Thing(s) to be careful about	How you will be careful
Create 3 whirly gigs with different wing lengths	Keep the wing width and stem length constant	Use the template and only change the wing length
Have one person stand on a chair and drop a whirly gig.	Drop each whirly gig from the same height each time.	The same person should drop the whirly gig each time.
Have another person use a stopwatch to time how long the whirly gig is in the air.	Make sure the time is accurate.	Start the stopwatch as soon as the dropper lets go; stop the watch as soon as the whirly gig hits the ground.
For each whirly gig, repeat steps 1-2 five times.		
Find the average of the 5 trials for each whirly gig to see which one was the slowest.	Getting the average	Use a calculator

Figure 14-3. Additional prompting for "My Plan" and "Step-by-Step Procedure" in the software templates for "My Experiment" found in SMILE (See Kolodner et al., 2004, for more detail)

4.2 Launcher Units for introducing scripted activity structures

Experiences in the classroom show that, in addition to what cognitive apprenticeship suggests about repeated deliberative practice, learners needed help getting started with each practice (Kolodner et al., 2003; Holbrook & Kolodner, 2000). Asking students to learn the scripted sequences of activities and the purposes of each step in their sequencing at the same time they were learning new and difficult science concepts, was difficult for students to achieve and difficult for teachers to facilitate (Kolodner, Camp, et al., 2003). At the request of teachers the LBD research group worked with, the research group created introductory *launcher units* (Holbrook & Kolodner, 2000; Kolodner et al., 2003) with four weeks of activities in them that introduce students to scripted activity and discourse structures and sequences (classroom scripts) in the context of a series of science activities that require only simple content. Launcher units are intended to help students quickly learn basic scripts for each activity and discourse structure that include sequencing and purposes.

For example, in their first activity in the year, students attempt a simple design challenge in small groups and show their results to the class in a gallery walk (after reading about what a gallery walk is). They get to work and create something very quickly, usually with little deliberation about what the options are. During the gallery walk, they notice that not everyone had the same understanding of the challenge, providing the teacher with an opportunity to point out what it means to understand a challenge and its importance before going off and trying to achieve it. They continue by defining the challenge better, this time in terms of criteria (goals to be achieved) and constraints (limitations and availability of resources, time, and so on). Then, the teacher asks them to attempt it again. As they engage in planning their designs this time (*plan design*), each group deliberates about how well they are achieving the criteria and keeping within constraints, and when they do their next gallery walk, each group presents not only its solution but why they think it is a good solution. This second time through, however, students notice that they've copied from each other. They also notice how much better their designs are as a result of considering the criteria more deliberately, considering the goodness of options they considered, and integrating in the ideas of others. The teacher helps them recognize how much they learned from each other but that fairness requires giving each other credit. At the conclusion of this second gallery walk, they discuss what they've learned about planning a design (specify and consider criteria and constraints) and how to participate well in a gallery walk, articulating how important it is to give credit to others for their work. A major addition most students make to their personal scripts for gallery walks (which can be observed next time they participate) is that while one is showing off a solution, one must discuss what work that solution builds on and who was responsible for that work.

Students engage in similar mini-challenges to introduce them to the need to control variables, measure accurately, run procedures in a consistent way, and so on. They also watch a movie where they have a chance to observe scientists, engineers, or designers engage in these same kinds of activities – collaborating, investigating an unknown, making a well-formed scientific argument, designing an experiment, and so on. Other activities go into more depth in different parts of the design and investigation cycles, introducing the range of scripted activity structures (e.g., pin-up sessions, poster sessions, designing an experiment) and the scientific reasoning and practices they include (interpretation of data, scientific explanation, and so on). As their last activity in a launcher unit, students tackle a relatively simple but full design challenge, in which they participate in the full LBD cycle (as in Figure 14-1). In the physical science launcher, they spend 8 days designing parachutes and learning a bit about combining forces; in the earth science launcher, they spend 3 weeks designing a way to manage erosion in a designated area and modeling their solutions in a stream table, learning about earth's surface processes and interactions between people and the environment.

By the time they've finished a launcher unit (4 - 6 weeks), students have engaged several times in each of LBD's classroom scripts, and they've had one full run-through of the LBD cycle. Students come away from these launcher activities with the want to collaborate, basic ability to participate in each classroom script, and an appreciation of many of the practices scientists engage in (Gray, Camp, Holbrook, & Kolodner, 2001; Kolodner et al., 2003).

Each additional unit, lasting 3 to 10 weeks, makes at least one run through the entire LBD cycle and several runs through each of the classroom scripts. Discussions before small-group activities remind students of the ways they've carried out these activities previously, and discussions after each of the discourse activities focus on both content that is being learned and their added sophistication in carrying out the scientific reasoning and science practices that they've worked on in their small groups. There is often reference back to experiences during the launcher unit, as these are the activities that classroom scripts were originally learned from.

5. DISCOURSE, COLLABORATION, AND LEARNING

The examples at the beginning of this chapter show that at least some students in LBD classrooms are able to participate productively in LBD's classroom scripts and the collaborative discourse that goes with them. A variety of evidence of development of student discourse capabilities has been collected, most reported in research articles. Three very robust findings with respect to scientific discourse fall out of analyses:

1. Using performance assessments and comparing learners in LBD classes and learners in matched inquiry science classes, we find that LBD students consistently participate more and with better quality than non-LBD students in science practices and discourse. This comparison holds as early as right after the launcher unit and continues throughout the school year after students engage in additional LBD units (Kolodner et al., 2003; Gray et al., 2001).

2. When comparing the performance of LBD students early in the school year and later in the school year on these same performance assessments, their participation in discourse increases both in quantity and quality. The more LBD units they've engaged in, the more their discourse capabilities increase (Kolodner, Camp et al., 2003).
3. The more attention teachers focus on whole-class discussions at the end of discourse forums, the more participation in discourse increases over the school year. (Ryan, 2003; Ryan & Kolodner, 2004)

The first two results are consistent in data collected over a 5-year period – across matched classes, and with average ability, honors, rural, suburban, urban, low-income, and high-income populations. The third is based on less data but is no less significant. All of these results come from performance assessments in which students are asked to work in small groups, first to design an investigation, then to carry out an investigation, and then to interpret results and apply them. The science content in performance assessments is purposely at a basic level so that all participants have adequate science understanding to be able to participate. Data are coded for degree of collaboration, remindings of previous experiences, and quantity and quality of scientific discourse. Results pertaining to scientific discourse come from three coding categories: science content talk, science practice talk, and self-checking of science practice. Performance assessments are done in LBD classes after each unit, and those same performance assessments are done in classrooms matched to each LBD class later the same week or the week after. Content in performance assessments has always been covered in non-LBD classes.

The discussion below, for example (extracted from Kolodner et al., 2003), happened midway through the school year as students were deriving a procedure for measuring the speed of a battery-operated toy car. While the discourse certainly isn't fluent, it's quite good for middle-school students (grade eight; age 14); all of the students in the group participated, and they talked about science content (what is speed) and several science practice issues – what is expected of them, how to measure, what to measure, how to collect data, and what it would take to get trustworthy results ("three times and it will come about the same. ... it's got batteries.").

B2- We must have the measurements of the distance the car travels and the time that it took to travel. The average speed of the car. What if the car is not stopping? The car keeps going.

B1- You don't have to put how far it went. Just put the speed of it. Just put how fast it goes.

B2- I know but I am trying to figure out speed.

B1- Distance divided by time.

B2- We must have a measuring device.

B2- That wouldn't be the average speed.
B3- We just have to do it a couple of times.
B2- Just a couple of times and then divide by that number of times.
B1- You can find the speed of an object.
B3- It says average speed, John (name changed).
B1- Okay, just turn it on and let it go for about five seconds and then you'll get the same thing about every time.
B3- How about we make it start here[indicates one side of the table] and end it here [points to the other end of the table].
B1- You can just write. Do the test about two or three times and it will come about the same. Cause it is not like your balloon cars, it's got batteries.

In general, discourse among eighth graders doesn't look a lot like discourse among mature and knowledgeable grownups. The transcripts that go with the three examples at the beginning of this chapter have the same qualities as the discourse shown above – it is not always possible to figure out what the children mean, they interweave several conversations with each other, they intersperse real science discourse with discussions about hair, who has better handwriting, romantic interests, and the like. They don't always get to the depth we would like. And their discussions are often quite disconnected. For example, the discussion from the first example about straw length effecting mass looks like this:

JG: Don't you think when you're cutting the straw it's changing the mass?
AB: Uh, yeah.
MY: MV: (at same time) Yeah, that's the straw length.
KD: Yeah but that's the...
KD, KK: (at about same time) Straw length.
KD: You can't make it shorter, and not cut it, without changing the mass.
...
?: (at same time as KD) You could tape it.
CF: (at same time as KD) Yeah, but...
JG: (at same time as CF) You but it, you can just put the 10 cm through it in the, that part.
AB: (at same time as JG) But that's still not, that's not. something they tested.
...
KD: (talking over JG) (gesturing) Wouldn't it change the the mass of yours too?
JG: Yeah all the different air...
KD: All of the masses would change, in yours.

AB: Yeah, our mass changed too.

On the other hand, it is notoriously difficult to get eighth graders to participate in these kinds of discussions. LBD students participate far better than their matched counterparts in science discourse, and, in general, in scientific activity, and their participation gets better over time with additional practice. LBD students come to know the scripts for the classroom, and they come into the classroom ready to participate. During performance assessments, they move directly into the activities they are asked to do with little or no time needed to figure out what's expected (Kolodner et al., 2003).

6. CONCLUDING THOUGHTS

Many of the chapters in this book are about providing scripts to collaborators to help them carry on a conversation, and almost all are about on-line collaboration and instructional scripts that can be used to promote productive discourse on-line – all quite different from LBD's model. Nonetheless, the analysis of LBD, I think, can contribute several things to discussion about the potential role of scripts in promoting good collaborative discourse. First, LBD contributes the notion (a reminder, really) that while one can design sequences of events or activities to be used as scripts, it is important to remember that their use will depend on how well they are learned as scripts. Both the cognitive and socio-cultural literatures emphasize this. Such learning requires a combination of observation, participation, repetition, identification of the sequencing, understanding of the purposes of steps in the sequencing, and experience with and understanding of variations in the classroom scripts that promote discourse. Such learning may not happen quickly, and learners need a variety of opportunities for practicing to learn the basics of scripts and the variety of ways of engaging in each.

Second and related, it seems that learning how to participate in discourse is like learning how to participate in other cognitive activities and practices. Because the reasoning involved in participating is invisible, and because learning to participate requires reflective practice, there's a need for facilitating the learning of discourse practices that will be used in collaboration in the same way the learning of other reasoning activities and practices are facilitated. As such, as Collins et al. (1989) propose with respect to promoting learning of cognitive skills, promoting learning of discourse skills and practices might require such things as modeling behavior for observation, coaching, prompting, scaffolding, facilitating reflection and articulation of the steps, and so on. That is, participating in classroom scripts that promote certain kinds of discourse isn't enough. Classroom scripts need to match discourse needs, and instructional strategies need to include ways of introducing each classroom script and of promoting reflection on and articulation of their sequencing, purposes, variations, and so forth (making the invisible visible).

Third, effective collaboration is always about something. Our mode of encouraging good collaborative discourse in LBD has been to help learners learn how to engage proficiently in targeted reasoning. This gives collaborators tools they need for doing and critiquing reasoning together. LBD's activities don't directly teach ins and outs of collaborative discourse; rather, they teach children to reason scientifically, giving them reason to hold each other to high standards of scientific reasoning. This is probably not enough for all collaborative discourse, but it can go a long way.

How does this advice translate into design of computer-supported scripts for interaction? Computer-supported scripts, as any other kind of script, need to be designed to promote students taking on appropriate goals, to match perceived needs of learners, and in a complete system that takes into account what scripts need to be learned by learners, the kinds of scripted activities that would afford such learning, and the instructional strategies to promote targeted learning. Each designed classroom script needs to be enacted repeatedly and deliberatively and with modeling, coaching, and/or scaffolding in contexts of authentic need and use; each script needs to be introduced as a way of achieving perceived goals and deliberated over repeatedly so that learners will develop more embellished cognitive structures over time and with practice; and learners need scaffolded experience with the major variations of each script. Most importantly, I believe, is that the system of classroom scripts and instructional strategies needs to include opportunities for small-group as well as public (whole-class/large group) practice. Similar to what Vygotsky (1978) claims about learning in individuals, LBD's results show that small groups and individuals get better at collaborative discourse to the extent that they get to participate in, observe, reflect on, and identify the ins and outs of similar collaborative discourse that happens in public forums.

REFERENCES

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing and Mathematics. In L. B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser* (pp. 453-494). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Gray, J. T., Camp, P. J., Holbrook, J., & Kolodner, J. L. (2001). *Science talk as a way to assess student transfer and learning: Implications for formative assessment.* Meeting of the American Educational Research Association, Seattle, WA. Retrieved from: http://www.cc.gatech.edu/projects/lbd/pdfs/aerasciencetalk.pdf

Hmelo, C. E., Holton, D. L., & Kolodner, J. L. (2000). Designing to Learn about Complex Systems. *Journal of the Learning Sciences*, *9*(3), 247-298.

Holbrook, J., & Kolodner, J. L. (2000). Scaffolding the Development of an Inquiry-Based (Science) Classroom. In *Proceedings, International Conference of the Learning Sciences 2000 (ICLS)* (pp. 221-227). Mahwah, NJ: Lawrence Erlbaum Associates.

Kolodner, J. L., Camp, P. J., Crismond D., Fasse, B., Gray, J., Holbrook, J., et al. (2003). Problem-Based Learning Meets Case-Based Reasoning in the Middle-School Science Classroom: Putting Learning by Design™ into Practice. *Journal of the Learning Sciences, 12*(4), 495-548.

Kolodner, J. L., Crismond, D., Gray, J., Holbrook, J., & Puntambekar, S. (1998). Learning by Design from Theory to Practice. In *Proceedings of ICLS 98* (pp. 16-22). Charlottesville, VA: AACE.

Kolodner, J. L., & Gray, J. (2002). Understanding the affordances of ritualized activity for project-based classrooms. In *Proceedings of the International Society of the Learning Sciences 2002 (ICLS)* (pp. 222-228). Seattle, WA: Lawrence Erlbaum Associates.

Kolodner, J. L., Gray, J., & Fasse, B. B. (2003). Promoting Transfer through Case-Based Reasoning: Rituals and Practices in Learning by Design Classrooms. *Cognitive Science Quarterly*, 3(2), 183-232.

Kolodner, J. L., Owensby, J. N., & Guzdial, M. (2004). Case-Based Learning Aids. In Jonassen, D. H. (Ed.), *Handbook of Research for Educational Communications and Technology* (2^{nd} Ed., pp. 829-861). Mahwah, NJ: Lawrence Erlbaum Associates.

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate peripheral participation*. New York: Cambridge University Press.

Puntambekar, S., & Kolodner, J. L. (1998). The Design Diary: Development of a Tool to Support Students Learning Science by Design. In *Proceedings of the International Conference of the Learning Sciences '98 (ICLS)* (pp. 230-236).

Puntambekar, S., & Kolodner, J. L. (2005). Toward implementing distributed scaffolding: Helping students learn science from design. *Journal of Research on Science Teaching, 42*(2), 185-217.

Ryan, M. T. (2003). *Using a new design rules practice and science talk development to enhance conceptual understanding, scientific reasoning, and transfer in Learning by Design ™ classrooms*. Unpublished Thesis. University of Kansas, Lawrence.

Ryan, M. T., & Kolodner, J. L. (2004). Using 'Rules of Thumb' Practices to Enhance Conceptual Understanding and Scientific Reasoning in Project-based Inquiry Classrooms. In *Proceedings of the International Conference of the Learning Sciences 2004 (ICLS)* (pp. 449-456). Mahwah, NJ: Lawrence Erlbaum Associates.

Schank, R. C. (1982). *Dynamic Memory*. New York: Cambridge University Press.

Schank, R. C. (1999). *Dynamic Memory Revisited*. New York: Cambridge University Press.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Chapter 15

EDUCATIONAL PERSPECTIVES ON SCRIPTING CSCL

Päivi Häkkinen[1] and Kati Mäkitalo-Siegl[2]
[1]University of Jyväskylä, Finland; [2]Knowledge Media Research Center, Tübingen

Abstract: This chapter discusses different educational approaches to collaboration scripts. When carefully designed, scripts can push learners to that kind of situations in which meaningful interaction can take place. However, many conditions need to be met for this to happen in authentic classroom contexts. One of the biggest educational challenges in instructional design of computer-supported collaboration scripts is to better integrate them into wider social planes such as overall classroom activities. Scripts could also be considered as contextual and situated resources in collaborative learning environments. Furthermore, a challenge for future research is to explore how external scripts can be gradually replaced by individual self-regulation. In order to face many of these challenges, longer-term follow-up studies should be conducted in research on collaboration scripts.

1. INTRODUCTION

The use of online learning environments has increased in different educational settings. The problem has been that simply offering online learning environments for student use does not guarantee that they will interact in a way that promotes learning. Also, teachers in the field need pedagogical guidance to use new learning environments. For example, the research done recently in four Scandinavian countries reveals that of the two thirds of the teachers who have received ICT training, only one third of them felt qualified to use ICT in their teaching (E-Learning Nordic, 2006).

At the same time, increasing interest in research on collaborative learning, particularly in computer-supported settings, has provided knowledge that can guide and support student interaction and collaboration. Through scripting, learners would convey an introduction to the activities that they would not otherwise engage in on their own. Scripts have proved to be a valuable approach to facilitate specific forms of interaction and collaborative activities in online learning environments, which can promote different kinds of learning objectives without compromising the idea of self-guided learning (e.g., Dillenbourg, 2002; Weinberger, Ertl, Fischer, & Mandl, 2005).

The basis of the research on collaboration scripts, as also represented in this book, is the integration of different sciences - cognitive psychology, computer science and educational science - which makes the theoretical background stronger than what would be represented by only one discipline (Fischer, Wecker, Schrader, Gerjets, & Hesse, 2005). The role of educational science is to offer practical insights into exploring the use of scripts in real-life educational settings. In addition, there are pedagogical challenges we will face when implementing scripts into practical educational settings. The articles in this section deal with the design principles and effects of collaboration scripts. Further, they raise several questions related to pedagogical challenges, as well as to methodological questions of studying scripted collaboration.

2. DIFFERENT NOTIONS OF SCRIPTING

Collaboration scripts comprise a number of rules, which describe the way in which learners should interact with each other and collaborate on a task (O'Donnell & Dansereau, 1992). Specifying learners' collaboration processes through scripts is intended to help learners to enter into activities that serve productive interaction and collaborative knowledge construction. Scripts are meant to assign actions in such a way that all learners will carry out in turn the action specified or perform a predefined series of specified actions (Weinberger, 2003). Scripts also provide collaborative learners with a complex set of instructions detailing several goal dimensions, for example, supporting meta-cognitive and elaborative activities or fostering epistemic activities or social processes in particular. Subsequently, scripts aim to enhance the probability of productive interactions.

Recent research on collaboration scripts has made a distinction between macro- and micro-scripts (Dillenbourg & Jermann, this volume; Kobbe, Weinberger, Dillenbourg, Harrer, Hämäläinen, & Fischer, 2006). Micro-scripts lean more toward a psychological, process-oriented perspective, whereas macro-scripts are based on an educational perspective that influences the process more indirectly. According to Dillenbourg and Jermann (this volume), a micro-script scaffolds the interaction process per se by providing sentence starters, question prompts or descriptions. A macro-script, on the other hand, sets up conditions in which favourable activities and productive interaction should occur. Macro-scripting targets to push learners to engage in those kinds of activities that promote interaction, but no specific support, for example, on how learners should interact is given. Compared to micro-scripts, macro-scripts also typically describe longer time segments and are spread over more social planes, emphasizing the orchestration of activities within the classroom. In this section of the book, the scripts presented in the articles by Weinberger, Stegmann, Fischer, and Mandl, as well as by Ertl, Kopp, and Mandl, more or less represent micro-scripting. The script presented in the article by Kolodner, on the other hand, represents macro-scripting.

