

Instr Sci (2012) 40:623–643
DOI 10.1007/s11251-011-9185-2

Successfully carrying out complex learning-tasks through guiding teams' qualitative and quantitative reasoning

B. Slof · G. Erkens · P. A. Kirschner · J. Janssen ·
J. G. M. Jaspers

Received: 16 January 2011 / Accepted: 17 August 2011 / Published online: 4 September 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract This study investigated whether and how scripting learners' use of representational tools in a computer supported collaborative learning (CSCL)-environment fostered their collaborative performance on a complex business-economics task. Scripting the problem-solving process sequenced and made its phase-related part-task demands explicit, namely defining the problem and proposing multiple solutions, followed by determining suitability of the solutions and coming to a definitive problem solution. Two tools facilitated construction of causal or mathematical domain representations. Each was suited for carrying out the part-task demands of one specific problem-solving phase; the causal was matched to problem–solution phase and the mathematical (in the form of a simulation) to the solution–evaluation phase. Teams of learners ($N = 34$, Mean age = 15.7) in four experimental conditions carried out the part-tasks in a predefined order, but differed in the representational tool/tools they received during the collaborative problem-solving process. The tools were matched, partly matched or mismatched to the part-task demands. Teams in the causal-only ($n = 9$) and simulation-only ($n = 9$) conditions received either a causal or a simulation tool and were, thus, supported in only one of the two part-tasks. Teams in the simulation-causal condition ($n = 9$) received both tools, but in an order that was mismatched to the part-task demands. Teams in the causal-simulation condition ($n = 7$) received both tools in an order that matched the part-task demands of the problem phases. Results revealed that teams receiving part-task congruent tools constructed more

B. Slof
University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands

B. Slof (✉)
University Centre for Learning & Teaching, University of Groningen,
Landleven 1, 9747 AD Groningen, The Netherlands
e-mail: B.Slof@rug.nl

G. Erkens · J. Janssen · J. G. M. Jaspers
Utrecht University, P.O. Box 80.140, 3508 TC Utrecht, The Netherlands

P. A. Kirschner
Open University of the Netherlands, P.O. Box 2960, 6401 DL Heerlen, The Netherlands

task-appropriate representations and had more elaborated discussions about the domain. As a consequence, those teams performed better on the complex learning-task.

Keywords Complex learning-tasks · Computer-supported collaborative learning · Qualitative and quantitative representations · Representational scripting · Learner interaction

Introduction

The current interest in complex learning is often regarded as education's response to the rapidly changing demands of society and work complex learning is necessary to carry out the activities endemic to modern real-life tasks which are complex because they (1) cannot be described in full detail, (2) give no certainty about what the best solution is, and (3) require different perspectives on the problem and the problem-solving strategy for their solution (Jonassen 2003; Van Merriënboer and Kirschner 2007). To this end, educational approaches such as collaborative problem-solving are increasingly incorporated into training programs and curricula. The premise underlying this approach is that externalizing one's knowledge, discussing it with peers, and establishing and refining (e.g., specifying and correcting) a team's shared understanding of the problem and problem-domain beneficially affects learning (Hmelo-Silver et al. 2007). That is, teams and individuals may acquire knowledge and skills which can be effectively transferred to and applied in different situations. Educators and instructional designers, however, must realize that these elements of collaborative problem-solving are those which are carried out by experts and that learners (e.g., novices) need ample instructional support and guidance to approximate such a problem-solving approach (Kirschner et al. 2006; Mayer 2004). Without guidance, learners focus on superficial details of problems instead of on underlying domain principles (Corbalan et al. 2009), and employ weak problem-solving strategies such as working via a means-ends strategy towards a solution (Simon et al. 1981).

To address this, the support provided should gradually increase the learners' level of expertise, for example by mimicking the processes of experts in a way that learners are supported in acquiring and applying a well-developed understanding of the domain in question (Reiser 2004). In most domains, this understanding consists of the availability of both qualitative and quantitative representations of the domain which enable constructing meaningful problem representations and flexibly coordinating them (Jonassen 2003; Löhner et al. 2003). Combining representations is beneficial because different representations initiate different kinds of operators which act to produce new information supporting problem solvers in coming to suitable solutions to problems (Frederiksen and White 2002; Scaife and Rogers 1996). *Qualitative representations* represent the concepts underlying a particular domain and the inference rules which interrelate them and, thus, give them meaning. These representations stimulate reasoning about the concepts, their underlying causal principles, and the circumstances under which those principles can legitimately be applied, enabling problem solvers to effectively define the problem and propose multiple solutions for solving it. *Quantitative representations* represent the formalism(s) underlying a particular domain to describe the definitions of concepts and their functional relationships, for example via algebraic equations in the domain of business-economics. Such representations stimulate reasoning about the concepts and their mathematical relationships, enabling evaluation of the effects of proposed solutions and, thus, reaching a solution (Jonassen 2003; Ploetzner et al. 1999). Working with multiple

representations, thus, might be a good way to guide and support complex learning. Although it is acknowledged that this can foster understanding and problem-solving, not all studies confirm this. Common here is that learners experience considerable difficulties translating information from different kinds of representations and coordinating between them (Ainsworth 2006; Vekiri 2002). Learners, for example, might not understand/know:

- which parts of the domain are represented,
- the relationship between the representations and the task/problem at hand,
- how to select, use or construct appropriate representations,
- whether and how they should interrelate the different kinds of representations.

This raises the question whether and how educators and instructional designers can effectively guide learners' problem-solving process and, thus, their complex learning-task performance. The research reported on in this article introduces an instructional approach—*representational scripting*—as a possible solution and examines how and why this affects complex learning-task performance.

Representational scripting

Design principles

Integrating scripting with representational tools (i.e., *representational scripting*) is intended to guide learners in their acquisition of a well developed understanding of a domain and to apply this understanding while solving a problem. Using such tools can facilitate constructing domain-specific representations and, thereby, guide reasoning about the domain. A tool's ontology (i.e., its objects, relations, and rules for combining objects and relations) provides specific representational guidance which makes certain concepts and/or interrelationships (e.g., causal, mathematical) salient above others. In this way, a tool's representational guidance supports externalization of knowledge and ideas about specific aspects of the domain (Fischer et al. 2002; Slobin et al. 2010a). This fosters understanding since it stimulates cognitive and meta-cognitive activities such as (1) selecting relevant information, (2) organizing information into coherent structures, (3) relating information to prior understanding, and (4) determining knowledge and comprehension gaps (Hilbert and Renkl 2008; Stull and Mayer 2007). Embedding representational tools in collaborative settings, such as computer supported collaborative learning (CSCL)-environments, may even further stimulate the elaboration of these representations, due to the environment's emphasis on dialogue and discussion, so that multiple perspectives on the domain arise (De Simone et al. 2001; Janssen et al. 2010a).

The mere availability of a representational tool, however, will not automatically support solving complex problems since such problems are composed of different *part-tasks*, namely (1) determining what the problem to be solved is, (2) proposing possible multiple solutions to the determined problem, (3) judging the suitability of the different solutions and (4) reaching the solution. To do all of this, multiple perspectives of the problem domain (i.e., problem representations) are required (Van Bruggen et al. 2003). Problematic here is that specific representational tools, each with its specific ontology, guide learners in constructing and discussing specific representations of the domain and are, thus, not appropriate for carrying out all aspects of the task (Ainsworth 2006; Schnotz and Kürschner 2008). In other words, a tool's ontology provides a specific kind of guidance, which is specified through its expressiveness and processability (see Table 1).

