Learning science by creating models

Doctoral committee

Chair:	Prof. dr. K.I. van Oudenhoven-Van der Zee
Promotor:	Prof. dr. A.J.M. de Jong
Assistant Promotor:	Dr. A.W. Lazonder
Members:	Prof. dr. W.R. van Joolingen
	Prof. dr. P.J.C. Sleegers
	Prof. dr. J.J.G. van Merriënboer
	Prof. dr. M. Valcke
	Prof. dr. R. Rikers
	Dr. L. Kester

ISBN: 978-90-365-3340-9

LEARNING SCIENCE BY CREATING MODELS

PROEFSCHRIFT

ter verkrijging van

de graad van doctor aan de Universiteit Twente,

op gezag van de rector magnificus,

prof. dr. H. Brinksma

volgens besluit van het College voor Promoties

in het openbaar te verdedigen

op donderdag 19 april 2012 om 16:45 uur

door

Yvonne Geertruida Mulder

geboren op 26 juli 1983

te Zwolle

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. A.J.M. de Jong

En assistent-promotor:
Dr. A.W. Lazonder

Dankwoord

Een proefschrift maak je niet alleen. Er zijn vele mensen, zowel binnen als buiten de werkomgeving, die de afgelopen jaren in Twente voor mij tot een uiterst plezierige tijd hebben gemaakt. Uit ruimte-economische overwegingen kan ik hier helaas niet iedereen bij naam noemen, maar ieders bijdrage hieraan was belangrijk en heb ik bijzonder gewaardeerd!

In de eerste plaats wil ik mijn promotor en copromotor ontzettend bedanken voor de plezierige samenwerking tijdens de afgelopen jaren. Ard, als dagelijks begeleider was jij het dichtst betrokken bij de totstandkoming van dit proefschrift. Jouw ideeën, kritische kijk en duidelijke sturing hebben mede voor dit eindresultaat zorg gedragen. Ik wil je bedanken dat je deur altijd open stond, maar nog meer voor die keren dat je juist even van je liet horen als ik te lang niet langs was geweest. Ton, ik wil je danken voor de gelegenheid die je me hebt gegeven om binnen jouw onderzoeksgroep onderzoek te doen. Je gaf me de vrijheid mijn eigen weg te bewandelen, maar hield wel een vinger aan de pols. Hartelijk dank voor alles wat ik heb mogen leren.

Vervolgens wil ik al mijn collega's bedanken die de dagelijkse werkzaamheden tijdens het promoveren kleur hebben gegeven. Ik noem niet iedereen bij naam, maar hoop van harte dat allen zich hier aangesproken voelen: bedankt voor de collegiale sfeer en de gezellige (thee)pauzes! In het bijzonder wil ik hier noemen: Wout mijn kamergenoot en paranimf. Bedankt voor de gezelligheid en goede gesprekken in de afgelopen jaren en succes nog met het afronden van jouw promotie! Verder gaat mijn dank uit naar Wilco, Frank, Jan, Alieke, Sylvia en Mieke voor de onmisbare hulp bij de experimenten; naar Larissa en Daphne voor de secretariële assistentie en naar Jakob, Anjo en Lars voor de (vele) technische ondersteuning.

Ook ben ik dank verschuldigd aan de promovendi van de ProIST bijeenkomsten en de ICO cursussen voor het meedenken over mijn onderzoek. Fijn dat in een vaak wat minder formele setting alle serieuze en minder serieuze zaken rondom promoveren bespreekbaar waren. Bert, jou heb ik ook via een ICO cursus leren kennen, bedankt voor de inhoudelijke discussies en fijne vriendschap.

De studies beschreven in dit proefschrift hadden niet tot stand kunnen komen zonder de medewerking van scholen. Ik wil de docenten, ICT ondersteuning en leerlingen van het Bonhoeffer College, CSG Het Noordik en SG de Grundel bedanken voor hun enthousiaste medewerking tijdens mijn studies.

Ten slotte een speciaal woord van dank voor al mijn familie en vrienden voor hun betrokkenheid in de afgelopen jaren, voor hun oprechte interesse in mijn onderzoek en voor het soms vooral NIET informeren naar de voortgang van mijn proefschrift waardoor ze mij het waardevolle gevoel gaven dat mijn leven uit meer bestond dan werk alleen! Enkele personen die ik toch nog bij naam wil noemen zijn:

Karoline, mijn lieve zus, bedankt dat je altijd enthousiast bent en zorg draagt voor de broodnodige ontspanning. Peter, mijn grote broer en paranimf, bedankt dat je er altijd voor me bent als ik je nodig heb (niet alleen bij het eindeloos spelen voor helpdesk). Fijn dat je ook op mijn promotie aan mijn zijde wilt staan.

Lieve papa en mama, ik wil jullie graag bedanken voor alle steun, hulp en liefde die jullie mij altijd hebben gegeven.