Kolodner's study was conducted in face-to-face situations as a long-term study in authentic classrooms. Her article describes how scripts can help to integrate aspects of interaction and make them suitable for use in educational settings. According to Kolodner, it is important that learners garner various experiences when participating in the kinds of practices and activities, such as observing, repeating and reflecting, which enhance their membership and active learner role in the scientific community. It is not enough to just promote certain kinds of discourse. Learners also need to have a good reason for discourse, and classroom scripting can be matched with different discourse needs (see also ArgueGraph; Dillenbourg & Jermann, this volume). The Learning by Design model presented by Kolodner is in many ways different from the scripts presented in the other two chapters in this section. The LBD model represents macro-scripting involving some micro-scripting (prompts, coaching). This macro-scripting provides the reasons for learners to participate in productive discourse with their fellow learners in the LBD model. Although reporting better participation in scientific activity in the Kolodner's study, the learning outcomes were not reported. This will raise a question as to whether different learners learned the content better than without the LBD model.

In the study by Weinberger and his colleagues, the university students solved three cases using the attribution theory in the online learning environment. The results show that when supporting students' social interaction, their interaction was not only more productive and meaningful, but also their epistemic activities were enhanced. It might be interesting to see analyses on how the students proceed from one case to another, whether they develop their own scripts regardless of the scripts which were given, and whether these groups get better when solving the second and the third case. Also, repeating the same activities might provide a clue as to whether the learners adopted these scripts or created scripts suitable for the particular group as in Kolodner's studies. Planning a long-term study, where the same students solve different tasks, might help to determine whether these scripts will be adopted and whether students can transfer these scripts into different situations. This way we could also explore how students use the scripts as situated resources of the learning context (see also Stahl, this volume).

In the study by Ertl, Kopp, and Mandl, the content schemes were effective in both of their studies. In their first study, the scheme was effective in terms of collaborative learning outcomes, whereas, in the second study, it has a positive effect on both collaborative and individual learning outcomes. One of the basic ideas behind collaborative learning is that groups should perform better and produce something greater than individuals alone could perform or produce. This study has been able to find a way to support groups to collaborate in such a way that they are successful as a group. Further questions in this line of research might be: Are collaboration scripts as useful in videoconferencing as in face-to-face situations without training and without reflection? This particular study was a short-term study, whereas repeating and using the same guidelines several times might give different results and might help learners to internalize scripts.

To sum up all the three studies in this section, it seems evident that with regard to collaborative learning, it is important to support not only the content, but also the social level of interaction (see Barron, 2003). It seems that learners face a dual-problem space of this kind as they are supposed to work and learn collaboratively. Crucial problems concerning interaction in different educational settings can emerge in the relational space, that is, at the social and emotional levels of collaboration. In Kolodner's LBD cycles, there are two types of classroom scripts represented: action and discourse. Action-based activities (e.g., designing an experiment) are associated with skills and practices of science and design, they happen in small groups, and they provide context for discourse. Discourse activities, on the other hand, have discourse as a major activity, and they sequence and specify who has the floor and what the content of discussions is. Weinberger and colleagues differentiate between three process dimensions: epistemic dimensions, referring to arguments as steps towards solving the learning tasks, an argument dimension, referring to formal criteria for solving the learning task, and a dimension of social modes of co-construction, referring to how learners interact with each other. Ertl and colleagues talk about content schemes as content-specific support and task-specific support as collaboration scripts.

Altogether, there is growing evidence that learning in collaborative environments cannot be explained as constituting only the result of specific abilities, but appears as the product of complex and dynamic interactions between cognitive, social, affective and motivational variables (Pintrich, Marx, & Boyle, 1993). What is needed now is to better understand how individuals' mental processes relate to social and situational factors that influence cognitive performance and learning. Furthermore, the activities involved in collaborative learning are much more complicated than what outcomes alone reveal (see also Dochy, 2005).

3. METHODOLOGICAL CHALLENGES

The studies in this section also raised several methodological questions and challenges. While seeking methodological accounts for capturing, e.g. the processes of collaborative learning or scripted collaboration, we should bear in mind that the analysis of collaborative interaction cannot be isolated from the context in which it is embedded (Crook, 2000). To find out more about the nature of collaborative learning processes and what promotes collaborative knowledge building, different features affecting learning must be studied in the context of the joint activity, i.e. with relation to and in the form they occur in different learning environments. Furthermore, it is also important to develop methods for identifying how scripts are used as situated or contextual resources as suggested earlier. Consequently, new methods are needed to capture the process of collaborative interaction and its contribution to learning.

Altogether this requires longer term follow-up studies with the same groups, not just analysis of short episodes of interaction. We should move towards micro-level analysis of interaction in the study of scripted collaboration. The long-term follow-up studies could enlighten, for example, whether scripts can be faded out in order to see if learners adjust their techniques when there are no longer scripts to guide them. However, this demands that group processes are followed in running time in order to trace these problems learners are facing.

4. DESIGN ISSUES AND PEDAGOGICAL CHALLENGES

One of the crucial questions from the perspective of educational research is the impact of basic research. How can the research on collaboration scripts inform us in developing pedagogical practice? And what are the biggest challenges in designing and implementing scripted collaboration, such as presented by the authors in this book, for authentic learning environments?

A notably challenging task is to transfer the implications of research projects out into the field. One of the challenges therein is to modify and revise the existing practices to form a new culture of schooling. Stahl (2005) puts forth an interesting idea about a theoretical confusion between learning and group knowledge. This can be seen as a barrier to both educational practice and educational research. Learners, teachers and researchers have a tendency to see learning as an individual attribute, failing to grasp the true potential of collaborative learning, as they lack awareness that groups can construct knowledge together in a way that is impossible for single learners and that group learning can subsequently enhance individual learning (Stahl, 2005).

Teachers need instructional strategies when introducing classroom scripts, as well as promoting reflection and articulation of their sequencing, purposes, variations, etc. In some of the studies, students repeat the problem solving task three times (for example, Weinberger et al., this volume), but they do not reflect on what they were doing and why and how. Therefore, they might have problems in transferring the scripts into different situations. Kolodner described in her article that when repeating and afterwards reflecting on these activities and practices, collaborators became more involved in participating in the activities and discourse. The quality of participation and discourse also increased. According to Kolodner, classroom scripts and instructional strategies need to include opportunities for small-group as well as public (whole-class/large group) practice.

From the instructional design perspective, scripts should allow flexible mobility between different social planes (individual – group – classroom). According to Dillenbourg (2002), the effectiveness of scripts is based on the idea of integrating usually separate activities: individual, cooperative, collaborative and collective activities. Furthermore, scripts enable the integration of co-present activities and computer-mediated activities. They also introduce a time frame in distance education where students often lack landmarks for their time management. The other side of the coin in designing well-defined scripts is the risk of over-scripting collaboration. Predefined scripts can disturb the richness of natural interaction and problem solving processes. Furthermore, this kind of "educational engineering" approach can lead to striving for effectiveness at the cost of the genuine notion of collaborative learning. The balance between the benefits and risks of structuring collaboration depends on the core mechanism that the script is based on, in other words, how the designer or teacher aims to foster productive interactions and learning.

What has not been studied much yet is how teachers adopt the use of scripts or how their own role and conceptions of learning fit with the ones represented by scripting. Can the use of scripts even create a conflict in a classroom regarding the role of teacher? This definitely depends on whom the scripts are designed by. Therefore, one of the crucial questions is how to facilitate the teachers' design and use of collaboration scripts. One option suggested by European Research Team CoSSICLE ("Computer-Supported Scripting of Interaction in Collaborative Learning Environments"; funded by the Kaleidoscope Network of Excellence) is to develop tools to help in designing collaboration scripts. Compared to any authoring tool, the idea behind these tools should be to make pedagogical rationale behind the scripts explicit.

5. CONCLUSIONS

It is evident, also on the basis of the studies presented in this book, that scripted collaboration does not happen without problems and challenges. Different groups will act differently regardless of the same instructional interventions and environments. Therefore, the question arises whether scripts allow enough freedom for the group members to choose the best way for them as a group to collaborate and learn together. In the design of collaboration scripts, we often refer to so-called "ideal scripts", whereas *actual* scripts are the ones that tell us what really happens and emerges in interaction situations. Furthermore, we should consider what other scripts may already be operating in the learners' mind(s) or in the learning environment (internal scripts & external scripts; see Kollar, Fischer, & Slotta, 2005). Learners have learned particular interaction patterns in everyday situations or in educational situations, which they try to transfer to, e.g. collaborative learning situations. However, learners might be unfamiliar with collaborative learning situations, and therefore, may fail to use skills and knowledge, which they already possess, in daily situations that enable them to collaborate.

Weinberger and colleagues raise an important challenge for the future research of CSCL environments. Namely, we should focus on how scripts can be designed not to substitute, but to facilitate discourse and cognitive activities related to individual knowledge acquisition. There is also a need to investigate the interaction of different script components that may be adapted to the already existing internal scripts. Internal scripts can play a crucial role, i.e., when using scripts for videoconferencing like in the study by Ertl and colleagues. Using the same guideline several times in videoconferencing scenarios may encourage learners to internalize it as a script. And, after a time, this internalized script may be able to support learners as suggested by Schank and Abelson (1977). Also, Kolodner emphasizes that while one can design sequences of events or activities to be used as scripts, it is important to remember that their use will depend on how well they are learned as scripts. Therefore, one of the most crucial questions here is how external scripts can gradually be replaced by individual self-regulation.

Based on the three different studies highlighted in this section, certain kinds of scripts do enhance learner interaction in a meaningful and productive way with regard to collaborative learning. However, learners will not interact with each other in a productive way, if there is not a good reason for it. Therefore, an authentic need for collaboration as suggested by Kolodner is required. For example, in the LBD model, effective collaboration is always about something, namely, LBD aims to encourage the learning of the reasoning that needs to be done collaboratively. With scripts, learners can be guided to the kind of situation where learners need to interact with each other and support each other to interact meaningfully.

REFERENCES

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

Crook, C. (2000). Motivation and the Ecology of Collaborative Learning. In R. Joiner, K. Littleton, D. Faulkner, & D. Miell (Eds.) *Rethinking Collaborative Learning* (pp. 161-178). Free Association Books: London.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL: Can we support CSCL?* (pp. 61-91). Heerlen: Open University of the Netherlands.

Dochy, P. (2005, August). 'Learning lasting for life' and 'assessment': How far did we progress? Residential address at the 20th anniversary of the European Association for Research on Learning and Instruction, at the 11th Conference of the European Association for Research on Learning and Instruction (EARLI), Nicosia, Cyprus. Retrieved September 14, 2005, from http://www.earli.org/about/articles/EARLI2005%20presidential%20 address%20FINAL.pdf.

E-Learning Nordic 2006. (2006). Copenhagen: Ramboll Management.

Fischer, F., Wecker, C., Schrader, J., Gerjets, P. H., & Hesse, F. W. (2005, August). *Use-inspired basic research on the orchestration of computer-supported processes of learning and instruction*. Presentation at the SIG Invited Symposium "Instructional design and empirical research: experiments and/or design experiments" the European Association for Research on Learning and Instruction (EARLI) conference, Nicosia, Cyprus.

Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., & Fischer, F. (submitted). Specifying Computer-Supported Collaboration Scripts.

Kollar, I., Fischer, F., & Slotta, J. D. (2005). Internal and external collaboration scripts in webbased science learning at schools. In T. Koschmann, D. Suthers, & T.-W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years* (pp. 331-340). Mahwah, NJ: Lawrence Erlbaum Associates.

O'Donnell, A. M., & Dansereau, D. F. (1992). Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interaction in cooperative groups: The theoretical anatomy of group learning* (pp. 121-140). New York: Cambridge University Press.

Pintrich, P. R., Marx, R. W., & Boyle, R. A. (1993). Beyond cold conceptual change: The role of motivational beliefs and classroom contextual factors in the process of conceptual change. *Review of Educational Research, 63*, 167-199.

Schank, R., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding*. Hillsdale, NJ: Erlbaum.

Stahl, G. (2005). Group cognition in computer-assisted collaborative learning. *Journal of Computer Assisted Learning, 21(2)*, 79-90.

Weinberger, A. (2003). *Scripts for computer-supported collaborative learning. Effects of social and epistemic collaboration scripts on collaborative knowledge construction.* Doctoral thesis. Ludwig-Maximilian University, Munich. Available at: http://edoc.ub.uni-muenchen.de/archive/00001120/01/Weinberger_Armin.pdf.

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

PART IV

INTERDISCIPLINARY PERSPECTIVES

Chapter 16

DESIGNING INTEGRATIVE SCRIPTS

Pierre Dillenbourg and Patrick Jermann
Ecole Polytechnique Fédérale de Lausanne (EPFL)

Abstract: Scripts structure the collaborative learning process by constraining interactions, defining a sequence of activities and specifying individual roles. Scripts aim at increasing the probability that collaboration triggers knowledge generative interactions such as conflict resolution, explanation or mutual regulation. Integrative scripts are not bound to collaboration in small groups but include individual activities and class-wide activities. These pre- and post-structuring activities form the didactic envelope of the script. In many cases, the core part of the script is based on one among a few schemata: Jigsaw, conflict, reciprocal. We propose a model for designing this core component. This model postulates that learning results from the interactions that students engage in to build a shared understanding of a task *despite* the fact that it is distributed. Hence, the way the task is distributed among group members determines the interactions they will engage in. Interactions are viewed as the mechanisms for overcoming task splits. A large variety of scripts can be built from a small number of schemata, embedded within activities that occur across multiple social planes, activities which are integrated with each other by few generic operators.

1. INTRODUCTION

When teachers ask students to carry out collaborative activities, they usually provide them with global instructions such as "do this task in groups of three". These instructions are completed with implicit expectations with respect to the way students should work together, for instance an even group participation is often believed as desirable. A script describes the way students have to collaborate: task distribution or roles, turn taking rules, work phases, deliverables, etc. This contract may be conveyed through initial instructions or encompassed in the learning environment.

Scripts illustrate the convergence between instructional engineering and socio-constructivism. The need for engineering collaborative learning results from empirical studies on the effectiveness of collaborative learning. These studies show that this effectiveness depends upon multiple conditions such as the group composition (size, age, gender, heterogeneity, etc.), the task features and the communication media. These conditions are multiple and interact with each other in such a complex way that is not possible to guarantee learning effects (Dillenbourg, Baker, Blaye, & O'Malley, 1995). What predicts learning outcomes is the richness of social interactions (conflict resolution, elaborated explanations, mutual regulation, ...). Scripts aim at enhancing the probability that these knowledge productive interactions occur during collaboration. Hence, the key design issue: which interactions need to be scaffolded in order to reach the educational objectives?

Most chapters in this volume address the notion of scripts in computer-supported collaborative learning (CSCL). Within the classification proposed by King (this volume), our approach clearly belongs to the pedagogical stream: our scripts are pedagogical artifacts designed by educators and explicitly imposed on learners. The striking similarity between the many existing scripts resembles an invitation to produce a design model. Beyond the sake of modeling, this model could be used to foster exchanges among teachers or designers and to build tools for authoring CSCL scripts. It is not presented as a cognitive model of collaborative learning processes but as a design metaphor, i.e., a way to envision scripts. Its basic principle is to introduce a perturbation in a distributed system, so that the system will trigger repair mechanisms. These repair mechanisms require the knowledge-intensive interactions that the script aims to trigger.

2. EXAMPLES OF CSCL SCRIPTS

We present four scripts that we have developed and used with our own students. These examples will enable us to better describe the variety of scripts (section 3) and then to explain our design model (sections 5 and 6).

2.1 The "Concept Grid" script

The best-known collaborative script is the Jigsaw: each group member has only access to a subset of the information needed to solve the problem (Aronson et al, 1978) and therefore no individual can solve the problem alone. Group members should not simply forward information to each other: the member who owns a body of information has to process it, to become an "expert" of that sub-domain, in order to share it and to contribute to problem solving. The information given to group members defines their role. There exist multiple variations of the Jigsaw model. Some scripts alternate two types of meetings: students work in mixed groups (role-x role-y role-z), but from time to time, they form perpendicular groups, also called *expert groups* (role-x role-x role-x...) to share their expertise. In our example, knowledge distribution is induced by the script, but another script may also exploit 'natural' differences in prior knowledge: students with qualitative versus quantitative knowledge in physics (Hoppe & Ploetzner, 1999), students in medicine versus students in psychology (Hermann, Rummel, & Spada, 2001), students from different countries (Berger et al, 2001, see 2.4),...

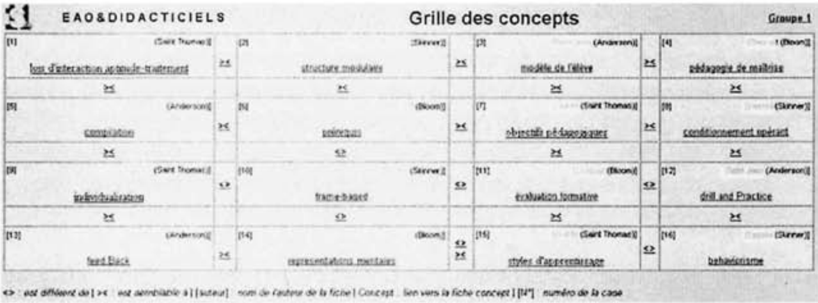


Figure 16-1. ConceptGrid Script in phase 4: Students build a grid of concepts. Each concept links to a definition they have written in phase 3. For each symbol between cells, they write a text explaining the similarity/difference between neighbor concepts. The 2 names in the cells are their own name (blurred) and the name of the role they are playing.

We implemented an instance of Jigsaw, the Concept Grid, in a master course on learning theories for educational software. Students have to learn the key concepts of the domain and the underlying theoretical framework. Figure 16-1 shows a grid produced for the first chapter, concerning learning theories in traditional computer-based teaching. The script runs as follows:

- Phase 1. Groups of four students are freely formed. They distribute roles among themselves. Roles correspond to theoretical approaches to be learned. In Figure 16-1, the roles are Skinner, Bloom, Anderson and Saint-Thomas. New roles are proposed for each chapter except for 'Saint-Thomas': his role is to be skeptical with regards to the effectiveness of the educational software under study and hence to review experimental studies. To enter into their role, students have to read three papers describing the related theory or studies.
- Phase 2. Groups receive a list of concepts to be defined. Examples of concepts appear in the cells of Figure 16-1. They cover the key notions that the teacher expects the learners to acquire. The group distributes

concepts to be defined among its members. The teacher does not specify which role is knowledgeable for which concepts.

- Phase 3. Each student writes a 10-20 lines definition of the concepts that were allocated to him/her.
- Phase 4. Groups assemble the concepts into a grid (see Fig. 16-1) and define the relationship between grid neighbors: The "<>" and "><" symbols are links toward a short text that describes relationship between two concepts: the symbol "<>" links to explanations that discriminate similar concepts (and could be confused by students) and the symbol "><" links to explanations that articulate concepts that are apparently unrelated. Groups have to try many organisations of the concepts on the grid before being able to define all relationships.
- Phase 5. The teacher analyses all grids before the debriefing session. During this session, he points out the inconsistencies between grids produced by different groups, the cases where close concepts have not been recognized as being similar and, vice-versa, concepts that have been associated while they have a very different meaning.

This script is not fully collaborative. Phase 3 is cooperative (each student individually writes a text). The core part is Phase 4: the only way to build the grid and to define the relationship between two concepts C1 and C2 is that the student who read about C1 explains it to the student who read about C2 and vice-versa. It cannot be a shallow explanation; they have to reach a reasonable level of shared understanding to write these "relationship" texts.