Table 1 Specification of a representational tools' ontology and representational guidance

Ontology				Representational guidance
Expressiveness		Processability		
Specificity	Precision	Elaboration	Order	
Low–medium	Causal directed relations	Quasi-structured	First	Qualitative inference-making
High	Model directed relations	Structured	Second	Quantitative inference-making

Expressiveness refers to which concepts and interrelationships can be represented (i.e., a tool's specificity) and how accurately this is done (i.e., a tool's precision). *Processability* refers to the differences in processing the information from the representation caused by the differences in expressiveness, and which determines the number and quality of inferences that can be made. Less expressive (i.e., less specific and less precise) ontologies have the advantage of being highly processable (Larkin and Simon 1987) making it easy to make many inferences from them (i.e., elaboration). Such ontologies guide learners in elaborating on the concepts of the domain and in relating them to the problem (e.g., Jonassen 2003). These ontologies, however, do not have much expressive power (Cox 1999); the inferences made from them are neither specific nor precise. The *order* of an ontology (Frederiksen and White 2002) determines the quality of the inferences that can be made (i.e., kind of reasoning used). A *first order* representational tool supports reasoning about causal relationships and guides discussion and/or thought about the problem and possible solutions. A *second order* representational tool is more expressive—and thus more specific and precise—and supports quantitative inference-making enabling negotiation and/or determination of suitability of the proposed solutions.

When the tools' ontology is incongruent with the demands of a specific part-task this will lead to communication problems and decreased performance (Slof et al. 2010b; Van Bruggen et al. 2003). A reason for this might be that the tool used is not expressive enough for all part-tasks. To this end, it might be beneficial if learners are provided with different representational tools for which the representational guidance of each tool is congruent (i.e., ontologically matched) with the demands of each part-task. To ensure alignment of the tool, its use, and the part-task demands *scripting* can be employed (Dillenbourg 2002; Kollar et al. 2007). According to Dillenbourg, a script is “a set of instructions regarding to how the group members should interact, how they should collaborate and how they should solve the problem” (p. 64). Integrating scripting with representational tools sequences the part-tasks, makes the different part-task demands explicit, and tailors the congruence of the representational guidance to the part-task demands. This should actively engage learners in a process of making sense of the domain in question by articulating and discussing multiple perspectives on the problem and of the problem-solving strategy (Hmelo-Silver et al. 2007; Ploetzner et al. 1999). Representational scripting, thus, is intended to stimulate learners to carry out *cognitive activities* such as (1) discussing the goal of the problem-solving task/part-tasks, (2) discussing and selecting concepts, principles, and procedures in the domain, and (3) formulating and revising their decisions (Slof et al. 2010b). Learners may also be induced to employ a proper problem-solving strategy and reflect on its suitability through carrying out *meta-cognitive activities* (Moos and Azevedo 2008). This requires that learners discuss (1) how they should approach the problem, (2) whether they have finished the part-tasks on time, and (3) how suitable their approach was.

Fostering complex learning-task performance in business-economics

In the research reported on here, learners collaborated on solving a case-based business-economics problem in which they had to advise an entrepreneur about changing the business strategy to increase profits. To gain insight into the part-tasks and their required domain-specific representations, a learning-task analysis (Anderson and Krathwohl 2001) was conducted. Based on these insights, the sequence and demands of the part-tasks were specified and part-task congruent representational tools were developed (see Table 2).

In the *problem–solution phase* learners, first, have to determine what the problem is and what the most important factors are for its solution. Then they have to formulate possible business-strategy changes (i.e., interventions) and elucidate how the changes might solve the problem (i.e., problem–solution) by describing how the changes affect outcomes (i.e., company result). The representational tool should, thus, facilitate construction and discussion of a causal problem-representation by causally relating the concepts to each other and to possible interventions. Figure 1 shows an expert’s qualitative representation of the domain. The *causal representational tool* facilitates representing the concepts, the interventions and their causal interrelationships. Selecting relevant concepts and interventions and causally relating them supports the effective exploration of the solution space and, thus, of finding multiple solutions to the problem. Learners receiving such a tool could, for example, make explicit that an intervention such as “receiving a rebate from a supplier” affects the “total variable costs” which in turn affects the “total costs”. Through gradually increasing learners’ understanding of the underlying qualitative principles governing the domain, it should be easier for them to come up with an intervention that will solve the problem.

In the *solution–evaluation phase* learners have to determine the financial consequences of their proposed interventions and formulate a definitive advice by discussing the suitability of the different interventions with each other. The representational tool must, therefore, facilitate construction and discussion of a quantitative representation by specifying the relationships as algebraic equations. Figure 2 shows a quantitative presentation of the domain as seen by an expert. The *simulation representational tool* facilitates representing the concepts and their mathematical interrelationships. Selecting relevant concepts and specifying the interrelationships as algebraic equations supports evaluating the effects of the proposed interventions and, thus, in choosing a suitable advice. Learners receiving such a tool could, for example, simulate how an intervention such as “receiving a rebate from a supplier” affects the “total variable costs” and whether this affects the “total costs”. By manipulating the input values, the values of all other related concepts are automatically computed. Since such quantitative representations can only be properly understood and applied when learners have a well-developed qualitative understanding of the domain, this kind of support is only appropriate for carrying out this type of part-task.

Table 2 Matching the representational tools’ guidance to the task demands of each problem phase

Problem phase	Task demands	Representational tool	Representational guidance
Problem–solution	Defining the problem and proposing multiple solutions to the problem	Causal	Representing and discussing causal relationships between the concepts and the possible solutions
Solution–evaluation	Determining suitability of the solutions and coming to a definitive solution to the problem	Simulation	Representing and discussing mathematical relationships between the concepts and enabling manipulation of their values