2.2 The "ArgueGraph" script

The "ArgueGraph" script was used in an educational technology course. The goal of the session was that students relate courseware design with learning theories. We tested several versions of this script, within two CSCL environments, with different combinations of co-presence and distance (Jermann & Dillenbourg, 2003). It includes five phases:

- Phase 1. Each student takes a multiple-choice questionnaire produced by the teacher. The questions have no correct or wrong answer; their answers reflect theories about learning. For each choice, the students enter an argument in a free-text entry zone.
- Phase 2. The system produces a graph in which students are positioned according to their answers (Figure 16-2). A horizontal and vertical score is associated to each answer of the quiz and the students' position is simply the sum of these values. Students look at the graph and discuss it informally. The system or the tutor forms pairs of students by selecting

peers with the largest distance on the graph (i.e., that have most different opinions).

- Phase 3. Pairs answer the same questionnaire together and again provide an argument. They can read their individual previous answer.
- Phase 4. For each question, the system aggregates the answers and the arguments given individually (Phase 1) and collaboratively (Phase 3). During a face-to-face debriefing session, the teacher asks students to comment on their arguments. The set of arguments covers more or less the content of the course but is completely unstructured. The role of the teacher is to organize the students' arguments into theories, to relate them, to clarify definitions, in other words, to structure emergent knowledge
- Phase 5. Each student writes a synthesis of arguments collected for a specific question. The synthesis has to be structured according to the theoretical framework introduced during the debriefing (Phase 4).

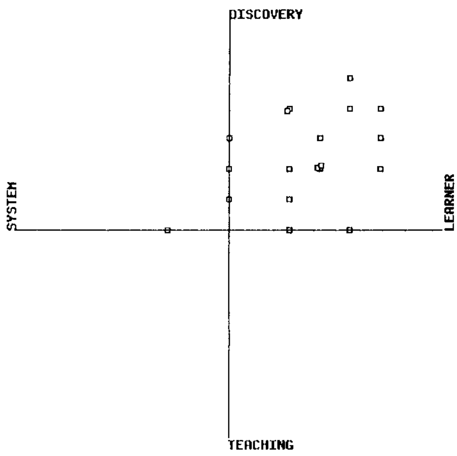


Figure 16-2. ArgueGraph, phase 3: Graph representing individual answers (names have been erased).

2.3 The "UniverSanté" script

This "UniverSanté" script was designed for teaching public health (Berger et al, 2001) in a course jointly given at the Universities of Geneva (Switzerland), Beirut (Lebanon), Monastir (Tunisia) and Yaounde (Cameroon). The students were divided into five thematic groups: AIDS, cancer, infectious diseases, cardiovascular diseases and accidents. Each thematic group includes four students of each country and a tutor. The script includes seven phases: starting from a clinical case (Phases 1 & 2), students address public health issues (3 to 5), explore methods of epidemiology (5 & 6) and build strategies to cope with public health problems (Phase 7).

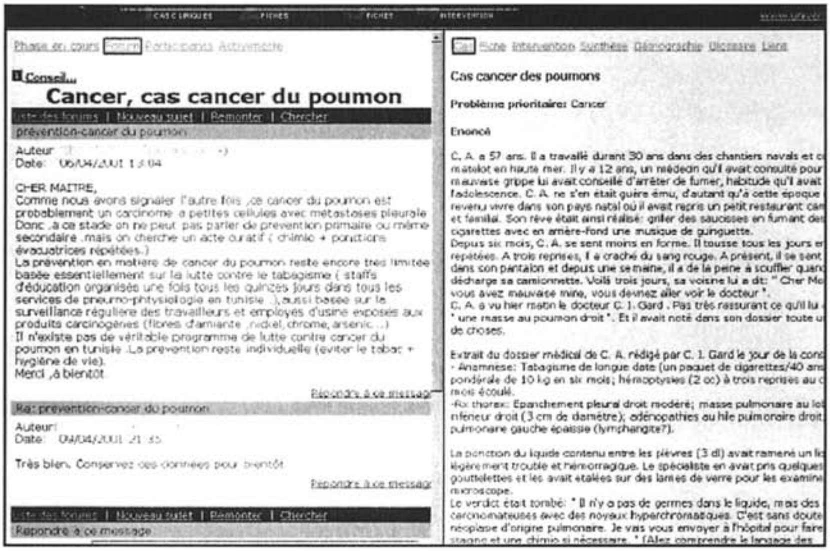


Figure 16-3. A snapshot from the UniverSanté environment.

- Phase 1. Each group receives a clinical case. For example, one "cancer" group works on the case of a woman with breast cancer whereas a second "cancer" group receives a case of a man with lung cancer. Each group discusses the case in a specific forum. The tutor guides the discussion in order to help the students identify and discuss the case with regard to public health.
- Phase 2. Two groups of the same country working on the same theme (e.g., the two "cancer" groups from Monastir University) interact through an on-line forum. A synthesis of the elements identified by each thematic group is presented during a face-to-face debriefing meeting in each country.
- Phase 3. Within a thematic group, the students of each country create a fact sheet describing the status of this public health problem in their country. For example, the Swiss students in the cancer group create a fact sheet "Cancer-Switzerland", which they enter into the database. The "Cancer" group of every country produces the same data.

- Phase 4. The students of each thematic group from different countries discuss the differences and the similarities between the fact sheets of the four countries in the forum.
- Phase 5. Fact sheets are discussed during a face-to-face debriefing meeting in each country. The tutor prompts the students to identify any issue concerning the way in which statistical data were collected, treated or presented.
- Phase 6. Students modify their fact sheet according to the methodological comments received in Phase 5.
- Phase 7. Each thematic group is divided into two subgroups working on the cases they studied during Phase 1. Each subgroup proposes a health strategy to cope with the problem. The students enter their strategy (objectives, actions, resources, evaluation) into the knowledge base through an on-line form.

This script generates interactions by playing with differences: differences between clinical cases of the same disease (phase 2) aim at generating abstraction; differences between the statistics collected in different countries generate discussion on the salience of the disease (phase 4) but also on the methods for collecting comparable data (phase 5). Comparison of the different societal answers to disease generates awareness of the public health policies.

2.4 The "Studio" script

As last example, our Courseware Design Studio is an adaptation from the PhaseX script (Engeli, 2001) for supporting project-based learning. The goal of the project was to design a courseware. The project is segmented into phases. At each phase, all teams deposit their intermediate product in a shared space. At the next phase, each team is allowed to borrow the work produced by another team and to continue its work from it. The phases were goal definition, content analysis, activity design, and so forth. The rationale for this script is that the shared space allows for a permanent idea-seeding. However, while it seems to work very well in Engeli's 3D-design projects, our students were reluctant to exchange intermediate results in their design process.

3. THE DIVERSITY OF SCRIPTS

This book presents a variety of scripts. Our scripts illustrate different script schemata but are still rather similar to each other compared to other examples in this book. This section reviews different understandings of a script while the next section specifies categories within our own scripts.

3.1 Role: Why playing a script?

For Kollar, Fischer, and Hesse (in press) and King (this volume), the term "external script" refers to the pedagogical scenario that students are asked to play, while the term "internal script" describes the mental representation that students construct of the external script. The external script is a didactic artifact to be used during a training session. The internal script is a cognitive structure that, in many cases, existed before the training session (e.g., "How to argue with a peer?") and will continue to exist after the training session. When the goal is that students internalize the script in order to reuse it in future situations, the script is a pedagogical *objective*. This is for instance the case of the reciprocal teaching script (Palincsar & Brown, 1984) which effectively fostered a high level of internalization. More modestly, the internalization of our Studio script was also an objective for our students, since the segmentation of courseware design into phases was something they had to learn.

When the script is "only" a method to be used during a training session and not internalized for the future, students still have to build some internal script in order to be able to participate in the learning activities. We did not expect our students to remember the ArgueGraph or the ConceptGrid scripts a few weeks later; we expected them to have learned the content being discussed in the script but not the script itself.

In summary, when the script is a method, the internal script is instrumental to play well the external script; when the internal script is the objective, it's the other way around. These are not exclusive: an argumentation script in which roles rotate may have as objectives both the content of argumentation (script as a method) and the ability to take the other's perspective (script as an objective). It is important to make explicit the status of a script before conducting an empirical study because they imply different forms of assessment, such as transfer task when the script is the objective and knowledge task when the content is the objective.

Finally, Harrer, Bollen, and Hoppe (2004) use *scripting collaboration* to refer to another pedagogical method: the post-hoc analysis of the interaction log files by the students themselves. This reflective activity is namely a useful phase when the script needs to be internalized. Our scripts are prescriptive while their approach is descriptive.

3.2 Congruence: Do they play the script?

When the teacher sets up an (external) script for the students, each of them constructs some internal script that will – to some extent – be different from the external script. Within a group, since students develop their own internal script, the interactions that actually take place will – to some extent – drift away from the interactions prescribed by the script. The congruence between the external script and emergent interaction patterns depends upon four script features: the degree of coercion, the intelligibility of the script, the degree of granularity and its fit to the team distribution. We now review these four congruence factors.

The first congruence factor is the *degree of coercion* of the script. A script may be simply conveyed through initial instructions or be regularly enforced by prompts or other design features. Although this is a continuous variable, we identified five levels of coercion (Dillenbourg, 2002) presented in increasing order

1. Induced scripts. The communication interface induces interaction patterns; it implicitly conveys the designer's expectations with respect to the way students should tackle the problem and interact with each other. This low degree of coercion is elegant but often not sufficient to significantly shape the collaborative processes.
2. Instructed scripts. Students receive oral or written instructions that they have to follow. The coercion is higher than in the induced script since the teacher's expectations are made explicit, but they can of course be misunderstood, incorrectly applied, forgotten or completely ignored. Students have to build an internal script that corresponds to the external script presented by the teacher.
3. Trained scripts. Students are trained to collaborate in a certain way before using the script it in a real learning situation. The degree of coercion is higher than in the instructed scripts since the teacher may control the student's internal script.
4. Prompted scripts: The system displays cues that encourage the learners to take their respective role (Weinberger, Fischer, & Mandl, 2002). Their system delivers cues (text messages), that are supposed to lead students to take specific roles such as "analyzer" or "critic".
5. Follow-me scripts. Students interact with an environment that does not allow them to escape from the script.

A high degree of coercion reduces the gap between the external script and emergent interaction patterns but increases the risk of *overscripting* (see 4). Our scripts have a low degree of coercion, obtained in various ways. In the ArgueGraph, the coercitive factor was the interface. In a first environment we used, pairs could only provide one answer per question and argumentation was more intensive than in a second environment, where the interface enabled them to enter more subtle answers. In the ConceptGrid, coercion was induced by the grid structure which forces the students to explain concepts to each other. The UniverSanté degree of coercion was very low and tutors had to permanently reinforce the script. In the Studio script, the most coercitive feature was the linear structure of the project segmentation. When coercion is naturally induced by the interface, as in ArgueGraph, we could talk about *affordances*, which sound more positive than coercion.

The second congruence factor is the *intelligibility* of the script. We face intelligibility problems with the UniverSanté script that occurred to be too complex (Berger et al, 2001) in this international public health course (students from Switzerland, Lebanon). Since we were aware of the script complexity, we provided teams with a graphical representation of the script and offered a close follow-up by teaching assistants, but nonetheless the students – and even some tutors – did not manage to construct a clear internal script. The interaction patterns drifted away from the external script.

The third congruence factor, *granularity,* refers to the time scale (duration of each phase) and the grain size of phases (subtasks) definition. For instance, the Studio script included a "programming" phase that lasted four weeks, the whole script running over the academic year, while the ArgueGraph script ran over four hours with phases ranging from 5 to 100 minutes. At the lower end, finest grain scripts reach the utterance level, i.e., specify the authorized dialogue moves at the next utterance. Fine grained scripts tend to be more coercitive. The gap between the external script and emergent interaction patterns may increase if there is a mismatch between the natural granularity of the task and the granularity enforced by the script. A mismatch could occur if the questions in ArgueGraph or the concepts in the ConceptGrid were too specific to capture the key differences between the theories under scrutiny. Another mismatch would occur if the Studio script structured a design phase as a sequence of questions while designers would address these questions in parallel.

The fourth congruence factor, *fitness*, is important for scripts that specify a distribution of roles among group members. For instance, one group member is asked to be leader or coordinator while another one is in charge of taking notes. The interaction patterns depend on the good match between the role requirements and the group members' skills or profiles. Fitness inspired various jokes such as "If the French member is in charge of cooking and the German one in charge of organization... (high fitness), but, if it is the other way around..." Low fitness is detrimental to role adoption and role adherence (students do not stick to the roles very long). Fitness inherits from transactive memory (Moreland, 1999), that is the representation that each group member has of the skills of the others: what matters is not only that team members are able to play their role but also that their team mates believe they are able to play that role. We did not encounter fitness problems in the ConceptGrid, for instance cases where one student would not manage to play "Skinner" for personal reasons. The fact they choose the roles themselves probably increases fitness. The fitness question is a greater concern in project-oriented scripts that select one student as team leader.

3.3 Granularity: Macro versus micro-scripting

We introduced the notion of script granularity as a continuous variable. There is however a qualitative difference between *macro* and *micro scripts*. Let us illustrate these differences with scripts that aim at raising argumentation. A micro-script scaffolds the interaction process per se: when learners state a hypothesis, the script will for instance prompt their peer to produce counter-evidence. A macro-script sets up pairs in which argumentation should occur, as in the ArgueGraph, by pairing students with opposite opinions. The micro-script reflects a psychological perspective, acting on the internal script (scripting as a goal), while the macro-script reflects an educational perspective, influencing the process more indirectly (scripting as a method). Micro and macro-scripts do not constitute clear-cut categories but rather define a continuum. Most examples described in this volume are on the "micro" side: in the work reported by King (this volume), by Lauer and Trahash (this volume), by Weinberger et al. (this volume), by Carmien et al. (this volume), the script includes prompts that directly scaffold interactions and is expected to be internalized as higher-order thinking skills (argumentation, problem solving or metacognition). The grain size is somewhat coarser in the scripts of Rummel and Spada (this volume), and Ertl et al. (this volume), where the script prompts episodes of interactions. The example presented by Kolodner (this volume) is, like our examples, on the macro side. Ayala (this volume), and Haake and Pfister (this volume) describe environments that articulate micro-scripts within phases of a macro-script.

3.4 Integrated learning

We use CSCL scripts for promoting a vision of e-learning that is broader than what the *CSCL* label may indicate. Our script examples are neither strictly collaborative, nor strictly computerized; they illustrate our *integrated learning* approach that we define with 3 features:

- Despite the first *C* in *CSCL*, there is no reason to restrict CSCL scripts to distance interactions. ArgueGraph and ConceptGrid scripts have mostly been used in a situation where students were co-present. UniverSanté used distant interactions, since geographical diversity was the key principle, but still included key face-to-face discussion (one per country). Computers are justified by other reasons than simply connecting distant learners (see section 4). Integrated learning differs from the so-called 'blended learning', which is often the mere juxtaposition of face-to-face and computer-mediated activities. Integrated learning scripts articulate activities which are on-line or not, in front of a computer or not, occurring across a variety of places (classroom, lab, field trip, home, work ...). The rapid transition between activities with or without computers is facilitated by lighter/mobile hardware. Integration is pedagogical but also functional: scripts support data flow between multiple activities (see 7.4). For instance, in the ArgueGraph, the individual answers (phase 1) are used to form pairs (phase 2) and the pairs' answers and arguments are collected for the debriefing (phase 4).

- Despite the second *C* in *CSCL*, there is no reason why collaborative learning should be treated as an exclusive pedagogical approach. Instead, group activities gain from being integrated with other classroom activities. Scripts may include individual work (e.g., writing a synthesis, reading a paper,...) and/or class-wide activities (introductory lectures, debriefing, ...). In ArgueGraph, phases 1 (answering the quiz) and 5 (writing a summary) are individual while phases 2 (observing the graph) and 4 (debriefing) are done with the whole class. In the ConceptGrid, phase 3 is individual (reading papers and writing concept definitions) while phase 5 (debriefing) is at the class level. The designers' challenge is to integrate these diverse activities within one consistent script.
- Last but not least, the illustrated scripts maintain the teacher in his leading role. He or she is not properly teaching but is active and salient as the *chef d'orchestre* of the whole script: he or she may shorten a phase, regulate groups, give feedback, etc. We therefore should be concerned by the script *flexibility*, i.e., the possibility for the teacher to modify the script on the fly (see section 4).

These features define what we refer to *as integrated learning*, a pedagogical approach that is broader than the approach indicated by the terms *collaborative* and *computer* in CSCL. However, the breadth of this concept may weaken the identity of a *script*. Are scripts just a trendy word to refer to lesson plans? No! CSCL scripts are instructional sequences in which peer interactions are targeted to be the core learning mechanism. Therefore our design model distinguishes the core script, which governs collaborative interactions, from the didactic envelope, that encloses the core activities into other activities, forming the integrated learning approach.

4. BENEFITS AND RISKS IN COMPUTERIZED SCRIPTS

This volume concerns scripts in computerized environments. What is the added value that technology brings to the use of scripts? What are the drawbacks? We start with the advantages:

- *Connecting*: When scripts include remote activities, technology is simply the communication tool.
- *Sharing*: Computers provide a space for sharing products, allowing teams to get inspired by what other teams produce, as in the Studio script. This *simple* feature is important, as long as plagiarism can be controlled.
- *Management*: Computerized scripts off-load teachers from some logistics duties such as time management (reminding deadlines, ...) and information flows (e.g., distributing data to different group members).
- *Reification*: Computerized scripts provide students with a concrete representation of the external script, which is dynamically updated.
- *Scaffolding*: Computerized scripts offer opportunities for shaping communication with semi-structured communication interfaces and dialogue grammars or both (as illustrated by Runde et al, this volume).
- *Traceability*: Computerized scripts enable recording interactions and outputs, which, despite privacy concerns, enable teachers to analyze and regulate teamwork and enable students to reflect upon previous steps.
- *Adaptivity*: Computerized scripts enable *dynamically* generated events that would be harder to create without computers, such as, in the ArgueGraph, finding peers with most opposite opinions. Real-time adaptations can be improved by real time analysis of interactions among peers (e.g., Soller, Martinez, Jermann, and Muehlenbrock, 2005).

Among the drawbacks of computerized scripts, we find the general disadvantages of computer-mediated communication versus face-to-face communication. It is not the place here to review them (see Bromme, Hesse, & Spada, 2005). With integrated scripts, these drawbacks are compensated by face-to-face situations (see 3.4).

A key problem is the loss of *flexibility*. Good teachers adapt their plans on the fly, while a computerized script can hardly be modified in real time. Of course, the very idea of a script implies a decrease of flexibility: a script aims at structuring group processes, which requires some rigidity. However, implementing the script often generates constraints that are not part of the pedagogical intentions. Designers have to disentangle the flexibility loss inherent to the pedagogical intentions from the flexibility loss that is an undesired effect of translating the script idea into a computer program (Dillenbourg & Tchounikine, accepted).

Another risk is what we called *over-scripting* (Dillenbourg, 2002), i.e., situations that constrain natural collaboration in a way that makes it sterile, inhibiting the natural peer interaction mechanisms. Factors of over-scripting are:

- Disturbing natural interactions. If a learner wants to express A while the CSCL system only offers means for interactions B or C, either the learner will fail to say what he wanted to say or he will pervert the system (e.g., re-purpose B to say A). If similar breakdowns occur frequently, they may spoil the collaboration process. This risk concerns scripts that cumulate a high degree of granularity and a high degree of coercion.
- Disturbing natural problem solving processes. A script usually segments a global task into a sequence of activities. In our Studio script, this segmentation was a problem for students who had a holistic approach of courseware design. The script proposed an approach that was very linear. Some students rejected this artificial linearization. Our Grid script also introduces coercion with respect to the task: it is easier to draw a free concept map than to arrange concepts on a two dimensional grid. To some degree, coercion may become incompatible with the students' cognitive processes. Overscripting may then make the task so hard that it spoils the students' motivation.
- Increasing cognitive load. Complex scripts may interfere with the main learning process by augmenting the learners' cognitive load. The extraneous load comes from the necessity to understand, memorize and execute the script. However, an alternative hypothesis is that scripts reduce cognitive load by partly offloading interaction management (Dillenbourg & Bétrancourt, 2006).
- "Didactising" collaborative interactions. Collaborative problem solving triggers natural interactions. A peer asks a question because he wants to know the answer, while a teacher usually asks questions which he already knows the answer to. Peers negotiate a concept when they disagree on interpreting the phenomenon they jointly observed while teachers discuss concepts for which they own the right definition. A danger of "didacticised" interactions is to miss the engagement that is expected from genuine collaboration.
- Goalless interactions. Collaboration is driven by a shared goal. Scripts being quite didactic, they may prevent students from adopting the script goals as their own goals. The more the scripts segment collaboration into subprocesses, the more it seems difficult for team members to forget the didactic nature of the script.

Pitfalls are numerous; scripts need to be thoughtfully designed. The rest of this chapter investigates the design of CSCL scripts.

5. THE STRUCTURE OF SCRIPTS

Integrated learning scripts include a kernel, the core script, and a set of pre- and post-structuring activities, the didactic envelope. The core script is the collaborative activity in which the interactions that the script is intended to trigger should appear. In the ArgueGraph script, the core activity is the formation of conflicting pairs and argumentation triggered for answering the questionnaire together (Phase 4). In the ConceptGrid script, the core activity is the distribution of knowledge and the mutual explanation process necessary to build the grid (Phase 4). In the UniverSanté script, the core activity is when students must identify similarities and dissimilarities between the ways different national health systems cope with the same medical issue. The core script defines how the knowledge or task is *distributed* over the group members. We therefore borrow the distributed cognition model as explained in section 6.

The didactic envelope encloses the core script with other activities that contribute to the script consistency. Pre-structuring activities provide the conditions necessary to make the core script activities work well: introductory lectures, readings, exercises to activate pre-requisite skills, metaphors, etc. They namely enable students to play their role in the script. Post-structuring activities include debriefing activities such as the comparison of multiple solutions, synthesis lectures or readings, summary writing, etc. These are mostly reflective activities, aimed at turning group experience into knowledge. The activities in the envelope make the difference between collaborative learning in a restricted meaning and integrated learning, as explained in section 3.4.

The envelope has two salient features, its temporal structure and its social structure. A clear time structure differentiates scripts from free collaboration: scripts define a sequence of phases and in many cases these phases are limited in time. The rationale for setting up a semi-rigid time frame is threefold:

- Time management is a critical factor in everyday educational practice, for both teachers and learners. It is even more important for web-based activities taking place outside the time habits that exist in schools.
- The time structure facilitates teacher regulation by providing him or her with an easy way to follow the teams' progress.
- The time structure makes the task distribution more salient, especially since deadlines define clear boundaries between consecutive subtasks.

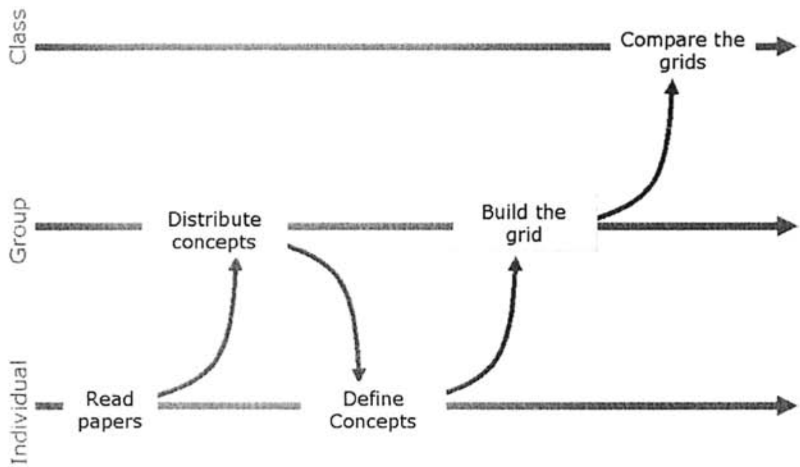


Figure 16-4. Structure of the 'ConceptGrid' script, time is represented horizontally and the social structure vertically.

The second dimension of integrated learning scripts is their social structure: activities occur at different social planes. Vygostky (1978) discriminated three planes: the intra-psychological plane, the inter-psychological plane and the social plane. The intra-psychological plane is individual. The difference between the inter-psychological and the social plane is not clear-cut, group size is a continuous variable, but there is a cognitive threshold: group activities occur at the inter-psychological plane as long as team members maintain some representation of their teammates' cognition; the social plane is the level where individual representations disappear behind the culture that the community members jointly constructed. If we relate these psychological levels to CSCL environments, we usually observe five levels of activity:

- Individual Plane: Solo activities.
- Group Plane: Activities in small groups ranging from two to, let's say, eight people. This is where proper collaboration occurs.
- Class Plane: Activities involving all students enrolled in the same course. We also refer to them as collective activities.
- Community Plane: Activities that involve external but identified actors such as other classes, expert groups, families. For instance, when a class

from school X designs a mathematical challenge for all other classes in the community, this activity is at the community level.

- World Plane: Activities that are accessible to unidentified actors, for instance when a class journal is produced on the web, the entire world may read it. If a survey is conducted via the web, any user may vote.

What matters here is not to agree on the exact definition of the levels but to stress the fact that script activities define moves across multiple planes. Figure 16-4 illustrates the time by social structure of the script "Concept Grid". One could argue that activities always occur at multiple planes: individual cognition does not freeze during class interactions and culture does not stop shaping our thinking during individual work. Activities do occur in parallel on multiple planes, but their focus varies with time.

The curved arrows on Figure 16-4 represent what we called *functional integration* in section 3.4. Functional integration refers to dataflow between activities at different planes. The output of an activity A_i at social level N is later on reused by an activity A_{i+1} at social level M, in many cases, N being different from M. This dataflow may appear as a technical feature, but in fact it affords the design of innovative scripts by combining storing, processing, distributing and representing data during collaborative learning. These data are student productions (answers in the ArgueGraph, concept definitions in the ConceptGrid and deliverables in the Studio) and student interactions (e.g., their arguments in the UniverSanté). We describe dataflow operators in section 7.4.

6. THE SWISH MODEL

How to design the core script activities? We propose a model, called SWISH, which borrows the distributed cognition vision, according to which a group of actors and the tools they use can be understood as a single cognitive system. The components of the system are the students who participate in the scripted teamwork as well as the tools and resources available. The script itself can be considered as a tool that shapes the functioning of the distributed system.

The core script defines the organization of a distributed cognitive system i.e., which team member will perform which subtasks. We refer to subtasks in a generic way: they can be independent from each other, like in cooperative work, or tightly coupled, like when one peer has to regulate the other. Scripts often define roles that induce a somewhat natural distribution of work into subtasks.

Why would we formalize task distribution while we aim to support collaboration? A formal task division appears to be in contradiction with the close interactions expected in collaboration (Dillenbourg, 1999). Since collaborative learning is often defined as the process of constructing and maintaining a shared understanding of the task (Roschelle & Teasley, 1995), it may sound counter-intuitive to split the task among different learners: this opens the door to misalignment of views, understandings and goals. To the same extent, scripts that foster conflict among peers would be detrimental to the construction of a joint solution. To bypass this counter-intuition, we rely on Schwartz' (1995) definition of collaborative learning as the effort necessary to build a shared understanding. Learning is the side effect of the cognitive processes triggered by the interactions (explanation, argumentation, mutual regulation, etc.) engaged to develop this shared understanding. Scripts that trouble a smooth collaboration increase the cognitive effort and hence are expected to augment the learning outcomes. In other words, learning results from over-compensating the drawbacks of task distribution.

This principle is the base of our design model: "Split Where Interaction Should Happen". SWISH can be formulated in three points:

1. Learning results from the interactions students engage in while constructing a shared understanding of the task *despite* the fact that the task is distributed.
2. Hence, the task distribution determines the nature of interactions. Interactions are mechanisms for *overcoming* task splits.
3. Hence task splits can be, following some kind of *reverse engineering*, designed for triggering the interactions that the designer wants to foster: Split Where Interaction Should Happen.

This model can be applied for describing the main script schemata, i.e., classes of scripts. We distinguish three basic schemata:

- The *jigsaw schema* distributes the knowledge or information necessary to solve the task, either by forming pairs that have complementary knowledge (e.g., in UniverSanté, students from different countries import knowledge of their national health system) or by providing them with complementary information (e.g., different readings in the ConceptGrid). Since none of the group members has enough information or knowledge to solve the task alone, they need to explain or justify their knowledge or contribution to others. For describing the ConceptGrid in SWISH terms, the split is performed by distributing information and it is compensated by explaining concepts to each other.
- The *conflict schema* triggers argumentation among group members by forming pairs of students with conflicting opinions (e.g., ArgueGraph), by providing them with conflicting evidence or by asking them to play conflicting roles. For describing the ArgueGraph in SWISH terms, the

split is performed by finding peers with conflicting opinion and it is compensated by argumentation.

- The *reciprocal schema* defines two roles in teams, one of the peers regulating the other and then switching roles. A well known example is the reciprocal teaching approach (Palincsar & Brown, 1984). For describing the reciprocal tutoring script in SWISH terms, the split is performed horizontally, between cognitive and metacognitive layers of the task and is compensated by mutual regulation. Since the cognitive and metacognitive subprocesses need to remain tightly coupled, the only way to build a shared solution is that peers continuously engage in mutual regulation interactions.

7. GENERALIZING SCRIPTS

As any pedagogical method, scripts raise hopes of generalization: can we reuse these scripts to teach a large variety of contents? The ArgueGraph script can be used for different subject matters but is only relevant in domains where key notions can be argued about. The ConceptGrid script can be generalized to many conceptual sets, but not all conceptual domains can be segmented as in the grid. The UniverSanté was very specific to the content to be taught, public health: using national differences is a natural way to let students discover the variety of societal answers to a similar medical problem. The Studio script can be generalized to a variety of design processes but with the constraint that this design process should be rather linear.

Generalisability is not bound by classical scientific boundaries (e.g., a script would be good for mathematics but not for social sciences) but by the specific learning objectives (ArgueGraph could be used in mathematics if students argue to choose among three ways to compute a value). In other words, there is definitely a potential of generalisability; a script is not universally relevant but can be reused in various domains.

7.1 Descriptive model

Scripts can be defined as variations of a generic template with a limited set of attributes. Most scripts can be defined with a limited number of components (groups, participants, roles, activities and resources) and mechanisms that capture the dynamics of scripts, i.e., how individual learners are distributed over groups (group formation), how roles, activities or resources are distributed over participants (component distribution) and how both components are distributed over time (sequencing) (Kobbe, Weinberger, Dillenbourg, Harrer, Hämälainen, & Fischer, submitted).

This simple description scheme could be translated within an educational modeling language (EML). These languages propose a well-structured terminology for describing instructional sequences. They do constitute a step forward compared to the content-centric approach of the educational metadata initiatives. However, they do not constitute a design model; they provide a description of the scripts but fail to capture the core idea of a script. A design model should describe the mechanisms by which the script is expected to generate learning. IMS Learning Design[1] could be expanded to model the core script, but groups are not defined explicitly but indirectly by assigning *roles*. This prevents for instance building a jigsaw script where team members have different roles within each team, or a reciprocal teaching script where roles rotate among group members at each script phase. The social structure of a script should be explicitly represented in the model.

Instead of producing yet another pedagogically neutral authoring language, we deliberately aim for a non-neutral model, i.e., a modeling scheme that conveys specific pedagogical ideas. This is the condition to produce scripts that differ from genuine lesson plans. Therefore, instead of looking for a highly abstract modeling scheme, we identify classes of similar scripts and infer their core idea, their identity.

7.2 Script schemata

Despite the diversity of scripts, there are recurrent patterns. We called them *schemata* instead of *patterns* to avoid confusions with the term *design pattern*, which has a more technical meaning in software engineering. A schema simply indicates commonalities among scripts, independently from the algorithms used by the CSCL environments to support these scripts. For instance, scripts that belong to the jigsaw schema have in common to distribute the necessary information among team members. Schemata are more abstract than programming structures, but if we translate them into software components, we could design tools that reduce the computational burden of CSCL script construction.

In section 6, we described three types of script schemata, the jigsaw schema, the conflict schema and the reciprocal schema. Other methods for group-based learning can also be described as script schemata:

- The *project* schema defines phases of a project, roles among teams (moderator, leader, writer, ...) and a calendar of intermediate deliverables. These scripts vary in coercion: does each team work on the same project; are they free to define the phases of their work and the calendar. The focus is often put on the regulation of project work.

[1] http://www.imsglobal.org/learningdesign

- *Problem-based learning* (Koschmann, Kelson, Feltovich, & Barrows, 1996) covers a variety of scripts that, despite differences, include similar phases: analysing the problem, defining learning objectives, acquiring the necessary knowledge and solving the problem collectively.
- The *science making* schema includes scripts in which team work is structured into a sequence of phases that drive learners through the scientific process of knowledge construction, as researchers are supposed to do. One example for these schemata is inquiry based learning (Hakkarainen & Sintonen, 2002).

We stress the fact that these schemata are not recipes for collaborative learning. They provide a general structure but the art of design is to apply this structure to the specific learning objectives, the peculiarities of the target audience and the specific content.

7.3 Generalization hierarchy

The ConceptGrid is a subclass of jigsaw schema. We could reuse the same script but replacing the Cartesian grid used in Phase 4 by a graphical concept map. The diversity of links between concepts that is offered in a concept map might be more appropriate to complex semantic fields. This new script, let's call it *ConceptGraph*, and the ConceptGrid are two subclasses of a higher script class, let's call it ConceptStructure.

In the ArgueGraph, a subclass of the argumentation schema, pairs are formed on the basis of their distance on the graph. The distance is computed by associating an X- and Y- value to each answer. These values are not computed in a scientific way, they are arbitrarily fixed by the designer. Their interest is to provide positions on the map with a semantic value. To avoid this arbitrary value allocation, one could use an algorithm that forms pairs of students with the lowest number of common answers. Let's call this new script *ArgueList* and their super-class *ArgueFromQuizz*. In another version, rather than using a chat or face-to-face discussion, we could have students argue with a semi-structured communication interface such as Belvedere (Suthers et al, 2001).

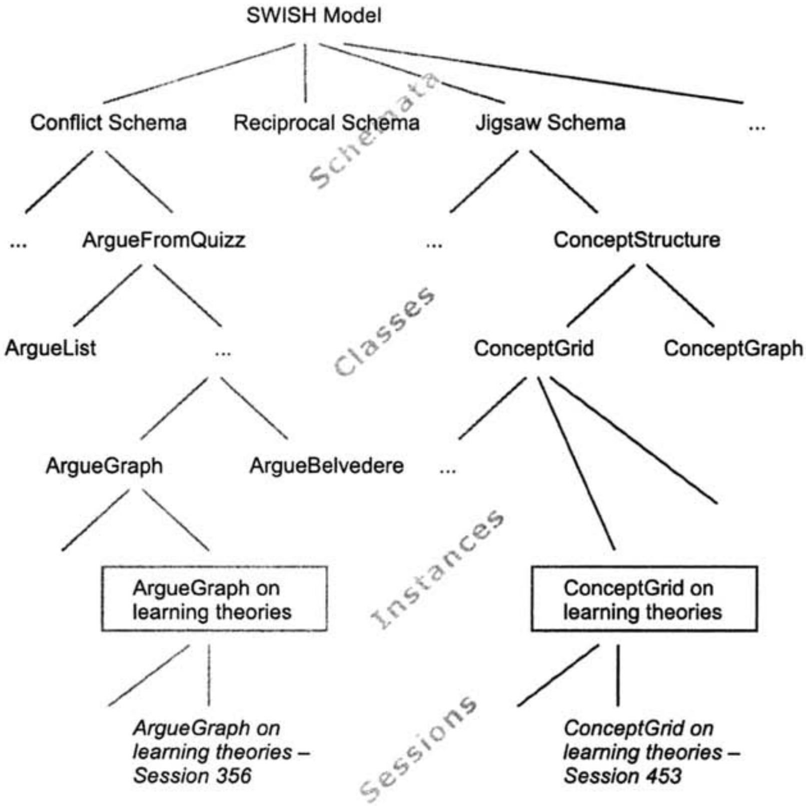


Figure 16-5. Generalization hierarchy for CSCL scripts.

Figure 16-5 represents the hierarchy of generalization. We arbitrarily discriminate four levels:

- Schemata describe the core mechanism of a large set of scripts.
- Script classes and subclasses define scripts, including their didactic envelope, independently from a specific content.
- Instances are scripts that have been instantiated with a specific content.
- Sessions are scripts instances with the student-specific data (users per groups, deliverables, …), dates, etc.

This hierarchy is not a proper tree: A script may borrow ideas from several schemata. The UniverSanté script for instance plays with both the complementarity (Jigsaw schema) and the conflicts among students knowledge.

One could argue that this tree-like representation is not appropriate. For instance, in the ArgueFromQuizz family (a subgraph), the mode of pair formation (Phase 2: graph distance versus common answers) is independent from the mode of argumentation (Phase 4: free versus semi-structured dialogues). Each of the pair formation modes could be combined with each of the argumentation modes. A script grammar, combined with these different modes as vocabulary, would be more powerful for describing all possible combinations. However, syntax may not carry semantics. A combinatorial approach may lead to assemble script elements into something that does not constitute a script, i.e., a sequence of events that will not trigger specific interactions. Script classes make the design space discrete, which is a simplification, but enables to convey the design rationale.

7.4 Executable model

As pointed out by Kobbe et al (submitted), a script can be defined as a number of mechanisms that manipulate a set of script components (roles, activities, ...). Some of these components are intrinsic to the script class (e.g., the grid structure of ConceptGrid; some prompts in ArgueGraph), some objects are specific to the script instance (e.g., the questions included in an ArgueGraph on biology; the list of documents to read in a ConceptGrid on history) and some objects are specific to a script session (e.g., the definition produced by ConceptGrid students; the answers produced by ArgueGraph students). An executable model of script has to manipulate these objects, e.g., to allocate individuals to groups or roles to individuals, to gather answers within a group or conversely to distribute data among group members, etc. We expect these mechanisms to be formalized as the combination of a limited number of basic operators.

Dataflow operators. Dataflow enables the design of dynamic CSCL scripts. The dataflow used in our scripts can be described by a small number of operators for moving up and down the planes of the social structure. The research in producing these operators should benefit from the advances on workflow technology, namely workflow standards such as WFXML[2].

- *Upward* operators are aggregate, list, differentiate, etc. They collect data at a social plane and turn them into a data structure at higher social plane. The type of processing depends on the nature of data. If data are structured in a table with social planes (individuals, groups, ...) in rows and task outputs in columns, we can define simple operators. The *aggregation* operator collapses columns, e.g., computes a value (sum, mean, ...) for all individuals. The *differentiation* operator collapses columns (data)

[2] http://www.wfmc.org/standards/wfxml_demo.htm

for each user. It is used in the ArgueGraph, where the answers to the 10 questions are summarized into a [X Y] pair for each individual in order to plot them on the graph. When data are too complex to be turned into a single value, they can simply be listed as it is the case in the Studio script (*list operator*).