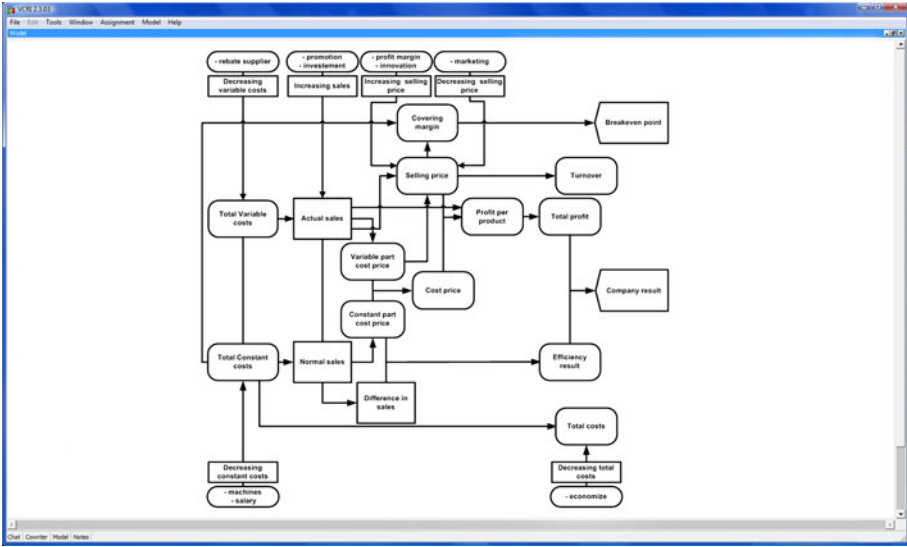


Fig. 1 Experts' qualitative representation of the domain

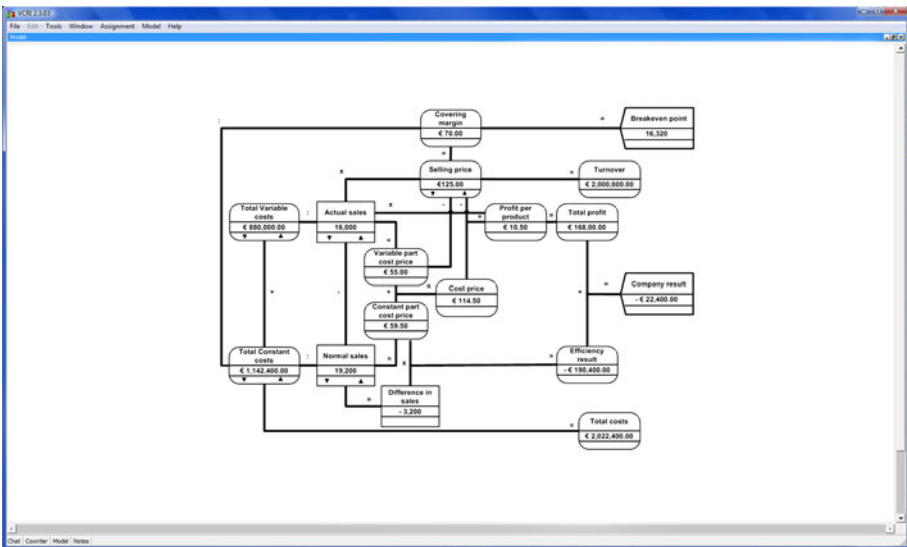


Fig. 2 Experts' quantitative representation of the domain

Research questions and hypotheses

The present study is aimed at answering the following research question: “How and why does constructing part-task congruent representations affect the collaboration process and complex learning-task in teams?” Due to the presumed match between tools' representational guidance and all part-tasks demands (i.e., representational scripting), it was

hypothesized that teams receiving and using such tools, in comparison to teams that did not, would:

H1 Achieve a better problem solving performance, evidenced by proposing better solutions and a better definitive advice to the problem.

H2 Experience a qualitatively better problem solving process, evidenced by

- (a) constructing representations that are more suited for carrying out the part-tasks, and
- (b) having more fruitful discussions about the problem, their problem-solving strategy and the problem domain.

Method

Participants

Participants were students from six business-economics classes in three secondary schools in the Netherlands. The total sample consisted of 102 students (61 male, 41 female; mean age = 15.7 years; SD = 0.56, Min = 14, Max = 17). The students were, within classes, randomly assigned to 34 teams; nine triads in the causal-only, simulation-only and simulation-causal conditions and seven triads in the causal-simulation condition. Since the collaborative problem solving task was developed in cooperation with their teachers it is regarded as a suited pedagogical activity for the students at that point in the curriculum. A pre-test (20 multiple-choice items, measuring factual, conceptual and procedural knowledge, $\alpha = 0.60$) was administered to determine students' prior understanding of the domain. On average, students scored 10.9 out of the maximum of 20 points, and there were no significant differences between conditions and classes.

Design

To study the effects of representational scripting, four experimental conditions were defined by matching, partly matching or mismatching the tool's representational guidance to the demands of each problem phase (see Table 3). The rationale behind this design is twofold, namely it may provide insight into the effects of (1) a specific representational tool and (2) the sequence in which the tools are provided. By doing so not only the value of qualitative and quantitative representations but also their interrelationship can be examined.

Table 3 Overview of the experimental conditions

Conditions	Problem phases and provided representational tools		Match/mismatch
	Problem–solution	Solution–evaluation	
Causal-only	<i>Causal tool</i>	Causal tool	Match for the solution phase only
Simulation-only	Simulation tool	<i>Simulation tool</i>	Match for the evaluation phase only
Simulation-causal	Simulation tool	Causal tool	Mismatch for all problem phases
Causal-simulation	<i>Causal tool</i>	<i>Simulation tool</i>	Match for all problem phases

Scripting the problem-solving process sequenced and made the part-task demands explicit. These demands are (1) defining the problem and proposing multiple solutions, and (2) determining the suitability of the solutions and coming to a definitive solution. Teams in all conditions had to carry out the part-tasks in a predefined order, but differed in the representational tool they received. Teams in the *matched* (i.e., causal-simulation) and the *mismatched* (i.e., simulation-causal) conditions received both representational tools in a phased order. The difference between these conditions was that the tools were part-task congruent or not. In the simulation-causal condition the teams received both tools, but in an order that was mismatched to the part-task demands (i.e., simulation tool for the definition phase and causal tool for the evaluation phase). In contrast, teams in the causal-simulation condition received representational tools considered to be well-suited to the part-task demands of each problem phase. In the *partly matched* conditions (i.e., causal-only, simulation-only), teams received either a causal or a simulation tool for carrying out both part-tasks and for constructing the part-task related representations. The tool's representational guidance matched only one of the part-task demands.

CSCL-environment

The teams worked in a CSCL-environment called Virtual Collaborative Research Institute (VCRI; Jaspers et al. 2005; see Fig. 3), a groupware application for supporting the collaboratively carrying out problem-solving tasks and research projects. For this study, the tools in VCRI were augmented with representational scripting. In the *Assignment menu*, team members can find the description of the task/part-tasks. Furthermore, additional information sources such as a definition list, formula list, and problem-solving clues were

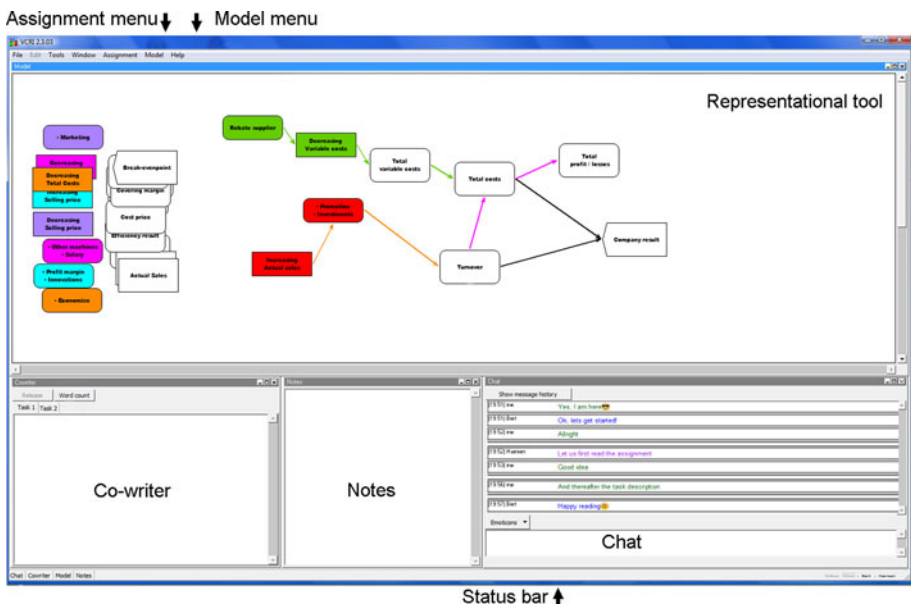


Fig. 3 Screenshot of the VCRI-environment (causal representational tool)

also available here. The *Model menu* enabled team members to construct and adjust their representations by adding or deleting relationships. At the start of the first lesson, all diagram boxes—representing the different concepts/solutions—were placed on the left side of the *Representational tool* so team members could select them when they wanted to add a new causal or mathematical relationship. The *Chat tool* enabled synchronous communication and supported team members in externalizing and discussing their knowledge and ideas about the content of the domain and their problem-solving strategy. The chat history is automatically stored and can be re-read by the team members. The *Co-writer* is a shared text-processor where team members could collaboratively formulate and revise their decisions concerning the part-tasks. The *Notes tool* is an individual notepad that allowed team members to store information and structure their own knowledge and ideas before making them explicit to the other members. The *Status bar* is an awareness tool that displayed which team members were logged into the system and which tool a member was using at any specific moment.