- *Downward* operators distribute an object among members of the lower social plane. For instance, in the ConceptGrid, each group of four students (social plane 2) is associated with four roles and 12 readings. A downward operator would distribute the roles and readings to each team members (social plane 1). As for aggregation, the simplest operator is non-transformational: it distributes the same object O from plane N to all members at plane N-1.

Social operators. Other operators transform the structure of the groups either by reallocating roles within a group (role rotation) or by moving individuals between groups (group rotation).

- The *role rotation* operator redistributes the roles (subtasks) among group members at different phases. Role rotation reinforces the distributed system model that underlies the SWISH model. A set of interrelated components can be depicted as "a system" if it is capable of plasticity, i.e., to reallocate dynamically subtasks to different subcomponents. The rotation operator enforces this plasticity.
- The *group rotation* operator redistributes individuals among teams. It is applied in scripts where individuals are member of two groups, namely in Jigsaw scripts, where an individual sometimes works with his or her team but sometimes works with the individuals that have the same role in other teams.
- The *group formation* operators determine how groups are formed from individuals: it relies on the difference of opinions in ArgueGraph and the complementary of knowledge in Hoppe and Ploetzner's (1999) scripts.

These few examples of operators stress both the usefulness and the complexity of developing abstract mechanisms that would apply to a variety of script domains.

8. SYNTHESIS

The SWISH model can be explained simply. First, one introduces a perturbation in a distributed system, by splitting it. Second, the system triggers repair mechanisms for reducing the perturbation. These repair mechanisms – hopefully – are knowledge-intensive interactions that produce learning.

Learning is therefore the result of over-compensating the drawback of task splits.

However, this model only holds if the group has both the ability and the will to compensate task splits. In some cases, solving conflicts, explaining complex concepts or regulating bad problem solvers may be beyond the skills of individuals. In other situations, the motivation to reach a shared understanding may be insufficient. The SWISH model is only valid for tasks that require a high level of shared understanding. If students manage to solve the task without constructing a shared understanding, repairing the system will not be worth the effort.

SWISH is not a cognitive model grounded in experimental results. We used these scripts with our own students, but only two of them have been formally assessed. However, no script could be proved to be generally effective. We cannot establish the effectiveness of a script class in general since it depends on its relevance for specific learning objectives and target groups. Nonetheless, by describing CSCL scripts in a structured way, this chapter may help researchers to clarify the variables they investigate when running experimental studies.

This framework contributes to design tools for authoring CSCL scripts. Most scripts are implemented in specific CSCL environments. Our script examples were implemented as dynamic web pages, generated with PHP programs from database contents (MySQL). Not all teachers can install a database and write PHP. Tools for authoring CSCL scripts aim at promoting practices of e-learning that are more innovative than those offered by existing learning management systems.

ACKNOWLEDGEMENTS

This chapter benefited from the discussions within CoSSICLE a research group of KALEIDOSCOPE, a European Network of Excellence on "*Concepts and methods for exploring the future of learning with digital technologies*", funded by the European Union and the Swiss Federal Office for Education and Science. This work is also partly funded by the Swiss Center for Innovation in Learning (SCIL; University of St Gallen, Switzerland). This chapter has been enriched by comments from Frank Fischer, Armin Weinberger, Karsten Stegmann, and the book reviewers.

REFERENCES

Berger, A., Moretti, R., Chastonay, P., Dillenbourg, P., Bchir, A., Baddoura, R., et al. (2001). Teaching community health by exploiting international socio-cultural and economical differences. In P. Dillenbourg, A. Eurelings, & K. Hakkarainen (Eds.). *Proceedings of the first European Conference on Computer Supported Collaborative Learning* (pp. 97-105). Maastricht.

Bromme, R., Hesse, F., & Spada, H. (Eds.) (2005). *Barriers and biases in computer-mediated knowledge communication. Computer-Supported Collaborative Learning Series.* New York: Springer.

Brousseau, G. (1998). Théorie des situations didactiques. Grenoble: La Pensée Sauvage.

Dillenbourg P. (1999). What do you mean by collaborative learning? In P. Dillenbourg (Ed.), *Collaborative-learning: Cognitive and computational approaches* (pp. 1-19). Oxford: Elsevier.

Dillenbourg, P., & Bétrancourt, M. (2006). Collaboration Load. In J. Elen & R. E. Clark (Eds.), *Handling complexity in learning environments: research and theory* (pp. 142-163). Advances in Learning and Instruction Series, Pergamon

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen: Open Universiteit Nederland.

Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. (1995). The evolution of research on collaborative learning. In H. Spada & P. Reimann (Eds.), *Learning in humans and machine: Towards an interdisciplinary learning science* (pp. 189-211). Oxford: Elsevier.

Dillenbourg, P., & Tchounikine, P. (in press). Flexibility in macro-scripts for CSCL. *Journal of computer assisted learning.*

Engeli, M. (Ed.). (2001). *Bits and spaces, architecture and computing for physical, digital, hybrid realms.* Basel: Birkhäuser Publishers.

Hakkarainen, K., & Sintonen, M. (2002). The interrogative model of inquiry and computer-supported collaborative learning. *Science & Education, 11*(1), 25-43.

Harrer, A., Bollen, L., & Hoppe, U. (2004). Processing and transforming collaborative learning protocols for learner's reflection and tutor's evaluation. In E. Gaudioso & L. Talavera (Eds.), *Proceedings of the European Conference on Artificial Intelligence.* Valencia.

Hoppe, U. H., & Ploetzner, R. (1999). Can analytic models support learning in groups. In P. Dillenbourg (Ed.), *Collaborative-learning: Cognitive and Computational Approaches* (pp.147-168). Oxford: Elsevier.

Hutchins, E. (1995). How a cockpit remembers its speeds. *Cognitive Science, 19,* 265-288.

Jermann, P., & Dillenbourg, P. (2003). Elaborating new arguments through a cscl scenario. In G. Andriessen, M. Baker, & D. Suthers. (Eds.), *Arguing to learn: confronting cognitions in computer-supported collaborative learning environments. Computer-Supported Collaborative Learning Series.* Amsterdam: Kluwer.

Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., & Fischer, F. (submitted). *Specifying Computer-Supported Collaboration Scripts.*

Kollar, I., Fischer, F., & Hesse, F. W. (in press). Computer-supported collaboration scripts - a conceptual analysis. *Educational Psychology Review.*

Koschmann, T., Kelson, A. C., Feltovich, P. J., & Barrows, H. S. (1996). Computer-supported problem-based learning: A principled approach to the use of computers in collaborative learning. In T. Koschmann (Ed.), *CSCL: Theory and practice of an emerging paradigm.* Mahwah, NJ: Lawrence Erlbaum Associates.

Lave J. (1991). Situating learning in communities of practice. In L. Resnick, J. Levine, & S. Teasley (Eds.), *Perspectives on Socially Shared Cognition* (pp. 63 - 84). Hyattsville, MD: American Psychological Association.

Moreland, R. L. (1999). Transactive memory: Learning who knows what in work groups and organizations. In L. Thompson, D. Messick, & J. Levine (Eds.), *Shared cognition in organizations: The management of knowledge* (pp. 3-31). Mahwah, NJ: Lawrence Erlbaum Associates.

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

Palincsar A. S., & Brown A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 1*(2), 117-175.

Roschelle, J., & Teasley S. D. (1995). The construction of shared knowledge in collaborative problem solving. In C. E. O'Malley (Ed.), *Computer-supported collaborative learning.* (pp. 69-197). Berlin: Springer-Verlag

Salomon, G. (1993). No distribution without individual's cognition: a dynamic interactional view. In G. Salomon (Ed.), *Distributed cognitions. Psychological and educational considerations* (pp. 111-138). Cambridge, USA: Cambridge University Press.

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

Soller, A., Martinez, A., Jermann, P., & Muehlenbrock. M. (2005). From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. *International Journal of Artificial Intelligence in Education*, 15, 261-290.

Suthers, D., Connelly, J., Lesgold, A., Paolucci, M., Toth, E., Toth, J., et al. (2001). Representational and Advisory Guidance for Students Learning Scientific Inquiry. In K. D. & P. J. Feltovich (Eds.), *Smart machines in education: The coming revolution in educational technology* (pp. 7-35). Menlo Park, CA: AAAI/Mit Press.

Vygotsky, L. S. (1978). *Mind in society. The development of higher psychological processes.* Cambridge, MA: Harvard University Press.

Weinberger, A., Fischer, F., & Mandl, H. (2002). Fostering computer supported collaborative learning with cooperation scripts and scaffolds. In G. Stahl (Ed.), *Computer support for collaborative learning: foundations for a CSCL community. Proceedings of the International Conference on Computer Support for Collaborative Learning (CSCL) 2002* (pp. 573-574). Hillsdale, NJ: Lawrence Erlbaum Associates.

Chapter 17

THE INTERPLAY OF INTERNAL AND EXTERNAL SCRIPTS

A Distributed Cognition Perspective

Stefan Carmien[1], Ingo Kollar[2], Gerhard Fischer[1], and Frank Fischer[2]
[1]University of Colorado, Boulder; [2]Knowledge Media Research Center, Tübingen

Abstract: This chapter describes different script types that are involved when a person X is accomplishing a particular task Y. We refer to concepts and ideas from distributed cognition theories. It is assumed that individuals are holding internal scripts that guide them in the way they process tasks they are faced with, and these internal scripts are standing in a complex relationship to the external scripts provided by an artifact or by other persons. Three factors are regarded as crucial in order to describe the accomplishment of a task, namely (a) the actual activity, (b) knowledge underlying the activity, and (c) the executive function, a (meta-)cognitive instance that is setting the goals for the task and controls the system's task accomplishment. For each of these three main factors, several sub-categories are introduced, on which two script approaches are compared. The first approach represents the socio-technical environment Memory Aiding Prompting System (MAPS) designed to support individuals with cognitive disabilities in accomplishing everyday tasks with a focus on "tools for living". The second approach, the so-called collaborative argumentation script, represents a computer-supported collaborative inquiry learning environment to facilitate students' collaborative argumentation with a focus on "tools for learning". Implications of the comparison for the design of external scripts are derived and directions for future research are discussed.

1. INTRODUCTION

Research on scaffolding tools has often adopted a technology-centered approach. Typically, individuals are provided with a technological tool and asked to perform a specific task, followed by measuring task performance as a function of using the tool or not (Pea, 2004). The personal development of the individual as well as changes of the context as a function of the interactions between the individual and the tool (i.e., an individual facing a new situation after having used a tool) are rarely subject to theorizing and research. Salomon (1990) described the latter instance as the *effects with* tools which stand in contrast to *effects of* tools, meaning the cognitive residuals that an individual holds after having interacted with a tool. These cognitive residuals then describe "learning" in a deeper sense. The aims of developing a technological tool that can support an individual's accomplishment of a task can be both to invoke effects of and effects with. To understand how an individual accomplishes a task, it is necessary to take into account the different factors that contribute to task accomplishment. These factors are comprised of the technological device, the target individual herself, as well as the context in which the individual uses a particular technology (cf. Stahl, 2002). To conceptualize the complex interplay between these factors, we refer to the term *script* since it has been used in all three disciplines that can contribute to solving this problem, namely psychology (Schank & Abelson, 1977), education (O'Donnell, 1999), and computer science (Ayala, this volume; Miao, Hoeksema, Hoppe, & Harrer, this volume). Although scripts are conceptualized differently in the three domains (see F. Fischer, Kollar, Mandl, & Haake, this volume), they share in common being seen as structures guiding sequences of activities. In other contexts, scripts are referred to as *checklists* (G. Fischer, Lemke, Mastaglio, & Morch, 1991). How these approaches differ is in the question of where this guiding structure resides (in the mind of an individual vs. in the mind of the designer of an externally provided script vs. in the design of an artifact). The basic aim of this article is to articulate a perspective of an individual accomplishing a particular task as being guided by (a) the *internal* scripts individuals are holding with respect to the target activity, (b) the *external* scripts that are provided in the external surround of the actor(s), and (c) an interplay between those internal and external scripts. We are analyzing two scenarios:

- In the first scenario, individuals with cognitive disabilities are provided with a Personal Digitial Assistant (PDA) prompting them in executing everyday tasks like taking the bus, which they would be unable to execute without the tool (Carmien, 2006b).
- In the second scenario, dyads of learners collaborating in a web-based inquiry learning environment are provided with a collaboration script guiding them in how to engage in argumentation (Kollar, F. Fischer, & Slotta, 2005), thereby getting learners to internalize parts of the collaboration script so that they can use the imposed strategy even when the collaboration script is not present.

The chosen scenarios point to a distinction between *tools for living* and *tools for learning* (Carmien, 2005). Tools for living are external artifacts that empower human beings to do things that they could not do by themselves without that individuals are required to internalize the knowledge residing in these artifacts (Engelbart, 1995; Norman, 1993); they support distributed cognition or distributed intelligence (Pea, 1993), i.e., they serve as artifacts that augment a person's capabilities within a specific task for which an internalization is not required or aimed at (e.g., a hand-held calculator). Tools for living can be tailored to specific tasks and to specific individuals. Tools for living do not change over time, remain a constant factor during task accomplishment and are rarely abandoned (Carmien, 2005). In contrast, tools for learning support people in learning a new skill or strategy with the objective that they will eventually become independent of the tool. Tools for learning often serve a *scaffolding* function (Pea, 2004) meaning that the strategy that is represented in the tool should be gradually internalized by the learners.

2. SCRIPTS FROM A DISTRIBUTED COGNITION PERSPECTIVE

In most traditional approaches, cognition has been seen as existing solely inside a person's head, and studies on cognition have often disregarded the physical and social surroundings in which cognition takes place. Gregory Bateson remarked that memory is half in the head and half in the world (Bateson, 1972). We live in a world of *distributed cognition* (Salomon, 1993; Hollan, Hutchins, & Kirsch, 2001; G. Fischer, 2003; Pea, 2004): the shopping list that "remembers" for us, the speedometer on our car, the position of the toggle on our light switch.

In his *person-plus-surround* conception, Perkins (1993) adopts a systemic view on cognition that goes beyond the individual actor: A system engaging in cognition usually consists of an individual (person-solo) and his immediate physical and social surround. This surround might include tools (such as hand-held calculators, spelling correctors, prompting systems, Mathematica software) as well as other persons (person-plus-surround), and the person-solo and its surround are standing in a complex interplay. To perform a task, it matters less *where* the needed knowledge is represented – what counts are the *access characteristics* of that knowledge, i.e., how easily the system consisting of a learner and the immediate social and artifactual surround can access the relevant knowledge. For example, a person might consider a hand-held calculator as harboring the necessary knowledge to compute 3532*32131, and estimate that using the hand calculator requires less effort than calculating mentally. A system can further be characterized as dependent on which of its components has the *executive function* with respect to the task being accomplished. By executive function, Perkins (1993) means the (meta-)cognitive control over the system's actions. For example, a French language book can take over the executive function for the system consisting of an individual learner and the book when it includes orders like "conjugate 'aller'". According to Perkins, transferring knowledge to an external tool is adequate if the tool only performs routine tasks that cost too much to internalize (e.g., some mathematical calculations). *Higher-order knowledge* (e.g., knowledge about argumentation), as opposed to knowledge about routine tasks, should however reside in the person-solo (or be internalized by the person-solo), and not be transferred to the surround in order to give the individual the opportunity to internalize this knowledge and to be able to transfer this knowledge to different upcoming situations. The person-solo should be able to access this knowledge in situations in which an external tool is not present, i.e., to hold accessibility of the relevant knowledge as high as possible for different situations.

To describe situations in which an individual together with an external artifact accomplishes a particular task, scripts in various forms come into play. Instructional psychology (e.g., O'Donnell & Dansereau, 1992) uses the term script to describe instructions providing individuals (mostly members of a group) with procedural information with respect to performing a specific task (e.g., a manual for creating a table in WORD). These scripts can for example be represented graphically in a computer-based learning environment or can be given by a teacher's oral instructions. Scripts are – at least when they are presented for the first time – located in the *external surround* of the individual, aiming at improving an individual's (or a group's) performance with respect to a specific task. Considering the term as used in cognitive psychology (Schank & Abelson, 1977), scripts can be seen from a *person-solo* perspective as well: Most people already possess knowledge guiding them how to act in specific situations and in performing a specific task before actually performing it. For example, to use a PDA properly, one needs to have prior experiences concerning how to scroll down a menu, open files, etc. In the following, we elaborate in depth the importance of scripts for an individual performing a particular task, first talking about scripts residing in the person-solo (internal scripts), then about scripts residing in the person-solo's surround (external scripts) and finally provide thoughts about their interplay.

2.1 Scripts residing in the person-solo: Internal scripts

From a person-solo perspective, the term script describes the knowledge and strategies that an individual possesses and which guides actions and understanding in a specific situation (see Kolodner, this volume). In cognitive psychology, "a script is a structure that describes appropriate sequences of events in a particular context. A script is made up of slots and requirements about what can fill those slots. The structure is an interconnected whole, and what is in one slot affects what can be in another" (Schank & Abelson, 1977, p. 41). Schank and Abelson (as well as Schank, 1999) use the term predominately with respect to rather well-defined situations, the knowledge about which is acquired through repeated experiences with similar situations and which can be assumed as being culturally shared to a certain extent (e.g., a "restaurant script"). However, they also introduced *personal scripts*, meaning personal knowledge and strategies that guide an actor in acting in a situation that perhaps only herself interprets in this specific way and which is not culturally shared. For example, person A might possess knowledge of how to attack other arguments by creating counterarguments, whereas person B holds knowledge guiding her in finding an integration of different viewpoints. Such personal scripts can be highly flexible – experiencing an impasse can quickly trigger a change in the sequence of the personal script so that a different sequence gets instantiated.

Referring to Schank and Abelson's (1977) notion of personal scripts, individuals may hold scripts for many situations they have experienced before. In our view, a script can be more or less flexible, well- or ill-defined depending on at least three conditions: (a) the stability of the previous experiences collected in similar previous situations, (b) the individual's abilities to abstract and generalize from these specific situations to similar new ones, and (c) the degree of structuredness or openness of the particular situation to rather situated actions and reactions. There can occur problems with an individual's internal scripts. First, internal scripts might not yet be well developed because the individual did not go through a specific situation often enough to develop an internal script already solid enough to prescribe a definite sequence of activities. This might be true for a middle school student who just started to learn algebra and has not yet developed an internal script concerning how to solve equations with two unknowns. Second, an internal script might not be adequately activated, perhaps because a person is performing two tasks simultaneously ending up with two scripts competing for too limited cognitive capacity. A third problem occurs when internal scripts are inaccessible or no longer accessible at all, as might be the case for people having had an accident that resulted in severe brain injury. In that case, internal scripts, for example for using public transportation, might not be accessible any more and can provide an opportunity for an external device designed to support an accomplishment of this task. A fourth problem could be that an internal script can be activated that does not fit current realities, for example a person with cognitive disabilities having activated the "board the express bus"-script but arriving at a bus stop that serves only local busses or a student creating a summary of a text when the actual task is to discuss strengths and weaknesses of the text. Fifth, in a collaborative learning scenario, collaborators might have activated inadequate or maybe too heterogeneous internal scripts which hamper interaction, collaboration, and in the end learning. For example, when learners have the task of understanding the concepts of velocity and gravity by manipulating a computer model of a ball (similar to the task that was used by Roschelle & Teasley, 1996), one learner might have activated a trial-and-error-like internal script, whereas the other learner might have activated an internal script that guides her in thinking about the concepts in a more theoretical sense.

Depending on the nature of the misfit of an internal script with respect to the external task, whatever of the five problems just described might have caused it, technology can help to provide external scripts to complement those deficient or inadequate internal ones.