The different conditions were information equivalent and, thus, only differed in the way the representational tools were intended to guide performance. All teams had to carry out the part-tasks in a predefined order namely starting with the problem–solution phase and ending with the solution–evaluation phase. When the team members agreed that the part-task demands of the first phase were completed, they had to ‘close’ that phase in the assignment menu. This ‘opened’ the second phase, which had two consequences for all team members, namely they were instructed to carry out the part-task demands of this phase and then revise their representation of the domain so it concurred with the decisions they made when carrying out this part-task. Teams in the causal-only and simulation-only conditions were facilitated in elaborating on their previously constructed representation. Since those teams kept the same representational tool, all concepts and their relationships remained visible and could be revised as the team members deemed appropriate for carrying out the task demands of the following phase. Teams in the simulation-causal and causal-simulation conditions were facilitated in acquiring and applying a different qualitative or quantitative perspective of the domain. Their previously selected concepts remained visible and they were instructed to replace the relationships by specifying them in either a causal manner (i.e., simulation-causal) or as algebraic equations (i.e., causal-simulation) with the aid of their new tool.

Procedure

All 34 teams spent four, 45-min lessons solving the problem during which learners worked on separate computers. Before the first lesson, learners received an instruction about the team composition, the complex learning-task and the CSCL-environment. The instruction made clear that their score on the complex learning-task would serve as a grade affecting their GPA. Learners worked on the problem in the computer classroom and all actions (e.g., constructed representations, contributions to the chat-discussion, and decisions concerning the part-tasks) were logged. During the lessons, the teacher was on stand-by for task-related questions and a researcher was present for technical support.

Variables and analyses

To gain insight in how and why the representational scripting affects learning-task performance in CSCL an effect oriented and a process oriented research approach were combined (e.g., Janssen et al. 2010b). Data on both learning results (i.e., complex

learning-task performance) and learning process (i.e., constructed representations and learner interaction) were collected.

Complex learning-task performance

To examine performance quality, an assessment form for both part-tasks and for the quality of the definitive advice was developed. Table 4 provides a description of the aspects on which the decisions were evaluated, the number of items, and their internal consistency scores (i.e., Cronbach's alpha). All 28 items could be coded as '0' (wrong), '1' (adequate) or '2' (good); the higher the code, the higher the quality of the decision. Teams could, thus, achieve a maximum score of 56 points for their complex learning-task performance (28 items \times 2 points) and a minimum of 0 points. The internal consistency score for the whole complex learning-task performance was 0.84.

The effect of condition was examined through conducting a one-way ANOVA on the total performance score that the teams received. Planned orthogonal contrasts were constructed to examine whether a significant difference could be found between the (1) partly matched conditions and the matched/mismatched conditions), (2) matched condition (i.e., causal-simulation) and the mismatched condition (i.e., simulation-causal), and (3) two partly matched conditions (i.e., causal-only versus simulation-only).

Constructed representations

A content analysis was conducted on the phase-related representations to examine the quality of the constructed representations. To this end, the representations were selected at the end of each problem phase just before a phase was 'closed', and transferred from the log-files using the Multiple Episode Protocol Analysis program (MEPA; Erkens 2005). The

Table 4 Items and reliability of complex learning-task performance

Criteria	Description	Items	α
Suitability	Whether the teams' decisions were suited to the different part-tasks	6	0.65
Elaboration	Number of different business-economics concepts or financial consequences incorporated in the decisions to the different part-tasks	6	0.47
Justification	Whether the teams justified their decisions to the different part-tasks	6	0.51
Correctness	Whether the teams used the business-economics concepts and their interrelationships correctly in their decisions to the different part-tasks	6	0.55
Continuity	Whether the teams made proper use of the decisions from the prior problem phase	1	–
Quality advice	Whether the teams gave a proper definitive advice Number of business-economics concepts incorporated in the advice Number of financial consequences incorporated in the advice Whether the definitive advice conformed to the guidelines provided	3	0.71
Total score	Overall score on the complex learning-task performance	28	0.84

representations were automatically coded by comparing them with the expert’s representations (see Figs. 1, 2).

The effect of condition was examined by analyzing the part-task related representations of the concepts, their relationships and the correctness of those relationships.

Learner interaction

MEPA was also used to examine the quality of learner interaction. The content of the chat-protocols was assumed to represent what learners know and consider important for carrying out the problem-solving task (Chi 1997; Moos and Azevedo 2008). MEPA uses a multidimensional data structure, allowing chat-protocols to be segmented into multiple levels for analysis, here the episodic level and the event level. Measurement at the *episodic level* was aimed at gaining insight into the learners’ *meta-cognitive, cognitive and off-task activities* (see Table 5). An episode is regarded as a dialogue between minimally two learners in which a distinct discourse topic is discussed and which ends with a confirmation by at least two learners that they understood each other. For example, discussing the suitability of a problem solving strategy requires the involvement of multiple learners who each use more than one utterance to make their point (Mercer et al. 2004). The topics were hand-coded and Cohen’s kappa was computed for three independently coded chat-protocols (2,457 lines) by two coders. An overall Cohen’s Kappa of 0.74 was found, an intermediate to good result (Cicchetti et al. 1978).

Measurement at the *event level* was aimed at gaining insight into the discussion of concepts, interventions and the ways of interrelating them (see Table 6). A problem here is that even within in a single sentence, multiple concepts or statements may be expressed and, thus, would require multiple codes (Strijbos et al. 2006). Utterances were automatically segmented into smaller, still meaningful, subunits with a MEPA-filter using 300 ‘if-then’ decision rules. Punctuation marks (e.g., period, exclamation point, question mark) and connecting phrases (e.g., ‘and if’, or ‘but if’) were used to segment the utterances. After segmentation, coding was done automatically with a MEPA-filter which makes use

Table 5 Coding and category kappa’s (K_c) of the meta-cognitive, cognitive and off-task activities

Activities	Discourse topic	Discussion of	K_c
<i>Meta-cognitive</i>			0.75
	Planning	The problem-solving strategy; how and when the group has to carry out a specific activity	0.65
	Monitoring	Whether they have finished the part-tasks on time	0.71
	Evaluating	The suitability of their problem-solving strategy	0.78
<i>Cognitive</i>			0.72
	Preparation	The goal of the problem-solving task and the different part-tasks	0.55
	Executing	Content-related topics and formulating/revising their decisions to the part-tasks	0.85
	Ending	How, where, and when their decisions need to be registered	0.75
<i>Off-task</i>			0.79
	Social	Non-task related topics	0.83
	Technical	Problems with the CSCL-environment	0.76

Table 6 Coding and category kappa's (K_c) MEPA-filter of the discussion of the domain

Categories	Discussion of the	K_c
<i>Concepts</i>	Business-economics concepts	0.70
<i>Solutions</i>	Possible interventions	0.73
<i>Relations</i>	Different kinds of interrelationships	0.65
Causal	Causal relationship within/between concepts/solutions	0.69
Mathematical	Quantitative relationships within/between concepts	0.57

of 814 'if-then' decision rules containing explicit references to a concept, solution or relationship (e.g., name, synonyms, etc.) which were coded as representing that concept, solution or relationship. Comparison of the three hand-coded protocols (2,457 lines) to the automatically coded protocols yielded an overall Cohen's Kappa's ranging from 0.65 to 0.73.