2.2 Scripts residing in an individual's surround: External scripts

In contrast to cognitive psychology (Schank & Abelson, 1977), instructional psychology (O'Donnell & Dansereau, 1992) as well as computer science (Ayala, this volume; Carmien, 2006b) use the term script to describe guidelines in the surround of an individual or a group of individuals that provide procedural support for accomplishing a specific task or a class of tasks. *External* scripts can take on very different forms, i.e., they can be represented in very different styles and provide affordances for desired actions and constraints for undesired actions, and they can do so in an explicit or a more implicit manner (see Kollar, F. Fischer, & Hesse, in press). This broad definition allows us to discuss very different kinds of external scripts. We differentiate between scripts that are tools for living, i.e., scripts that were developed to help people in accomplishing everyday tasks like "riding a train", and scripts that are tools for learning that aim at encouraging learning processes on behalf of the users (Carmien, 2005). Using Perkins' (1993) terms, the main difference between these two approaches can be seen in the question whether the knowledge under consideration in these scripts is to be internalized by the learners or not. If this is the case, we are talking about a tool for learning, if not, the tool under consideration represents a tool for living.

In instructional psychology, much effort has been taken to develop scripts that are tools for learning. There, external scripts often provide rather clear procedural guidance. In the classical approach from O'Donnell and Dansereau (1992), for example, the script specifies that at first collaborators have to study a text individually, then one learning partner is playing the recaller while the other one adopts the role of a monitor pointing to omissions and mistakes in the recaller's summary, etc. Scripts can be viewed as inducing specific activities, which are to be shown in a certain sequence and which can be bound to certain roles. External scripts do not always have to be as constraint-based or prescriptive as the script developed by O'Donnell and Dansereau (1992). Other scripts rather provide *affordances* (Norman, 1993) for particular activities to be carried out by an individual without explicitly stating "Now do X", thereby being more permissive in nature. For example, scripts in inquiry-based learning environments tend to be rather open in that they afford very different activities to be conducted by the learners. Learners can engage in exploring information, in conducting experiments, in manipulating simulations, etc. What activities and what sequences of activities a learner is realizing depends on the structure of his internal script. It is this interplay between externally present or induced scripts and the individuals' internal scripts that is of interest in the next section.

2.3 Scripts in the person-plus-surround system: Interaction between internal and external scripts

We claim (1) that the design of an external script must take into account the internal scripts of the individuals that will be utilized to accomplish a specific task and (2) that it is not adequate to regard the interplay of internal and external scripts as a static relationship. Different individuals hold different internal scripts that can be complemented only by different external scripts, and in the case of scripts that are tools for learning, portions of the external script become more and more internalized by individuals, becoming encoded in their internal script with respect to perform a specific task. In the case of individuals with cognitive disabilities the internal scripts (innate abilities and skills) differ from the internal scripts of non-handicapped individuals in both content detail and in how to be best triggered externally. In the second, collaborative argumentation scenario we present later, two learners holding low-level internal scripts about how to engage in collaborative argumentation are guided by an external collaboration script to debate about the contents of a web-based collaborative inquiry learning environment. From a systemic perspective, the learners together with the external collaboration script form a person(s)-plus-system. As both learners repeatedly follow the rules of the external collaboration script, they might develop a more sophisticated internal script on how to perform this task. The executive function may shift gradually from the external collaboration script to the learners' personal cognitive systems, resulting in the artifact (the external collaboration script) becoming less and less important and learners being enabled to engage in fruitful discussions without being guided by an external collaboration script. Another perspective on the changing relationship of internal and external scripts and task support, is to acknowledge the changing environment and affordances that are available in pursuing the goal. As Suchman (1987) pointed out, the scripts required to attain the desired goal must change as the abilities and the environment change, and thus the external scripts must adapt to differing situations.

In the cognitive disabilities scenario, individuals lack the ability to detect similarities between a situation, in which an external script once helped and a similar new situation. Then, there is no opportunity to internalize contents from the external script, and no gradual shift of script information from the surround to the person-solo can occur. As a consequence, the external script has to remain active (e.g., can not be faded out) and accessible over time to support individuals in accomplishing the task again and again.

3. ANALYZING SCRIPTS FROM A DISTRIBUTED COGNITION PERSPECTIVE

We saw that different script types contribute to an individual accomplishing a specific task. However, a more systematic analysis of internal and external scripts and their interplay is needed. This analysis should focus on the different conceptual components scripts are made up of and try to delineate the interrelations between these components within and between internal and external scripts. Therefore, in this section we aim to extract the components of both internal and external scripts that are relevant to the models of distributed cognition and thereby draw on a model that was proposed by Kollar et al. (in press). A distributed cognition perspective is valuable, since it points to the importance of a person's internal script with respect to a particular task. We assume that accomplishing a task requires three factors: (a) the *activity* leading to task accomplishment, (b) *knowledge* underlying this activity, and (c) the *executive function*, i.e., the instance that chooses and controls how to conduct the activity and what knowledge to use in order to accomplish the task. Each of these three components can be broken down into several subcategories (see Table 17-1).

Table 17-1. Overview over the different script dimensions and sub-dimensions from a distributed cognition perspective.

Main dimension	Sub-dimensions
Activity	Goal
	Subactivities
	Sequencing
	Roles
Knowledge	Type of representation
	Locus of representation
	Accessibility characteristics
Executive function	Goal setting control
	Performance setting control

On behalf of the *activity*, we distinguish between four defining features. First, the activity can be described in terms of the *goal* it pursues. For example, a major goal might be "learning to drive". Second, these activities can include *subactivities* like "fastening seatbelts", "switching gears", etc. Third, these subactivities can be *sequenced* in a specific order. For the present example of "learning to drive", one sequence would be "getting into the car", "fastening seatbelts", "turning the ignition key", etc. Finally, a script can cluster activities to *roles*, for example a "driver" role or a "customer" role. These aspects can be evoked by the contents of both an internal, or an external script. Although we assume a certain equivalence with respect to functionality in a distributed cognition system (e.g., internal and external structures might both evoke specific cognitive processes), we do not assume a structural equivalence between internal and external scripts (cf. Cox, 1999).

With respect to *knowledge* that is underlying the performance of specific tasks, there can first be different *types of representation*. For example, (1) knowledge residing in an external script might be represented textually, like in a user's manual, or graphically like in a scaffold for assembling furniture, or (2) mentally in the cognitive system of a person. Second and in relation to this, there can be different *loci of representation* as well. In the case of internal scripts, knowledge is residing in the person-solo, whereas in the case of external scripts, knowledge is represented in the persons' surround. Often, the knowledge residing in an external script is supposed to become internalized by the individual interacting with it, so that the locus of representation thereby is gradually switching from external to internal (or from the surround to the person-solo). Knowledge necessary to perform a task can third be described in terms of its *accessibility characteristics*, hence different kinds and pieces of knowledge can be more or less easily accessible, which can have physical as well as cognitive reasons. For example, the information that 32*32 equals 1024 is highly accessible when an individual has a hand-held calculator at his or her disposal, whereas it is less accessible when she has to compute without this support.

With respect to the *executive function*, we differentiate two subcategories. First, scripts can be characterized with respect to who is setting and controlling the accomplishment of the intended goals (*goal setting control*). There might be instances in which an external person sets goals for an individual; in other situations, the individual is developing a script for herself, and in yet other situations an external tool sets the goals for the individual. Second, it is important to ask how it is assured that the specific individual in fact performs the activities and accomplishes the task she is supposed to perform (*performance control*). For example, technological tools can be designed in a way that they always give immediate feedback when the individual took the right steps and/or if performance was accurate. In other cases, it might be left to the individual to evaluate if she performed the activities correctly or not.

In the next section we use these categories to describe and analyze two scenarios in which we have explored external scripts that are suitable for specific types of individuals and specific tasks: (1) *The Memory Aiding Prompting System (MAPS;* Carmien, 2006a) is being developed in the context of the Cognitive Levers (CLever) project (Carmien, 2005; CLever, 2005) at the University of Colorado to provide external scripts for persons with cognitive disabilities, thereby representing a prototype of a tool for living; (2) *The collaborative argumentation script* for 8th to 10th graders, which was developed at the Knowledge Media Research Center in Tübingen (Kollar, et al., 2005), which can be viewed as a tool for learning.

4. EXAMPLES FOR AN INTERPLAY OF INTERNAL AND EXTERNAL SCRIPTS

4.1 Memory Aiding Prompting System (MAPS): A tool for living

Cognitively impaired individuals are often unable to live on their own because of deficiencies in memory, attention, and executive functions. These deficits can create an inability to consistently perform normal domestic tasks like cooking, taking medications, performing personal hygiene, and using public transportation. A common way of transitioning from assisted living to independent or semi-independent living is through the use of prompting systems. A prompting system decomposes a task into constituent parts, the parts comprising a script, and evoking each part with a prompt consisting of a pair of an image and a verbal instruction. MAPS (Carmien, 2002) consists of a mobile PDA based cellular phone (Figure 17-2) to be used by the person with the cognitive disability and a PC-based script editor tool (Figure 17-1) to be used by the caregiver to create scripts. At script design time the caregiver chooses appropriate images and verbal prompts and assembles them, using the MAPS script editor, into scripts, that can then be loaded into the hand held MAPS prompter. At use time the person with cognitive disabilities uses the multimedia prompts displayed on the hand held computer to cue internal scripts (Carmien, DePaula, Gorman, & Kintsch, 2003; Carmien, DePaula, Gorman, & Kintsch, 2005b).

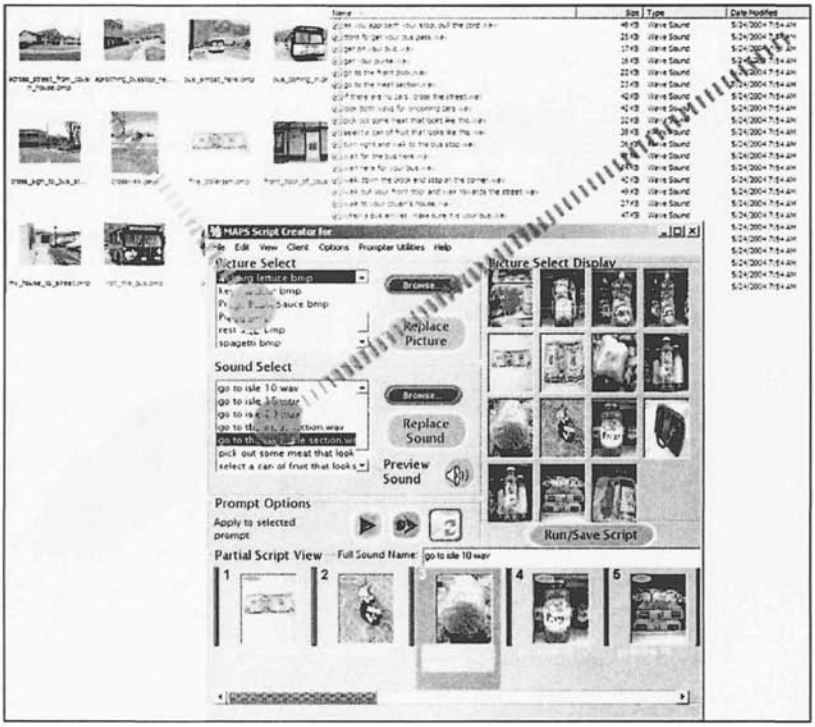


Figure 17-1. The MAPS Script Editor the upper left are the images in directories that may be inserted into the developing script, similarly the upper right shows a directory holding sound files to match with the images and make a prompt, a series of which for the script.

The MAPS script editor allows caregivers to easily create, store, and share scripts or prompts. MAPS implements multimedia prompting on its PDA platform by playing the sequence of pairs of images and vocal cues that step a user through a script to affect a task. Each prompt is an external script that triggers the stored/learned behavior of the users, their internal script. Additionally, the MAPS prompting system is designed to provide a learning tool to acquire skills and scaffolding for daily life. When used as a learning tool the repetition of the external scripts may cause the script itself to become an internal script; but for most, the MAPS prompter will be used as a tool for living. The target population for MAPS, is cognitively disabled individuals; using standard notation (The American Association on Mental Retardation, 1992) "trainable Mentally Handicapped' IQ 55-72 and the upper range of 'Severely Mentally Handicapped" IQ < 55. However diagnostic language does not describe the desired population as well as a list of needs and abilities: they cannot read and have significant memory and executive function deficiencies; they must be able to work well with prompting techniques; have social skills sufficient to use commercial establishments without problems; have fine enough motor coordination to use a PDA, and be sufficiently capable to not lose or damage a PDA.

The design of the caregivers' script editor reflecting a *meta-design perspective* (G. Fischer, 2004, 2006a) on design time and use time requirements provides a tool to non-programmers to create scripts, in effect creating small programs to be run on the MAPS handheld prompter (Figure 17-2: MAPS handheld prompter in use). Grounded in our distributed cognition framework, the computational environment allows users to operate within the band of optimal flow (Csikszentmihalyi, 1990). This is achieved by fitting the granularity of executive function cues, the elements of external scripts, to the existing internal scripts of the user with cognitive disabilities. By doing so we obtain the precise fit that does not "over-control" the user (many more cues than is necessary) nor "under-cue" the user (asking for tasks to be accomplished that the user can not achieve). In effect, MAPS mediates the collaboration between caregivers and persons with cognitive disabilities aimed towards more independence for the persons with cognitive disabilities, which benefits both stakeholders. Over time the MAPS logs (which reflect script usage and effectiveness) aid in refining this asynchronous process. MAPS additionally provides simple ways to backtrack or to start over, to allow for mistakes during task completion, and a *'panic button'*.

The MAPS user interface is twofold; the Graphical User Interface (GUI) for the user with cognitive disabilities and the GUI for the caregiver. Because the target population has a limited number of possible internal scripts, the set of available prompting scripts will not change dramatically, the same prompting scripts being used over and over. Thus, many prompting scripts can be constructed by reusing sub-scripts (i.e., the steps to "get from getting ready to go out to the closest bus stop"). What will change is the timing and, to a small degree, the content of the scripts. MAPS is equipped with GPS and wireless networking so that, for example, when users get to their bus stop, a specific bus coming in will trigger the prompt to get on the bus (Sullivan & G. Fischer, 2003).

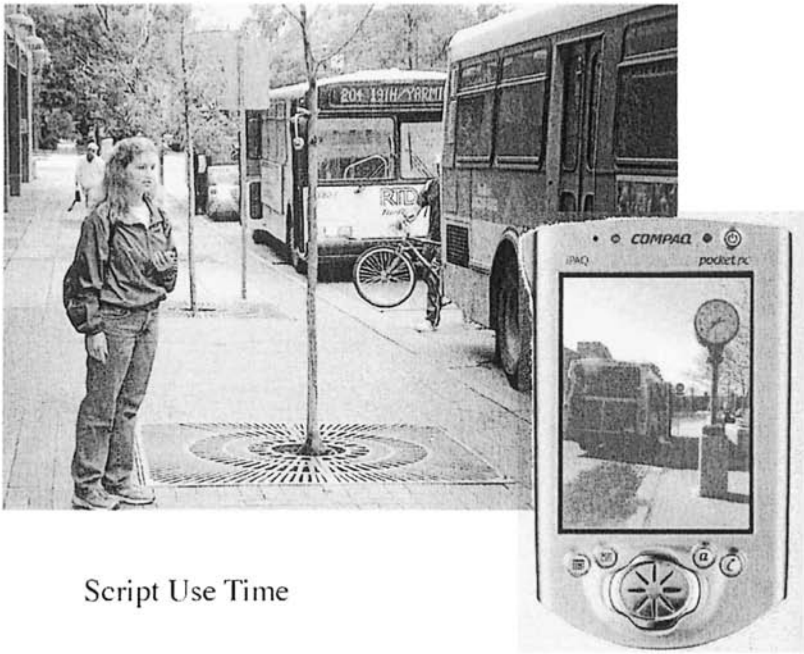


Figure 17-2. MAPS prompter guiding a user in boarding the correct bus. The middle image shows a user holding a MAPS prompter and being aided in selecting the correct bus. The lower image shows the correct bus on the MAPS prompter.

4.2 Collaborative argumentation script: A tool for learning

Kollar et al. (2005) developed a script aiming at improving high-school students' collaborative argumentation in a web-based collaborative inquiry learning environment. Background of this work was that students' collaborative argumentation often appears to be deficient, i.e., they have low-level internal scripts: they often have difficulties in generating well-grounded arguments (Toulmin, 1958), and they rarely generate longer argumentation sequences, which may contribute significantly to collaborative knowledge construction ("argument – counterargument – integrative argument"; Leitão, 2000). In order to address these problems, the authors developed an external script that was supposed to alleviate the construction of complete arguments and longer argumentation sequences and implemented it into a curriculum project of a web-based collaborative inquiry learning environment, namely the Web-based Inquiry Science Environment "WISE" (Slotta, 2004). In the

WISE curriculum project ("The Deformed Frogs Mystery"), dyads of learners learned that many frogs with physical deformities were found during the late 90's, and that biologists are discussing two hypotheses that might account for the problem. One hypothesis states that a parasite causes the deformities, whereas the other hypothesis states that there is an environmental-chemical substance in the water, which causes legs to develop strangely. The learners' task was to discuss and evaluate the two hypotheses against the background of various information they could explore in the learning environment (e.g., maps to see how the deformities are distributed, photographs of deformed frogs, journal articles about the phenomenon, etc.). The curriculum project included five content-specific parts (e.g., "The Problem", "Where are the frog deformities?", "What's in the water"), at the end of which the external script was implemented. Screenshots of a translated version of the script can be seen in Figure 17-3 and Figure 17-4.

When learners first clicked on the button with the inscription "Discuss the parasite hypothesis" (left hand side of the screen in Figure 17-3), they received general instruction concerning the way they were supposed to structure their argumentation, specifying that at first there should be an argument, then a counterargument, and then an integrative argument. Moreover, it was prescribed that each of these arguments was supposed to include data, a claim and a reason (see Figure 17-3). When learners then scrolled further down, several empty textboxes appeared, for each of which it was specified, who should fill them in and what argument component should be generated (see Figure 17-4). For example, for the first three textboxes, it was specified that learner A had to formulate her argument (in favor of the parasite hypothesis), typing the data she was referring to in the first textbox, the claim she wanted to make in the second one, and the reason specifying the relation between data and claim in the third box. During this time, learning partner B had to monitor whether the argument A was producing was complete or not. The next three textboxes were prestructured analogically, this time demanding B to develop a counterargument, and A monitoring the completeness of the counterargument. In the end, both partners had to generate an integrative argument and both had to monitor whether their argument was complete. In order to assure the correct application of the script instructions, for the first time learners were completing the task to generate an argumentation sequence, the textboxes always were headlined with sentence starters (e.g., "It has been found that...").

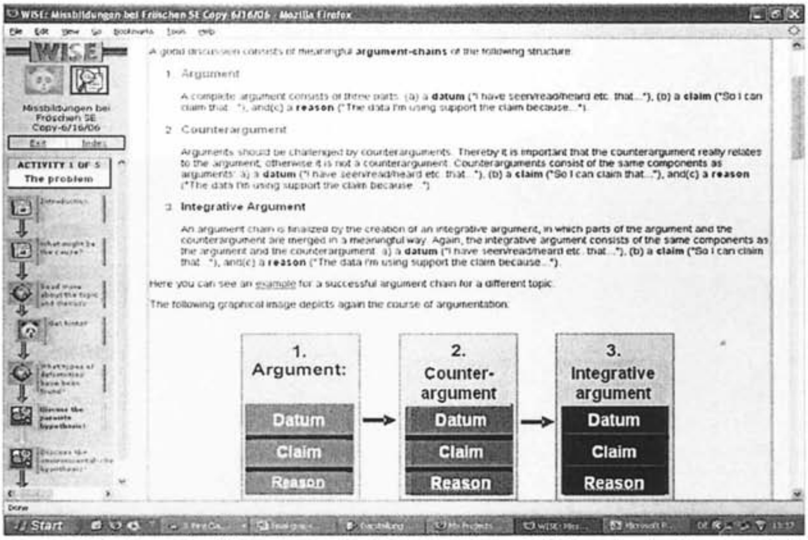


Figure 17-3. External collaboration script. Left hand frame: WISE navigation buttons. Right hand frame: Instructional text and graphical image depicting guidelines for collaborative argumentation (translated into English).