The effect of condition on the quality of the learner interaction was determined using multilevel analysis (MLA) which addresses the statistical problem of non-independence often associated with CSCL research (Cress 2008; Janssen et al. 2010a). Many statistical techniques (e.g., *t*-test, ANOVA) assume score-independence and violating this assumption compromises interpretation of the output of the analyses (e.g., *t*-value, standard error, *P*-value). Non-independence was determined by computing the intraclass correlation coefficient and its significance (Kenny et al. 2006) for all dependent variables relating to learner interaction. Its value demonstrated non-independence ($\alpha < 0.05$) for all tests, justifying MLA use. MLA entails comparing the deviance of an empty model and a model with one or more predictor variables to compute a possible decrease in deviance. The latter model is considered better when there is a significant decrease in deviance from the empty model (tested with a χ^2 -test). Almost all reported χ^2 -values were significant ($\alpha < 0.05$) and, therefore, the estimated parameters of these predictor variables (i.e., effects of condition) were tested for significance. Since there were specific directions of the results expected all analyses are one-tailed.

Results

Complex learning-task performance

First the effects of the four different conditions on the total score for team learning-task performance were examined. Inspection of the means and standard deviations (see Table 7) revealed differences between teams in the causal-only ($M = 28.22$, $SD = 7.50$), simulation-only ($M = 28.00$, $SD = 4.44$), simulation-causal ($M = 31.56$, $SD = 6.46$), causal-simulation ($M = 39.14$, $SD = 1.22$) conditions. One-way ANOVA revealed a significant effect of condition on learning-task performance, $F(3, 21.50) = 7.00$, $P < 0.01$, $\omega^2 = 0.33$ (Brown-Forsythe because homogeneity of variance assumption was violated). Next, the constructed planned orthogonal contrasts were carried out to compare the (1) single tool partly matched conditions to the multi tool matched and mismatched conditions, (2) matched condition (i.e., causal-simulation) to the mismatched condition (i.e., simulation-causal), and (3) two partly-matched conditions (i.e., causal-only versus simulation-only). Analysis revealed that teams in the multiple tool conditions significantly outperformed the

Table 7 Means and standard deviations for differences between conditions concerning complex learning-task performance

Criteria	Causal-only condition ($n_{\text{team}} = 9$) <i>M</i> (SD)	Simulation-only condition ($n_{\text{team}} = 9$) <i>M</i> (SD)	Simulation-causal condition ($n_{\text{team}} = 9$) <i>M</i> (SD)	Causal-simulation condition ($n_{\text{team}} = 7$) <i>M</i> (SD)
Suitability	9.89 (2.62)	9.89 (1.83)	10.00 (2.24)	12.00 (0.00)
Elaboration	6.22 (2.33)	6.33 (1.87)	7.22 (2.59)	9.00 (0.58)
Justification	3.00 (1.50)	3.11 (1.36)	4.00 (1.73)	5.14 (1.46)
Correctness	4.44 (1.67)	4.22 (1.20)	5.11 (1.54)	6.14 (0.38)
Continuity	1.44 (0.73)	1.56 (0.53)	1.56 (0.73)	2.00 (0.00)
Quality advice	3.22 (1.39)	2.89 (1.27)	3.67 (1.12)	4.86 (0.90)
Total score	28.22 (7.50)	28.00 (4.44)	31.56 (6.46)	39.14 (1.22)

teams in the single tool conditions, $t(21.61) = 3.97, P < 0.01$ (equal variances not assumed), $r = 0.65$ and that teams in the matched condition significantly outperformed teams in mismatched condition, $t(15.40) = 7.24, P < 0.01$ (equal variances not assumed), $r = 0.88$. No significant difference was found between teams in the causal-only and simulation-only conditions, $t(30) = 1.50, P > 0.05, r = 0.26$. To examine the differences between the mismatched condition and the partly matched conditions, post-hoc tests (Games-Howell) were carried out. No significant differences were found ($t(16) = 1.01, P > 0.05, r = 0.24$, and $t(16) = 1.36, P > 0.05, r = 0.32$ respectively), indicating that learning-task performance in the mismatched condition did not differ from performance in both partly matched conditions.

Overall, the results show that constructing different kinds of representations is beneficial to constructing only one kind of representation, but that this advantage is only significant when a tool’s representational guidance is matched to the task demands of each problem phase (i.e., the matched, causal-simulation condition).

Constructed representations

Content analyses of the quality of the constructed representations in relation to the task demands of the problem phases revealed several differences between conditions (see Fig. 4).

Compared to teams in the simulation-only condition, teams in the causal-only condition represented significantly more concepts ($t(16) = 2.56, P = 0.02$) and relationships ($t(16) = 4.24, P = 0.00$). Also, teams in the matched and mismatched conditions had a more diverse pattern in representing domain content. Those teams also adjusted their domain representations more often when carrying out the part-tasks. Compared to teams in the mismatched condition, teams in the matched condition significantly represented (1) more relationships during the problem–solution phase ($t(14) = 2.77, P = 0.03$) but made more errors representing them ($t(14) = 4.18, P = 0.00$), (2) fewer relationships during the solution–evaluation phase ($t(14) = -2.29, P = 0.05$) but made fewer errors representing them ($t(14) = -3.59, P = 0.00$).

Overall, these analyses show that teams using multiple representational tools, in contrast to teams using a single tool, varied more in representing the domain content. This was,

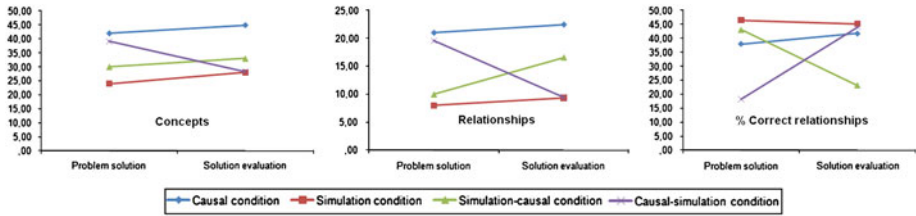


Fig. 4 Content analyses for effects of condition concerning learner tool use

however, only beneficial for teams in the matched condition since they became more selective in representing the concepts and in specifying their relationships as algebraic equations.

Learner interaction

Cognitive, meta-cognitive and off-task activities

Inspection of the means and standard deviations (see Table 8) revealed differences between conditions concerning the meta-cognitive and cognitive activities learners exhibited. MLAs revealed that condition was a significant predictor for these differences (see Tables 9, 10, 11). First, a category effect for *meta-cognitive activities* was found when comparing learners in the matched condition to learners in both the simulation-only ($\beta = 5.37, P = 0.07$) and mismatched conditions ($\beta = 6.17, P < 0.05$). Learners in the matched condition exhibited more meta-cognitive activities than learners in both other conditions. This was mainly due to the fact that learners in that condition more often discussed whether they had finished their part-tasks on time (i.e., monitoring) than learners in the simulation-only ($\beta = 3.96, P < 0.05$) and mismatched conditions ($\beta = 4.17, P < 0.05$). Also, learners in the matched condition more often discussed what the goal of

Table 8 Means and standard deviations for differences between conditions concerning meta-cognitive, cognitive and off-task activities

	Causal-only condition ($n_{\text{learner}} = 27$) M (SD)	Simulation-only condition ($n_{\text{learner}} = 27$) M (SD)	Simulation-causal condition ($n_{\text{learner}} = 27$) M (SD)	Causal-simulation condition ($n_{\text{learner}} = 21$) M (SD)
<i>Meta-cognitive</i>	17.55 (9.11)	14.78 (9.94)	13.85 (8.06)	20.14 (10.71)
Planning	5.14 (3.41)	3.81 (3.31)	3.46 (3.15)	4.62 (3.61)
Monitoring	10.50 (6.08)	9.19 (6.01)	8.92 (5.62)	13.14 (7.78)
Evaluating	1.91 (2.11)	1.78 (2.28)	1.46 (1.30)	2.38 (2.36)
<i>Cognitive</i>	15.36 (11.37)	17.63 (11.38)	14.23 (9.71)	20.52 (8.04)
Preparation	1.86 (1.94)	1.74 (1.70)	2.54 (2.23)	2.90 (1.70)
Executing	12.50 (9.62)	14.37 (9.93)	10.50 (7.69)	15.29 (7.96)
Ending	1.00 (1.16)	1.52 (1.55)	1.19 (1.47)	2.33 (2.20)
<i>Off-task</i>	11.00 (7.78)	11.89 (11.32)	7.77 (5.52)	9.62 (5.91)
Social	9.41 (7.16)	10.30 (10.16)	7.04 (5.75)	8.57 (5.96)
Technical	1.59 (1.84)	1.59 (2.37)	0.73 (1.49)	1.05 (1.50)

Table 9 Estimates for random intercept model for differences between conditions concerning meta-cognitive activities

	<i>Meta-cognitive</i>		Planning		Monitoring		Evaluating	
	β	SE	β	SE	β	SE	β	SE
γ_{00} = Intercept	19.63	2.29	4.59	0.75	12.30	1.40	2.74	0.60
β_1 = causal-simulation vs. causal-only	2.39	3.59	-0.53	1.19	2.57	2.18	0.37	0.94
β_2 = causal-simulation vs. simulation-only	5.37	3.45	0.80	1.14	3.96*	2.09	0.60	0.91
β_3 = causal-simulation vs. simulation-causal	6.17*	3.47	1.15	1.15	4.17*	2.10	0.86	0.91
Variance								
Group level	67.09		9.45		35.51		1.85	
Individual level	24.51		2.00		5.31		2.63	
Deviance	681.67		494.19		612.64		377.51	
Decrease in deviance	16.17**		8.32*		14.02**		5.05	

* $P < 0.05$, ** $P < 0.01$

Table 10 Estimates for random intercept model for differences between conditions concerning cognitive activities

	<i>Cognitive</i>		Preparation		Executing		Ending	
	β	SE	β	SE	β	SE	β	SE
γ_{00} = Intercept	21.11	2.50	2.90	0.43	15.29	2.59	2.33	0.36
β_1 = causal-simulation vs. causal-only	4.70	4.00	1.04*	0.61	2.34	3.57	1.33**	0.51
β_2 = causal-simulation vs. simulation-only	2.89	3.85	1.16*	0.57	0.92	3.45	0.81	0.48
β_3 = causal-simulation vs. simulation-causal	6.29	3.87	0.37	0.58	4.78	3.46	1.14*	0.49
Variance								
Group level	76.53		3.55		52.92		2.56	
Individual level	32.95		0.11		29.14		0.07	
Deviance	696.33		392.88		666.55		362.49	
Decrease in deviance	15.72**		6.82*		14.45**		8.17*	

* $P < 0.05$, ** $P < 0.01$

the problem-solving task and the different part-tasks were (i.e., preparation) than learners in both causal-only ($\beta = 1.04$, $P < 0.05$) and simulation-only conditions ($\beta = 1.16$, $P < 0.05$). Finally, learners in the matched condition more often discussed whether they should end a part-task (i.e., ending) than learners in the causal-only ($\beta = 1.33$, $P < 0.01$), simulation-only ($\beta = 0.81$, $P = 0.05$) and mismatched ($\beta = 1.14$, $P < 0.05$) conditions.

Overall, these analyses show that learners in the matched condition exhibited more meta-cognitive and cognitive activities than learners in the other conditions.

Concepts, solutions and relations

Differences were found for discussions of the domain between conditions (see Table 12). MLAs revealed two category effects when comparing learners in the matched condition to learners in the mismatched condition (see Table 13).

Table 11 Estimates for random intercept model for differences between conditions concerning off-task activities

	<i>Off-task</i>		Social		Technical	
	β	SE	β	SE	β	SE
γ_{00} = Intercept	9.62	2.18	8.57	2.16	1.05	0.45
β_1 = causal-simulation vs. causal-only	-1.47	3.03	-0.85	2.99	-0.58	0.63
β_2 = causal-simulation vs. simulation-only	-2.27	2.91	-1.72	2.88	-0.54	0.60
β_3 = causal-simulation vs. simulation-causal	1.88	2.92	1.58	2.89	0.31	0.61
Variance						
Group level	50.65		39.42		3.10	
Individual level	16.46		19.56		0.39	
Deviance	654.12		637.65		386.91	
Decrease in deviance	17.76**		12.67**		5.04	

* $P < 0.05$, ** $P < 0.01$

Table 12 Means and standard deviations for differences between conditions concerning the discussion of the domain

	Causal-only condition ($n_{\text{learner}} = 27$) M (SD)	Simulation-only condition ($n_{\text{learner}} = 27$) M (SD)	Simulation-causal condition ($n_{\text{learner}} = 27$) M (SD)	Causal-simulation condition ($n_{\text{learner}} = 21$) M (SD)
<i>Concepts</i>	17.05 (18.66)	24.48 (20.29)	15.92 (11.76)	27.95 (20.69)
<i>Solutions</i>	14.55 (15.53)	16.41 (17.58)	12.27 (10.33)	20.43 (15.65)
<i>Relations</i>	22.18 (19.81)	26.00 (18.28)	17.58 (13.25)	34.14 (20.61)
Causal	14.05 (13.20)	15.15 (12.44)	10.58 (9.73)	20.71 (13.35)
Mathematical	8.14 (7.37)	10.85 (7.78)	7.00 (4.75)	13.43 (9.03)

Table 13 Estimates for random intercept model for differences between conditions concerning the discussion of the domain

	<i>Concepts</i>		<i>Solutions</i>		<i>Relations</i>		Causal		Mathematical	
	β	SE	β	SE	β	SE	β	SE	β	SE
γ_{00} = Intercept	27.95	5.91	20.43	4.96	34.14	5.62	20.71	3.88	13.43	2.02
β_1 = causal-simulation vs. causal-only	9.60	8.14	4.97	6.82	10.66	7.75	5.80	5.35	4.96*	2.80
β_2 = causal-simulation vs. simulation-only	3.47	7.88	4.02	6.61	8.14	7.49	5.57	5.17	2.58	2.69
β_3 = causal-simulation vs. simulation-causal	11.91	7.90	8.22	6.62	16.47*	7.51	10.14*	5.19	6.36*	2.71
Variance										
Group level	158.72		104.22		184.54		78.18		39.75	
Individual level	191.87		137.25		159.45		79.35		15.30	
Deviance	783.55		746.81		790.15		714.55		634.31	
Decrease in deviance	20.33**		17.59**		21.68**		18.41**		16.85**	

* $P < 0.05$, ** $P < 0.01$

First, a marginally significant category effect for *concepts* ($\beta = 11.91$, $P = 0.07$) was found; learners in the matched condition discussed more concepts than learners in the mismatched condition.