As learners proceeded more and more through the learning environment, the instructional support provided by the external script continuously faded out, expecting a gradual internalization of the strategic knowledge provided in the external script. For example, in the end of the second part of the curriculum project, learners received only three textboxes (one per argument) without the sentence starters just described, in the end of the third part only one text box for creating a whole argumentation sequence, and finally only one text box for discussing both hypotheses. In order to avoid biased information processing for one or the other of the two hypotheses, roles concerning who had to advocate which hypothesis were switched several times during the learning process.

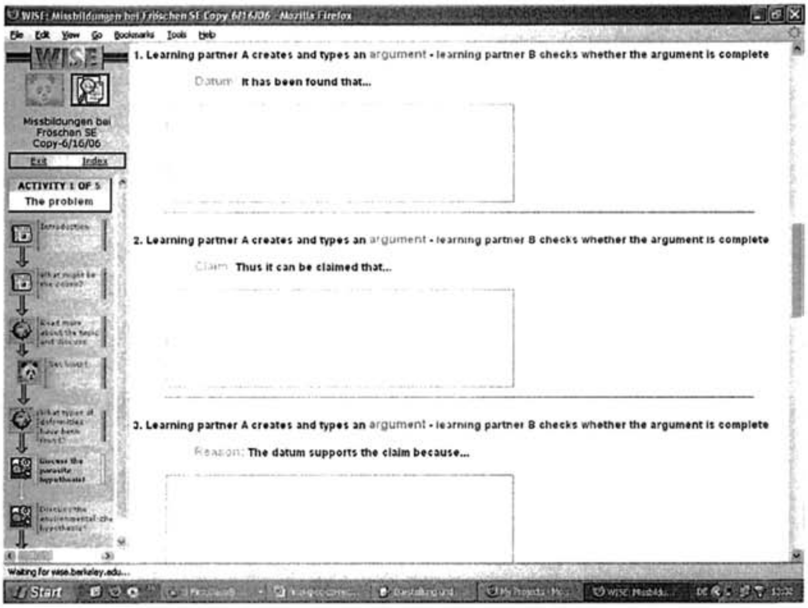


Figure 17-4. Collaborative argumentation script: blank text boxes with script prompts and role assignments (translated into English).

5. INTERNAL AND EXTERNAL SCRIPTS IN A TOOL FOR LIVING AND A TOOL FOR LEARNING SCENARIO

The contribution of this chapter is to describe and analyze situations, in which a person X is asked to perform a task Y from a distributed cognition perspective. We argued that during the process of task accomplishment, both internal and external scripts are important. In the following, we are using the categories described in section 3 to analyze and compare MAPS as an example for a tool for living and the collaborative argumentation script as an example for a tool for learning.

Activity Dimension. Concerning the *activity* level, there are both similarities and differences between the approaches. Although MAPS is a *device* to augment intelligence (Engelbart, 1995) and to change tasks (Norman, 1993), and the collaborative argumentation script is a *tool* to augment intelligence, the major goals of the two activities are rather distinct: In MAPS, the aim of the activity is the accomplishment of an everyday task like "using public transportation", whereas in the collaborative argumentation script, the goal is to engage dyads in learning about biology and argumentation. In both approaches, the goal of the activity is accomplished by conducting a variety of sub-activities that are externally induced in the target individual(s) ("walk to bus stop" or "leave bus here" in MAPS; "give data for your argument" or "formulate a claim for your counterargument" in the collaborative argumentation script) and that are bundled to specific types of roles ("customer" in MAPS; "advocate for parasite hypothesis" and "advocate for environmental-chemical hypothesis" in the collaborative argumentation script). At the start of a scripted session the external scripts of both systems provide a rather clear sequence concerning when to engage in which sub-activity as well as a clear description of what role the target individual is supposed to take on. As task accomplishment progresses, sequencing as well as the strictness of role assignment in the two approaches develop differently. While in MAPS, strict sequencing of activities is realized throughout the whole task performance, in the collaborative argumentation script sequencing features are faded out over time, meaning that individuals in the end can define their own sequence according to which they want to build arguments and argumentation sequences. Further, in MAPS, the target individual stays in his or her customer-role until the end of task accomplishment, while the collaborative argumentation script provides learners with growing degrees of freedom to choose which role (if any) they want to take on (e.g., monitorer or arguer-role).

Knowledge Dimension. With respect to the knowledge dimension, in the MAPS approach the knowledge necessary to accomplish the target task of "take bus to recreation center" is represented in graphical images on the screen of the PDA the target individual is carrying. To properly trigger the appropriate internal script, the target individual must build up an internal representation of the object that is presented on the PDA at a particular point in time. This internal representation may be likely to vanish in a short period of time, so the main *type of representation* remains graphical until the task has been accomplished, thus being prescriptive in a sense that the user is constantly reminded not to deviate from the activity portrayed by the external representation on the screen. In case of the collaborative argumentation script, however, at the beginning of the learning situation the knowledge necessary to engage in argumentation is represented in multiple forms, namely textually and graphically. As learners interact more and more on the basis of the external script, an internalization process is intended to occur, which results in the development of a gradually more stable mental representation (however, learners may already possess internal scripts that comply with the external script instructions prior to the collaborative learning situation). With the fading of the external script instructions, textual and graphical representations become less visible. Further, an individual can also access the learning partner's representations, i.e., she can ask the partner about how to proceed to accomplish the task, thereby receiving auditively coded sequential representations of the information. While in MAPS, the *locus of representation* is, from the beginning of the episode, strongly external, in the collaborative argumentation script it shifts more and more from *strongly external* to *strongly internal*. These differences in the two approaches can be attributed to the different notion of what kind of internal scripts the target individuals are holding. In MAPS, due to the end user's cognitive disabilities, the internal scripts are more static and less developable than is the case for the target individuals of the collaborative argumentation script. Through constant interaction with the collaborative argumentation script, learners get to internalize relevant portions, having the effect that the induced processes are being controlled by their internal scripts that are gradually improving and enabling them to lead better discussions in the future. The underlying assumption is that an optimal fit between internal and external scripts might be reached by internal scripts becoming more and external scripts becoming less sophisticated over time, i.e., the more the learners internalize through following the script instructions, the less specific the script instruction have to become. That way, the external script becomes more and more a prompting system for the activation of internal scripts. For Pea (2004), such fading is essential for a scaffold like an external collaboration script to be called a scaffold and thereby what can be called a tool for learning. If not faded, the external script would rather be an example for distributed intelligence, meaning that users would not necessarily have to learn what the script induces but rather use it as a tool for living that is constantly accessible. The *accessibility characteristics* of knowledge residing in the two script approaches are assumed to remain stable. In fact, one main aim of every scripting approach must be to hold accessibility of task-relevant knowledge high. In the MAPS approach, accessibility can only be guaranteed by locating knowledge in the external surround, due to target individuals not being able to build up an internal representation of relevant knowledge. In contrast, in the collaborative argumentation script approach, at the beginning of a learning episode relevant knowledge is made accessible in learners' external surround in a graphical and textual manner, but by constantly and intentionally using the external scripts, knowledge is becoming easily accessible in the person-solo, which is accounted for by making the external script continuously less accessible (via fading). However, it is likely that some learners might require having the external script longer externally accessible than it is the case here, because learners will differ in the amount of time they need to internalize the contents of the external script.

Executive Function Dimension. With respect to the executive function, MAPS and the collaborative argumentation script exhibit differences: Concerning *goal setting control*, in the MAPS approach there are at least two kinds of persons involved: the designer of the MAPS environment and the caregiver who designs the script for the particular needs of the target individual, making as efficiently as possible use of the design constraints set by the environment designer. Such a collaborative effort between several persons is not present in the collaborative argumentation script approach – there, it is solely the designer(s) of the external script who set(s) the goals for the target individuals. However, in both cases, the target individuals themselves have personal goals, which sometimes comply with the goals that are set externally. *Performance control*, in the collaborative argumentation script, is transferred in part to the learning partner, who is not supposed to generate an argument but to monitor whether the argument his or her partner is developing is complete in the sense of Toulmin's (1958) argument scheme. The interface design includes some low-level performance control that can sense whether one or more textboxes remained blank and then asks learners whether they want to go on anyway. In both script scenarios, the target individuals themselves could engage in performance control to a certain extent. A basic assumption of the MAPS approach is that the target person's cognitive disabilities are not so severe that they would not allow her to realize that something has gone wrong, so she can press the "panic button" informing the caregiver that help is needed. In a similar vein, learners in the collaborative argumentation script approach are expected to be able to monitor whether the external script is being followed by them or not.

6. CONCLUSIONS

In this chapter, we explored the value of a distributed cognition perspective on scripts for different situations. We illustrated this by using two examples:

- MAPS, a socio-technical environment creating external scripts representing *tools for living* by supporting people with cognitive disabilities in accomplishing everyday tasks like "using public transportation", and
- the collaborative argumentation script representing a *tool for learning* for supporting high school students in acquiring argumentation skills.

A distributed cognition perspective can be used to describe and analyze both – situations, in which scripts help in genuine living tasks as well as situations, in which external scripts are explicitly designed to facilitate learning. The provided conceptual framework is simultaneously broad enough to describe scripts from different backgrounds and to capture conceptual differences between scripts as tools for living and scripts as tools for learning.

Adopting a distributed cognition perspective can give new insights into how external scripts should be designed for better task accomplishment and thereby better learning. As we noted at the beginning of this article, the development of scaffolding tools is often focused on their design processes or their usage and simultaneously puts less attention on internal consequences of this usage. It has largely been ignored that different individuals hold different internal scripts (a fundamental challenge addressed by the CLever project; Carmien, 2005) that might require differently structured external scripts and that this interplay between internal and external scripts can change over time when an individual gradually represents contents internally that were originally represented in the external script. Different target persons and different prerequisites of an individual interacting with an external script might require this external script to be sometimes prescriptive and sometimes permissive. In MAPS, the external script needs to be prescriptive because of (a) the low reliability of individuals' internal scripts and (b) their restricted ability to internalize relevant information that is located in the external script. In the collaborative argumentation script, the external script becomes more permissive the longer dyads are interacting with it, i.e., by giving learners increasingly more degrees of freedom (after a while), they are provided with the opportunity to let their improved internal scripts guide their argumentation in a less restricted surround.

We presented two prototypical examples for a tool for living (MAPS) and a tool for learning (collaborative argumentation script). The main difference between the two is that tools for living are designed to augment intelligence and change tasks (G. Fischer, 2006b) by being continuously accessible in the surround of a person-solo and tools for learning representing a way of supporting learners to acquire new skills and knowledge (Pea, 2004). As a consequence, one main component of tools for learning is that they include fading mechanisms so that learners have the opportunity to practice the learned skills without external support being available. In the case of tools for living, such fading is not necessary, since there is no internalization intended. Defining (and designing) an external script as a tool for living or a tool for learning depends on user characteristics. If users do not have the capability to internalize external script contents, the script represents a tool for living – accordingly, it should remain stable in the external surround of the users. In contrast, if users do have the chance (and maybe even the task) to internalize the strategies that are imposed by the external script, it is a tool for learning.

One further potential of adopting a distributed cognition perspective on scripts is that it points to the relevance of the *accessibility characteristics* of scripts and script portions. It is clear that accessibility of script information should be high throughout an individual accomplishing the target task. But how does accessibility change through internalization and fading? How long and in what ways do users have to interact so that the script portions are as accessible in the person-solo as they were before when the script was represented externally? These are questions that are up to future research.

Earlier, we said that three domains are particularly concerned with scripts, as it is also represented in the structure of this book: computer science, cognitive psychology, and education. For each of these disciplines, specific challenges can be derived from our analysis.

In *computer science*, an important challenge for designers of software to be used by specific types of users is to create a design that accounts for the customers' needs in the best possible way, including *user-centered* (Norman, 1986) and *participatory design* approaches (Schuler, 1993). Most of the times, user groups are very heterogeneous with respect to important aspects like their prior knowledge about how to interact with a specific class of computer programs, thereby making it difficult to realize a high fit between software design and user needs. Very often, this problem is accounted for by providing specific help systems a user can access when experiencing a problem as well as including diverse opportunities for preference settings a user can individually design. This is what is called *meta-design* (G. Fischer, 2004, 2006a): a conceptual framework for socio-technical systems in which end-users (not only professional software developers) can create external scripts. MAPS is an environment supporting meta-design in which caregivers (knowing the internal scripts of the person with cognitive disabilities) can create external scripts fitting an individual. Meta-design is an important challenge computer science is facing to develop highly usable external scripts.

For *cognitive psychology*, an important challenge is to get a clearer picture of how different external scripts affect acting and thinking in particular situations and if and how they can change individuals' internal scripts with respect to these situations. Thereby, external scripts with respect to one person can have their origin in another person, and it might be interesting to see how the internal scripts of two persons are influencing each other. For example, Rummel and Spada (this volume) investigate how two individuals with different background knowledge (a psychologist and a medical doctor) collaborate in solving a psychological-medical case that requires a coordination of both individuals' internal scripts. Likewise, Runde, Bromme, and Jucks (this volume) analyze collaboration processes between experts and laypersons, in which the internal scripts of the interaction partners have to be coordinated to come to a satisfying problem solution. It is an interesting question whether and how components of the two internal scripts are transferred and what determines this transfer process.

For *education*, one main challenge is to investigate how the different script types that are distributed over a classroom can be used and instructionally designed in a way that learners are engaged in meaningful learning processes yielding significant learning outcomes. First, individuals can be conceptualized as holding internal scripts that guide them concerning how to engage in particular classroom activities. Second, a computer program can provide an external script guiding learners to process the specific classroom activities in a specific way. Third, the teacher can be conceived as holding a teaching script that is external to the learners and that influences the way learners are accomplishing the classroom activity. It is the question how these different script types can be orchestrated on a classroom level in order to realize productive learning. A distributed cognition model including different levels of regulation (e.g., Cole & Engeström, 1993; Dillenbourg & Jermann, this volume) seems highly valuable here to guide research and to help accumulate scientific knowledge appropriately in this respect.

ACKNOWLEDGEMENTS

The creation of this chapter was supported by the German-American Network in the Field of Technology-Supported Education, which was collaboratively funded by the Deutsche Forschungsgemeinschaft (DFG) and the National Science Foundation (NSF). The network helped us to bring together thoughts and ideas about scripting that were separately generated at the University of Colorado in Boulder and the Knowledge Media Research Center in Tübingen and merge them to develop an interdisciplinary perspective on scripting.

REFERENCES

Bateson, G. (1972). *Steps to an Ecology of Mind*, Chandler Publishing Company.

Carmien, S. (2002). *MAPS: PDA scaffolding for independence for persons with cognitive impairments*. Presentation at HCIC Conference 2002. Retrieved June 21, 2006, from http://www.cs.colorado.edu/~carmien/MAPS_boaster.pdf

Carmien, S. (2006a) *MAPS Website*, Retrieved June 31, 2006, from http://www.cs.colorado.edu/~l3d/clever/projects/maps.html

Carmien, S. (2006b) Moving the Fulcrum: Socio-Technical Environments Supporting Distributed Cognition for Persons with Cognitive Disabilities. Doctoral thesis. Univerity of Colorado, Boulder.

Carmien, S., Dawe, M., Fischer, G., Gorman, A., Kintsch, A., & Sullivan, J. F. (2005) "Socio-Technical Environments Supporting People with Cognitive Disabilities Using Public Transportation," Transactions on Human-Computer Interaction (ToCHI), 12(2), pp. 233-262.

Carmien, S., DePaula, R., Gorman, A., & Kintsch, A. (2003). *Increasing Workplace Independence for People with Cognitive Disabilities by Leveraging Distributed Cognition among Caregivers and Clients, ACM 2003 International Conference on Supporting Group Work (GROUP '03),* pp. 95 - 104, Sanibel Island, Florida, USA.

Carmien, S., DePaula, R., Gorman, A., & Kintsch, A. (2005b). Increasing Workplace Independence for People with Cognitive Disabilities by Leveraging Distributed Cognition among Caregivers and Clients, Computer Supported Cooperative Work (CSCW) - The Journal of Collaborative Computing, 13, 443-470.

Carmien, S., & Fischer, G. (2005) "Tools for Living and Tools for Learning," *11th International Conference on Human-Computer Interaction*, Las Vegas, CD Rom.

CLever (2005) CLever: Cognitive Levers -- Helping People Help Themselves. Retrieved from http://l3d.cs.colorado.edu/clever/

Cole, M., & Engeström, Y. (1993). A cultural-historical approach to distributed cognition. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations* (pp. 1-46). Cambridge, UK: Cambridge University Press.

Cox, R. (1999). Representation construction, externalised cognition and individual differences. *Learning and Instruction, 9*, 343-363.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience.* New York: HarperCollins Publishers.

Engelbart, D. C. (1995). Toward Augmenting the Human Intellect and Boosting Our Collective IQ. *Communications of the ACM, 38*(8), pp. 30-33.

Fischer, G. (2003). Distributed cognition: A conceptual framework for design-for-all. In C. Stephanidis (Ed.), *Proceedings of HCI International 2003, Vol. 4* (pp. 78-82). Mahwah, NJ: Lawrence Erlbaum Associates.

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A. G., & Mehandjiev, N. (2004). Meta-Design: A Manifesto for End-User Development. *Communications of the ACM, 47*(9), 33-37.

Fischer, G., & Giaccardi, E. (2006a). Meta-design: a framework for the future of end user development. In H. Lieberman, F. Paternò, & V. Wulf (Eds.), *End user development – empowering people to flexibly employ advanced information and communication technology.,* Dordrecht: Kluwer Academic Publishers. (in press)

Fischer, G. (2006b). Distributed Intelligence: Extending the Power of the Unaided, Individual Human Mind. In Proceedings of Advanced Visual Interfaces (AVI) Conference, Venice, May 23-26, 2006, (in press).

Hollan, J., Hutchins, E., & Kirsch, D. (2001). Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research. In J. M. Carroll (Ed.), *Human-Computer Interaction in the New Millennium* (pp. 75-94). New York: ACM Press.

Kollar, I., Fischer, F., & Hesse, F. W. (in press). Collaboration scripts - a conceptual analysis. *Educational Psychology Review.*

Kollar, I., Fischer, F., & Slotta, J. D. (2005). Internal and external collaboration scripts in webbased science learning at schools. In T. Koschmann, D. Suthers, & T.-W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years* (pp. 331-340). Mahwah, NJ: Lawrence Erlbaum Associates.

Leitão, S. (2000). The potential of argument in knowledge building. *Human Development, 43*, 332-360.

Norman, D. A. (1993). *Things that make us smart.* Reading, MA: Addison-Wesley Publishing Company.

Norman, D. A., & Draper, S. W. (Eds.). (1986). *User-Centered System Design, New Perspectives on Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

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

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

Pea, R. D. (1993). Practices of distributed intelligence and design for education. In G. Salomon (Ed.), *Distributed cognitions: psychological and educational considerations* (pp. 47-87). Cambridge, UK: Cambridge University Press.

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.

Perkins, D. N. (1993). Person-plus: a distributed view of thinking and learning. In G. Salomon (Ed.), *Distributed cognitions: psychological and educational considerations* (pp. 88-110). Cambridge, UK: Cambridge University Press.

Salomon, G. (1990). Cognitive effects with and of computer technology. *Communication Research, 17*(1), 26-44.

Salomon, G. (1993). No distribution without individuals' cognition: a dynamic interactional view. In G. Salomon (Ed.), *Distributed cognitions: psychological and educational considerations* (pp. 110-138). Cambridge: Cambridge University Press.

Schank, R. C. (2002). *Designing world-class e-learning.* New York: McGraw-Hill.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding.* Hillsdale, NJ: Erlbaum.

Schuler, D., & Namioka, A. (Eds.). (1993). *Participatory Design: Principles and Practices.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Slotta, J. D. (2004). Web-Based Inquiry Science Environment. In M. C. Linn, E. A. Davis, & P. Bell (Eds.). *Internet Environments for Science Education* (pp. 203-231). Mahwah, NJ: Lawrence Erlbaum Associates.

Stahl, G. (2002). Contributions to a theoretical framework for CSCL. In G. Stahl (Ed.). *Computer support for collaborative learning: Foundations for a CSCL community. Proceedings of CSCL 2002* (pp. 62-71). Hillsdale: Lawrence Erlbaum Associates.

Suchman, L. A. (1987). *Plans and Situated Actions.* Cambridge, UK: Cambridge University Press.

Sullivan, J., & Fischer, G. (2003). *Mobile Architectures and Prototypes to Assist Persons with Cognitive Disabilities using Public Transportation, 26th International Conference on Technology and Rehabilitation,* Atlanta GA, USA. (CD archive).

The American Association on Mental Retardation (1992). *AAMR definition of Mental Retardation.*

Toulmin, S. (1958). *The uses of argument.* Cambridge, UK: Cambridge University Press.

Chapter 18 – Discussion

SCRIPTING GROUP COGNITION

The Problem of Guiding Situated Collaboration

Gerry Stahl
Drexel University, Philadelphia

Abstract: The concept of scripts has considerable appeal as addressing or at least naming an urgent issue in CSCL: how to use the promise of networked computers to guide groups of students to engage in desirable and successful collaborative learning. However, the concept of scripts is often applied inconsistently or founded on problematic theoretical grounds. Reconceptualizing scripts as situated resources rather than implementable plans for action is therefore undertaken here to align the concept with current socio-cultural thought. Studying how such a resource is made sense of in detailed interactions is then recommended for studying how scripts can be designed to guide situated collaboration.

1. INTRODUCTION

The term *script* encapsulates many connotations. This grants it the power to bring diverse topics together to cross-fertilize each other, as has been done in this book. At the same time, the term's overloaded meanings threaten to dull its focus and emasculate its power; if it conjures up different visions for each reader, the term loses its power to build *shared* meaning.

The publication of this multi-perspective and trans-disciplinary book on scripting in CSCL reflects an important joining together of researchers under the banner of the term *script* to delineate a major contemporary movement within a field that often suffers from feelings of theoretical and methodological fragmentation. Perhaps a useful role for a discussant in trying to support this convergence is to highlight its central claims, trace its historical roots and clarify its foundations.

This chapter will proceed by commenting on the senses of the term *script* that can be associated with several of the theoretical sources referenced in Chapters 16 and 17: Schank and Abelson (1977), Vygotsky (1930/1978), Suchman (1987) and Schwartz (1995). In reviewing this history, the chapter will define a view of scripts that may differ from the term's commonly understood sense. It will then conclude by revisiting central claims of Chapters 16 and 17 in terms of this refined view.

2. SCRIPTS AS COGNITIVE MODELS

The script metaphor has its commonsense roots in the theater. Actors follow a script, which defines the narrative context, roles, actions and outcomes of a play, movie or television drama. Although the public idolizes the actors and remains ignorant of the script designers, the real agency lies in the script, not in the pretty faces that mouth it. The play's intelligence is that of the author, put into word and onto paper, reified and made persistent so that it can control the action that may later take place on camera, in the author's absence, for the benefit of a projected audience at yet another time and place.

Pop sociology would have us all playing socially defined roles. Somehow, conventions of our culture define what everyone (present company perhaps excluded) does, says and thinks. When we enter a restaurant, we supposedly slip into the customer role and interact with the person in the waitress role according to a well-defined script.

This is not quite the sense of script that Schank and Abelson's *Scripts, Plans, Goals and Understanding* (1977) proposed. In their pioneering contribution to artificial intelligence (AI) and cognitive science, they were exploring a computational model of how people understand stories. They proposed that people organize their memories of how events like visits to restaurants proceed by constructing data structures that represent knowledge of generalized events and connections among events, like causal relations. This theory of scripts is quite complex, attempting to incorporate much domain knowledge as well as linguistic structure. It is specifically designed to account for our ability to make sense of stories by speculating about mental representations of commonsense knowledge that allow us to fill in the implicit relationships between consecutive narrative utterances.

Written in the heyday of rationalist AI research, Schank and Abelson's concept of scripts assumed that human minds worked like computer programs, accessing data structures and drawing long sequences of logical conclusions. Motivated by toy problems like analyzing artificially simple narratives about restaurant visits, such theories have not stood up well to subsequent reflection, especially when people try to extend the theory beyond its original restricted domain of understanding stories to human activity more generally.

The restaurant script, with its necessarily large collection of associated variations, sub-scripts and related scripts might help one to analyze restaurant visits in stereotyped television plots or in boring visits to the local diner. But these are not necessarily events worth writing about. A story needs to have an element of novelty or interest – precisely something that goes outside of the generalized script. And every actual restaurant visit involves spontaneous human interactions that improvise around the assumed roles with personality, humor and humanity.

There is also the theoretical question of whether we really walk around with these huge, detailed, logically organized data structures covering all our commonsense, social and personal knowledge. It may be more reasonable to imagine that we *construct* on the spot generalized versions of something like restaurant scripts as spontaneous resources for thinking about specific stories or events as they confront us. This is not the way computers were programmed to organize knowledge in the 1970s, but it seems plausible given the way stories are actually told to people, at least in face-to-face situations. A story is designed by the teller to interact with the audience (Livingston, 1995). The teller continually adjusts the telling to form a desired interaction with the recipient of the story. Through subtleties of gaze, intonation, body position, facial expression, gesture, rhythm and word choice, the narrator and the recipients maintain an intimate alignment that ensures moment by moment that the story is actually being shared. Assumptions of what each other hold to be generalized patterns of, for instance, restaurant behaviors, may play significant roles in this dance of shared meaning-making.

3. SCRIPTS AS SOCIAL RESOURCES

The notion that we should look at the details of interactions among people in groups rather than speculating about mental representations in individual minds in order to understand human knowledge was developed in Vygotsky's *Mind in Society* (1930/1978). Inspired by a deep grasp of Marx's (1867/1976) social philosophy around the time of the Russian revolution, Vygotsky argued on theoretical and empirical grounds that what is distinctive about the way that people learn is the construction of new skills in interactions with others within cultural contexts: "Human learning presupposes a specific social nature and a process by which children grow into the intellectual life of those around them" (p. 88).

Vygotsky's concept of the zone of proximal development distinguishes a person's intellectual abilities when working alone from those when collaborating with others. The fact that learners have significantly higher skill levels when working in dyads or small groups suggests that intellectual development generally takes place during interactions with others. Vygotsky was able to show with controlled experiments that children could accomplish tasks with external memory aids and with collaboration that they could not do on their own. Older subjects could achieve these tasks on their own, suggesting that they had somehow internalized the intersubjective or environmental aids in the intervening years. Vygotsky was not able to study the detailed interactions whereby collaboration and external artifacts were used, let alone observe directly the mechanisms of internalization. However, his visionary – if sketchy – theories inspired the emphasis on collaborative learning in socio-cultural contexts within CSCL.

Vygotsky's theory of learning suggests that scripts not be taken as models of mental representations of individual learners, but be used for structuring social environments to foster collaborative interactions that can engender intersubjective knowledge building.

4. SCRIPTS AS COMPUTER-BASED RESOURCES

A methodology for studying the moment-to-moment interactions of dyads and small groups engaged in collaborative problem solving – with computer support – is motivated, described and illustrated in Suchman's *Plans and Situated Actions* (1987). The use of video analysis based on principles of ethnomethodology (Garfinkel, 1967) as practiced by conversation analysis (Sacks, 1992), allows Suchman to propose an approach that she explicitly contrasts with the AI approach of Schank and Abelson: "Instead of looking for a structure that is invariant across situations, we look for the processes whereby particular, uniquely constituted circumstances are systematically interpreted so as to render meaning shared and action accountably rational. Structure, on this view, is an emergent property of situated action" (p 67). For instance, structures of meaning, goals, roles or turn-taking in conversation are not pre-existing structures, but are constructed interactively by the on-going discourse itself (Garfinkel & Sacks, 1970; Sacks, Schegloff, & Jefferson, 1974).

For Suchman, plans such as the scripts of Schank and Abelson are not rigid blueprints for action that are simply implemented as stated, but are flexible resources that people construct, interpret, adapt and use in their specific, situated acts of making sense. People's commonsense understandings of their plans may be similar to the AI view, but if one studies closely the role that plans play in actual activities – such as accomplishing office tasks – one gets a different view. In Vygotsky's (1867/1976, e.g., pp. 28f) analysis, planning skills evolved out of resources for interpersonal interaction. Young children simply act and then may retroactively give a name to their action (e.g., to a drawing they did, when prompted for a description). Later, they verbalize actions to be taken: at first in an attempt to control another person's behavior (e.g., their caretaker), and subsequently to control their own future behavior. In such ways, verbalizations of action (plans) can function either before or after the actions as ways of making shared sense of the actions.

In Suchman's ethnomethodological terms, plans are resources that may be used to prepare for and guide up-coming actions or to give an accounting of on-going or completed actions (i.e., they are often retroactive rationalizations). Under this analysis, plans are not causal agents of the action, but are possible useful accompaniments to the action that play (at least originally) a largely interpersonal role rather than an individual mental function. The social functioning of verbal plans (or their silently internalized derivatives in thought) is hidden in the taken-for-granted everyday functioning of human existence, and plans are then conceptualized based on their adult, conscious appearances. Commonsense folk theories – and the rationalist abstractions of these theories in AI – project plans into mental representations that cause planned action.

Suchman studies the use of a computer-based help system for a sophisticated copying machine. The help system defines an AI-type script that was designed on assumptions about mental models of scripts in users' heads controlling their actions. Suchman documents the failure of this approach by showing how dyads of users negotiate their understandings of various problematic states of their copying tasks through interactively trying to make sense of various resources in their environment, including messages from the copier, their shared discourse, verbalizations of their goals, generalizations of past experiences and attempts at various actions.

The fundamental problem, as Suchman points out, is an asymmetry in the data that the copier computer has about the on-going work context and what the users understand about the situation. This asymmetry is closely related to the fact that people do not make sense of their activities according to generalized scripts. Rather, they make use of an unconstrained set of resources that they make relevant in their environment. Perhaps most importantly, they engage in subtle processes of problem solving to overcome breakdowns in the kinds of anticipated normal patterns of events that might be captured in scripts and plans. Such problem solving is critical to success because breakdowns are ubiquitous. Analysis of the discourse of dyads or small groups engaging in situated problem solving can reveal how people actually make use of available resources and where they get stuck trying to follow computer scripts. The detailed collaborative procedures captured on video and comprehended through intensive and repeated study are rarely what designers of computer-based scripts might have planned for.

The copier help system is a script that provides computer support for small groups to collaboratively learn how to use the copier. It is an instance of scripting for CSCL. It mediates the users' collaborative actions and their meaning making. It poses the central practical tension that gnaws at the enterprise of CSCL:

a) Collaborative learning is achieved under unique circumstances whose significance is interactively constructed by the learners and cannot be predicted.
b) Computer support attempts to define a specific context and to direct the meaning-making process in order to (i) guide the learning toward pedagogical goals and (ii) provide a real-time model of the learners' state that can steer the delivery of computational resources.

Based on her theoretical, methodological and empirical study, Suchman recommends (p. 181) that computer support compensate for its limitations by: (1) extending its access to the actions and circumstances of the user; (2) clarifying for the user the limits of the computer's access to the users' rich interactional resources; and (3) providing a wider array of alternative resources, particularly to help the users respond to unforeseen breakdowns. These recommendations should be implemented based on careful empirical study of a given application, along the lines of Suchman's video analysis of copier usage. Only this way will designers discover: (1) the relevant factors of the use situation; (2) the way that the user treats the computer as an interaction partner; and (3) the kinds of breakdowns that can occur and the resources that users take advantage of to make sense of and overcome the breakdowns.

5. SCRIPTING GROUP COGNITION

It is not easy to study the details of how people use situational resources to construct shared meaning in computer-mediated learning tasks. In particular, it is hard to delineate what is accomplished by individuals and what is best analyzed at the small-group unit of analysis. Hardest of all, perhaps, is to describe how individual and group cognition – once distinguished – work symbiotically. Schwartz' *The Emergence of Abstract Representations in Dyad Problem Solving* (1995) takes some steps in this direction.

Schwartz scripts three controlled experiments – one in a lab with video camera and two in classrooms – that compare individuals and dyads working on the same science problems. In order to get at the problem-solving process, Schwartz looks at the intermediate problem representations that the students construct, rather than at their final solutions. He finds that although there is little significant difference between individuals and dyads in their final solutions, the groups construct more abstract representations. Schwartz concludes from this that the group-level cognitive processes are qualitatively different from the cognitive processes of the isolated individuals: "Group cognitions sometimes yield a product that is not easily ascribed to the cognitions that similar individuals have working alone. In particular, groups have a tendency to construct representations that are more abstract than individuals' representations" (p. 322).

In the first experiment, where the activities were captured on video, Schwartz was able to see how the dyads were forced to construct collaborative representations, to negotiate their meaning and to overcome breakdowns in shared understanding. These unique, situated, unpredictable interactions and verbalizations produced and made visible joint articulations of the structures of the objects in the scientific problem, leading to insights into the final solution. Because of their interactive work in overcoming the additional hardships introduced by having to negotiate and maintain shared understandings between two people who started with independent ideas, the dyads performed significantly better than would be predicted based on combining the best individual performances of the dyad members.

Unfortunately, the other two experiments were not videotaped and therefore the interactions of the dyad members could not be analyzed. Consequently, Schwartz was largely reduced to speculation that if the interactions could be studied they would show that the processes of overcoming breakdowns in maintaining mutual knowledge fostered the joint construction of abstract graphical and verbal representations that were useful for problem solving: "I suspect that interactional studies would find numerous forms of negotiation depending on the individuals' knowledge and the affordances of the task at hand... Although the process and products of representational negotiation may take numerous forms, I believe that careful attention to the conditions preceding a period of representational negotiation will reveal strong evidence for the important role of mutual-knowledge problems in the co-construction of representations" (p. 348).

6. SCRIPTS FOR FRAMING COLLABORATIVE INTERACTIONS

The preceding quick review of Schank and Abelson (1977), Vygotsky (1930/1978), Suchman (1987) and Schwartz (1995) has attempted to reconceptualize the concept of scripts as situated resources rather than implementable plans for action so as to align the concept with current socio-cultural thought. It has recommended the micro-analysis of how such resources are made sense of in small group interactions in order to guide the design of scripts based on actual examples of the kinds of situated action for which the scripts are intended.

Dillenbourg and Jermann (this volume) display a healthy recognition of the nature of scripts as flexible resources. They take the concept of script not as a cognitive model of how people actually decide what to do, but rather as a design metaphor for finding the delicate balance between too little computer control to be helpful and too much control to allow for flexible group interactions.

Interestingly, they finesse the problem of constraining group interaction by confining scripting to the individual or whole-class activities that precede and that follow the core small-group collaborative activities. They define CSCL scripts to be instructional sequences that prepare for and then reflect upon, but do not interfere with peer interactions. Adopting Schwartz' conclusion that the power of collaborative learning comes from the effort necessary for the group to build a shared understanding, Dillenbourg and Jermann use scripts to set up situations in which groups will be forced to construct group meanings – their SWISH model. The meaning-making phase itself is then left unconstrained, for it is too fragile, complex and unpredictable to be supported by a script that is written in advance.

Chapter 16 is clearly a synthetic presentation, based on extensive experience using scripts in real learning contexts. It would be nice to see some of the detailed interactions that were observed during the experimentation as examples that motivate the principles enumerated in the chapter. Presumably, page limitations for the chapter prohibited that, and one must go back to the earlier individual studies for such examples.

7. SCRIPTS FOR LEARNING AND FOR LIFE

Carmien, et al. (this volume) call for a distributed cognition perspective to account for the interplay of mental and environmental phenomena. While this is an important move, the details of the particular theory developed are also decisive. The preceding discussion has argued for building more on Vygotsky and Suchman than on Schank and Abelson in defining an approach to distributed cognition or group cognition (for a fuller account, see Stahl, 2006). Rather than starting from a theory of individual cognition and then supplementing it to build a "person-plus" theory, it has invoked Vygotsky's theory in which individual cognition is a social-cognition-minus product of internalization processes. In place of adopting a view of scripts as controlling data structures, it has recommended Suchman's conception of situated resources.

Vygotsky's and Suchman's alternative approaches could be used to account for the design, study and analysis of tools for living and tools for learning. Computational tools mediate between people, for instance between a cognitively disabled person and their caregiver or a group of students and their teacher. The tool can be viewed as an externalization of the caregiver's or the teacher's guidance. The users must learn how to use the tool, and they may or may not be able to internalize its guidance to varying degrees.

Carmien, et al. cite Suchman and recognize the dangers of technology-driven design. Careful study – such as that done by Suchman – at a detailed level of interactional granularity would be needed to analyze the specific processes of internalization and externalization and to design the tools for a successful fit to the situated meaning-making interactions through which the tool is put into service. This would also ensure that the users' situated needs drive design.

Together, Chapters 16 and 17 pose central issues for theory building, assessment methodology and design practices in scripting CSCL. They present contrasting approaches themselves and stimulate the consideration of yet other alternatives.

REFERENCES

Garfinkel, H. (1967). *Studies in ethnomethodology*. Englewood Cliffs, NJ: Prentice-Hall.

Garfinkel, H., & Sacks, H. (1970). On formal structures of practical actions. In J. Mckinney & E. Tiryakian (Eds.), *Theoretical sociology: Perspectives and developments* (pp. 337-366). New York, NY: Appleton-Century-Crofts.

Livingston, E. (1995). *An anthropology of reading*. Bloomington, IN: Indiana University Press.

Marx, K. (1867/1976). *Capital* (B. Fowkes, Trans. Vol. I). New York, NY: Vintage.

Sacks, H., Schegloff, E. A., & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language, 50*(4), 696-735.

Sacks, H. (1992). *Lectures on conversation*. Oxford, UK: Blackwell.

Schank, R., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Stahl, G. (2006). *Group cognition: Computer support for building collaborative knowledge*. Cambridge, MA: MIT Press.

Suchman, L. (1987). *Plans and situated actions: The problem of human-machine communication*. Cambridge, UK: Cambridge University Press.

Vygotsky, L. (1930/1978). *Mind in society*. Cambridge, MA: Harvard University Press.

Index

COMPUTER-SUPPORTED COLLABORATIVE LEARNING

1.	*Arguing to Learn*	
	J. Andriessen, M. Baker, D. Suthers (eds.)	ISBN HB 1-4020-1382-5
2.	*Designing for Change in Networked Learning Environment*	
	B. Wasson, S. Ludvigsen, U. Hoppe (eds.)	ISBN HB 1-4020-1383-3
3.	*What We Know About CSCL*	
	J.-W. Strijbos, P.A. Kirschner, R.L. Martens (eds.)	ISBN HB 1-4020-7779-3
4.	*Advances in Research on Networked Learning*	
	P. Goodyear et al. (eds.)	ISBN HB 1-4020-7841-2
5.	*Barriers and Biases in Computer-Mediated Knowledge Communication: And How They May Be Overcome*	
6.	R. Bromme, F.W. Hesse, H. Spada (eds.)	ISBN HB 0-387-24317-8
7.	*Scripting Computer-Supported Collaborative Learning*	
	F. Fischer, I. Kollar, H. Mandl, J.M. Haake	ISBN HB 978-0-387-36947-1

Printed in the United States of America.