Second, a significant category effect for *relations* ($\beta = 16.47$, $P < 0.05$) was found; learners in the matched condition discussed more and different kinds of relationships than learners in the mismatched condition. MLAs also revealed that learners in the matched condition discussed more mathematical relationships than learners in both the causal-only ($\beta = 4.96$, $P < 0.05$) and mismatched ($\beta = 6.36$, $P < 0.05$) conditions.

Overall, these analyses show that teams in the matched condition had more elaborate discussions about the domain than teams in the mismatched condition.

Discussion

This study examined how and why scripting the use of representational tools (i.e., representational scripting) in a CSCL-environment affects a team's performance of a complex business-economics task. To examine the effects of this approach, a combined effect and process oriented research approach on collaborative learning was used (Janssen et al. 2010b).

The effect oriented view revealed that teams of learners receiving representational tools that were completely matched to the part-task demands of the problem phases, (i.e., a causal representation followed by a simulation representation) performed better on the complex learning task. That is, those teams formulated better decisions with respect to the part-tasks and came up with better definitive solutions to the problem than teams in the partly matched (i.e., causal-only, simulation-only) and mismatched (i.e., a simulation representation followed by a causal one) conditions. No significant difference between the partly matched and mismatched conditions was found.

To explain how and why representational scripting affected the learning process, a process oriented approach was used. Three differences concerning the quality of the learning process were found.

First, teams in the both the matched and mismatched conditions adjusted their domain representations to the part-task demands of the problem phases. However, this was only beneficial for teams in the matched condition since they started with the construction of a broad representation and gradually became more selective in representing the concepts and specifying their relationships as algebraic equations. This is the way that solving such a problem should theoretically be carried out (Van Merriënboer and Kirschner 2007). In contrast, teams who had access to only one of the representational tools (i.e., the partly matched conditions) showed a stable representation pattern of the domain content. Those teams either represented many concepts and relationships (i.e., causal-only) or did not (i.e., simulation-only) and were, thus, less occupied with fine-tuning their representations to the different part-task demands.

Second, teams in the matched condition carried out more cognitive and meta-cognitive activities than teams in the other conditions. They more often discussed (1) whether they had finished their part-tasks on time (i.e., monitoring), (2) what the goal of the problem-solving task and the different part-tasks were (i.e., preparing), and (3) whether they should end a part-task (i.e., ending). Carrying out those meta-cognitive and cognitive activities is often regarded as beneficial to collaborative problem-solving (Hmelo-Silver et al. 2007).

Third, teams in the matched condition had more elaborate discussions of the domain content than teams in the mismatched condition. The representational scripting shaped the

use of the representational tools and guided learners' content-related interaction towards acquiring and applying suitable qualitative and quantitative problem representations.

Although the results indicate that scripting learners' tool use seems beneficial for solving complex problems, some of the findings require further discussion. Unexpectedly, almost no differences for learners' discussion of the domain content were found in comparisons of teams in the matched condition to those in the partly matched conditions. The role of scripting might account for this. Structuring the problem-solving process into phases, each focusing on one of the part-tasks, could have affected the content related interaction in a phase-equivalent manner (Dillenbourg 2002). That is, all teams were instructed to construct a domain representation for each part-task and were, thereby, stimulated to discuss the domain content. This explanation is consistent other research on CSCL showing that collaborative construction of representations stimulates learners' cognitive activities (De Simone et al. 2001; Janssen et al. 2010a). This line of reasoning might seem to contradict the result that teams in the mismatched condition had fewer discussions about the content of the domain than teams in the matched condition. However, when the instructions for problem-solving are not completely congruent with the representational tools used, the scripting might negatively affects learners' discussions. Another limitation may lie in the measurement of the quality of the learning process. Solely coding and counting the number of concepts and relationships discussed and represented, though useful, might not lead to full understanding of the dynamics of collaborative learning (Hmelo-Silver et al. 2008). It does not, for example, provide insight into (1) the evolution of understanding and the correctness of the content-related interaction and (2) how learners translate information from and coordinate information between their constructed representations. One way to address this, is to determine how many errors learners make when interrelating the concepts per problem phase. Insight into the quality can be gained by comparing the number and kinds of errors made in each phase.

Implications and future research

Representational scripting appears to have positive effects on learning. When properly matched to part-task demands, a representation's specific ontology can evoke elaborate and meaningful discussion of the domain and foster complex learning-task performance (Ainsworth 2006; Slof et al. 2010a). These results are in line with those of others who stress the importance of creating and interrelating qualitative and quantitative representations of the domain for learning (Frederiksen and White 2002; Löhner et al. 2003). Those studies, however, do not provide guidelines for designing learning-environments (e.g., CSCL-environments) aimed at fostering complex learning-task performance. In this respect, the present study yields two important principles. First, to support the acquisition of a well-developed understanding of a domain, instruction should gradually increase the complexity of the domain; introducing qualitative representations before quantitative ones (Mulder et al. 2011). Second, to support application of that understanding, instruction should allow for constructing representations congruent with the tasks to be carried out (Schnotz and Kürschner 2008).

There are, however, multiple reasons to assume that these design principles do not automatically apply to other domains, learning tasks and settings. To address this, several remarks and suggestions for future research are provided.

First, whereas many domains (e.g., business-economics, meteorology, physics) require multiple problem representations, the effects of a particular design depend on the

characteristics of the learning task and the involved knowledge domains (Elen and Clarebout 2007). When designing tools and/or learning environments, one should take this carefully into account. To address this, educators and instructional designers should gain insight into the specifics of the learning tasks by conducting a learning-task analysis (Anderson and Krathwohl 2001). If analysis reveals that the entire task needs to be sequenced in part-tasks, their required domain-specific perspectives need to be determined. Based on these insights, the sequence and the demands of the part-tasks can be specified and part-task congruent tools can be developed.

Second, the effects of the design principles were studied in a collaborative learning setting. This strategy makes it hard to determine what actually caused the beneficial effect: constructing part-task congruent representations and/or discussing them with team members? Thus, from this study it may be concluded only that representing the domain in a part-task congruent manner and discussing those representations can foster complex learning. Since other studies have shown that individual learning-task performance can also be guided by providing representations or letting students construct their own (Larkin and Simon 1987; Vekiri 2002), it might be the case that the design principles can also be beneficially applied in this setting. Future research might address this by examining whether individual learners can also be guided when carrying out complex learning tasks.

Finally, guiding complex learning within a specific course may be beneficial, but is this also the case when the same design principles are employed throughout the whole curriculum? In other words, how much should learners' cognitive behavior be structured and when should this be more problematized (Reiser 2004)? There seems to be a delicate balance between the two since learners (i.e., novices) encounter difficulties when carrying out complex tasks without guidance when on a curricular level they should be able to perform such tasks on their own. Perhaps educators and instructional designers can address this by gradually diminishing the amount of instructional support (fading; Kollar et al. 2007). With regard to representational scripting, this might be achieved by letting learners carry out multiple but comparable tasks and decreasing the amount of guidance step-by-step. It would be interesting to study which aspect of representational scripting should be decreased first, sequencing the part-task and their task demands or part-task congruent support?

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Ainsworth, S. (2006). DeFT: A conceptual framework for considering learning with multiple representations. *Learning and Instruction, 16*, 183–198.
- Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. New York: Longman.
- Chi, M. T. H. (1997). Quantifying qualitative analyses of verbal data: A practical guide. *Journal of the Learning Sciences, 6*, 271–315.
- Cicchetti, D. V., Lee, C., Fontana, A. F., & Dowds, B. N. (1978). A computer program for assessing specific category rater agreement for qualitative data. *Educational and Psychological Measurement, 38*, 805–813.
- Corbalan, G., Kester, L., & Van Merriënboer, J. J. G. (2009). Selecting learning tasks: Effects of adaptation and shared control on learning efficiency and task involvement. *Contemporary Educational Psychology, 33*, 733–756.

- Cox, R. (1999). Representation construction, externalised cognition and individual differences. *Learning and Instruction*, 9, 343–363.
- Cress, U. (2008). The need for considering multilevel analysis in CSCL research: An appeal for the use of more advanced statistical methods. *International Journal of Computer-Supported Collaborative Learning*, 3, 69–84.
- De Simone, C., Schmid, R. F., & McEwan, L. A. (2001). Supporting the learning process with collaborative concept mapping using computer-based communication tools and processes. *Educational Research and Evaluation*, 7, 263–283.
- 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, The Netherlands: Open Universiteit Nederland.
- Elen, J., & Clarebout, G. (2007). Supporting learning: Increasing complexity? *Computers in Human Behavior*, 23, 1162–1166.
- Erkens, G. (2005). *Multiple episode protocol analysis (MEPA). Version 4.10*. The Netherlands: Utrecht University.
- Fischer, F., Bruhn, J., Gräsel, C., & Mandl, H. (2002). Fostering collaborative knowledge construction with visualization tools. *Learning and Instruction*, 12, 213–232.
- Frederiksen, J. R., & White, B. Y. (2002). Conceptualizing and constructing linked models: Creating coherence in complex knowledge systems. In P. Brna, M. Baker, K. Stenning, & A. Tiberghien (Eds.), *In the role of communication in learning to model* (pp. 69–96). Mahwah, NJ: Lawrence Erlbaum Associates.
- Hilbert, T. S., & Renkl, A. (2008). Concept mapping as a follow-up strategy to learning from texts: What characterizes good and poor mappers? *Instructional Science*, 36, 53–73.
- Hmelo-Silver, C. E., Chernobilsky, E., & Jordan, R. (2008). Understanding collaborative learning processes in new learning environments. *Instructional Science*, 36, 409–430.
- Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). Scaffolding and achievement in problem-based and inquiry learning: A response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist*, 42, 99–107.
- Janssen, J., Erkens, G., Kirschner, P. A., & Kanselaar, G. (2010a). Effects of representational guidance during computer-supported collaborative learning. *Instructional Science*, 38, 59–88.
- Janssen, J., Kirschner, F., Erkens, G., Kirschner, P. A., & Paas, F. (2010b). Making the black box of collaborative learning transparent: Combining process-oriented and cognitive load approaches. *Educational Psychology Review*, 22, 139–154.
- Jaspers, J. G. M., Broeken, M., & Erkens, G. (2005). *Virtual collaborative research institute (VCRI). Version 2.2*. Utrecht, The Netherlands: Utrecht University.
- Jonassen, D. H. (2003). Using cognitive tools to represent problems. *Journal of Research on Technology in Education*, 35, 362–381.
- Kenny, D. A., Kashy, D. A., & Cook, W. L. (2006). *Dyadic data analysis*. New York/London: The Guilford Press.
- Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 4(2), 75–86.
- Kollar, I., Fischer, F., & Slotta, J. D. (2007). Internal and external scripts in computer-supported collaborative inquiry learning. *Learning and Instruction*, 17, 708–721.
- Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11, 65–100.
- Löhner, S., Van Joolingen, W. R., & Savelsbergh, E. R. (2003). The effect of external representations on constructing computer models of complex phenomena. *Instructional Science*, 31, 395–418.
- Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American Psychologist*, 59(1), 14–19.
- Mercer, N., Littleton, K., & Wegerif, R. (2004). Methods for studying the processes of interaction and collaborative activity in computer-based educational activities. *Technology, Pedagogy and Education*, 13, 195–212.
- Moos, D. C., & Azevedo, R. (2008). Monitoring, planning, and self-efficacy during learning with hypermedia: The impact of conceptual scaffolds. *Computers in Human Behavior*, 24, 1686–1706.
- Mulder, Y. G., Lazonder, A. W., & De Jong, T. (2011). Comparing two types of model progression in an inquiry learning environment with modelling facilities. *Learning and Instruction*, 21, 614–624.
- Ploetzner, R., Fehse, E., Kneser, C., & Spada, H. (1999). Learning to relate qualitative and quantitative problem representations in a model-based setting for collaborative problem solving. *Journal of the Learning Sciences*, 8, 177–214.

- Reiser, B. J. (2004). Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *Journal of the Learning Sciences, 13*, 273–304.
- Scaife, M., & Rogers, Y. (1996). External cognition: How do graphical representations work? *International Journal of Human-Computer Studies, 45*, 185–213.
- Schnotz, W., & Kürschner, C. (2008). External and internal representations in the acquisition and use of knowledge: visualization effects on mental model construction. *Instructional Science, 36*, 175–190.
- Simon, H. L., Langley, P. W., & Bradshaw, G. (1981). Scientific discovery as problem solving. *Syntheses, 47*, 1–27.
- Slof, B., Erkens, G., Kirschner, P. A., Janssen, J., & Phielix, C. (2010a). Fostering complex learning-task performance through scripting student use of computer supported representational tools. *Computers & Education, 55*, 1707–1720.
- Slof, B., Erkens, G., Kirschner, P. A., Jaspers, J. G. M., & Janssen, J. (2010b). Guiding students' online complex learning-task behavior through representational scripting. *Computers in Human Behavior, 26*, 927–939.
- Strijbos, J.-W., Martens, R. L., Prins, F. J., & Jochems, W. M. G. (2006). Content analysis: What are they talking about? *Computers & Education, 46*, 29–48.
- Stull, A. T., & Mayer, R. E. (2007). Learning by doing versus learning by viewing: Three experimental comparisons of learner-generated versus author-provided graphic organizers. *Journal of Educational Psychology, 99*, 808–820.
- Van Bruggen, J. M., Boshuizen, H. P. A., & Kirschner, P. A. (2003). A cognitive framework for cooperative problem solving with argument visualization. In P. A. Kirschner, S. J. Buckingham-Shum, & C. S. Carr (Eds.), *Visualizing argumentation: Software tools for collaborative and educational sense-making* (pp. 25–47). London: Springer.
- Van Merriënboer, J. J. G., & Kirschner, P. A. (2007). *Ten steps to complex learning. A systematic approach to four-component instructional design*. New Jersey: Lawrence Erlbaum Associates.
- Vekiri, I. (2002). What is the value of graphical displays in learning? *Educational Psychology Review, 14*, 261–312.