Table of Contents

Chapter 6: Summary and general discussion

Chapter 1

General introduction

Introduction

“I hear and I forget. I see and I remember. I do and I understand”. This ancient Confucius quotation reflects the basic premise of many contemporary approaches to education. The idea that learners should not passively receive information but instead must be encouraged to actively construct knowledge is widely accepted (Cobb, 1994). Inquiry-based learning, which has its roots in the work by Dewey (1938) and Bruner (1991), is one example of how the concept of active, self-directed knowledge construction can be implemented in high school science classrooms. Inquiry learning, in short, requires students to learn science by doing science. Recent European reports advocate that improvements in science education should be brought about through inquiry-based approaches, as such a pedagogy is more likely to increase students’ interest and attainment levels (Osborne & Dillon, 2008; Rocard et al., 2007). A more elaborate definition of inquiry learning is given by the National Science Foundation (2000, p. 2), which characterized inquiry learning as "An approach to learning that involves a process of exploring the natural or material world, and that leads to asking questions, making discoveries, and rigorously testing those discoveries in the search for new understanding".

The inquiry learning process has been captured in various phase-like models that, despite their idiosyncratic differences, share at least three iterative activities: hypothesizing, experimenting, and evaluating evidence (cf. Klahr & Dunbar, 1988; Zimmerman, 2007). After an initial orientation phase, where students get acquainted with the phenomenon they will be investigating (e.g., gravity), students formulate hypotheses (e.g., I think that the weight of an object influences the speed with which it drops). In order to test these hypotheses, students can design experiments. An experiment to test the exemplary hypothesis would be to drop a heavy and light ball at the same time. Following the experiment, students have to evaluate the data (the balls landed at the same time) in order to draw conclusions (weight of an object does not influence the speed with which it drops). These inquiry activities are iterative and cyclical by nature in that conclusions generally lead to new hypotheses (e.g., perhaps the size of a ball influences the speed with which it drops), which in turn lead to new experiments, new conclusions, and so on.

Nowadays, computer-supported inquiry learning environments offer resources to facilitate inquiry learning. Computer simulations have long since lain at the heart of these environments, and currently these simulations are increasingly being supplemented with opportunities for students to build computer models of the phenomena they are investigating via the simulation. As in authentic scientific inquiry, modelling is considered an integral part of the inquiry learning process in that students can build computer models to express their understanding of the relation between variables (de Jong & van Joolingen, 2008; van Joolingen, de Jong, Lazonder, Savelsbergh, & Manlove, 2005; White, Shimoda, & Frederiksen, 1999).

The virtues of this pedagogy, in which inquiry learning and computer modelling are combined, was investigated in the four studies that comprise this thesis. As this synergistic approach to science learning is relatively new and little documented in the research literature, its key characteristics are introduced in the section below.

Learning with computer simulations and models

Static diagrams in books and on blackboards do not convey the intermittent nature of flows and of the varying rates of change found in dynamic systems (Riley, 1990). Simulations, models, and animations, on the other hand, add a temporal dimension to the representation of a phenomenon. As both simulations and models provide students with the additional possibility to control the flow of dynamic systems through time, they have been recognized as powerful tools to learn about dynamic scientific phenomena that are otherwise too costly, too dangerous, or too difficult to observe (Eysink et al., 2009).

Simulation-based inquiry learning

With the increasing availability of computers, the use of computer simulations in education has greatly expanded. The interest in simulation-based learning has increased accordingly as it has been the focus of 510 studies into science education over the past decade (Rutten, van Joolingen, & van der Veen, 2012). Compared to traditional, more expository forms of instruction, several studies have shown that learning with simulations is more effective for promoting science content knowledge, developing process skills, and facilitating conceptual change (e.g., Alfieri, Brooks, Aldrich, & Tenenbaum, 2011; Eysink et al., 2009; Marušić & Sliško, 2011; Scalise et al., 2011; Smetana & Bell, 2012). These promising results, however, only hold when the inquiry process is adequately structured and scaffolded.

Simulation-based inquiry learning enables students to infer the characteristics of the model underlying the simulation through experimentation (de Jong & van Joolingen, 1998). The two simulations that were used in the studies of this thesis are shown in Figure 1.1. Both simulations represent an electrical circuit containing a power source, two devices that act as resistors, and a capacitor. Participants in the first experimental study (Chapter 2) received the simulation that is depicted in the left pane of Figure 1.1, which models the influence of resistance on the charging

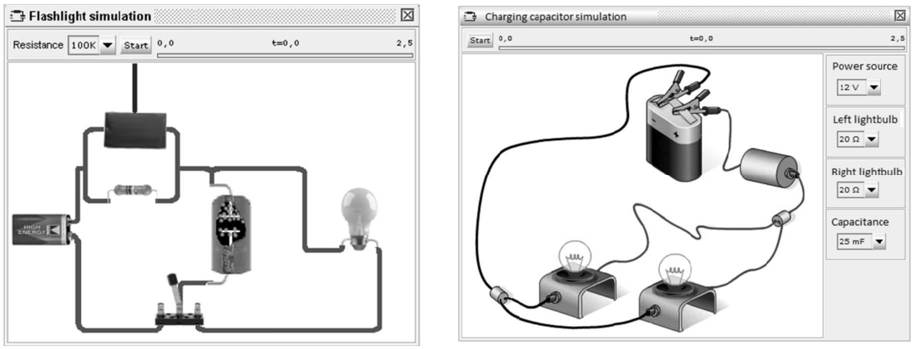

Figure 1.1. Screen capture of a simulation with one input variable (left pane) and four input variables (right pane).

of the capacitor. The right pane of Figure 1.1 displays the simulation that was used in Chapters 3, 4, and 5. This simulation had more input parameters (power source, left light bulb, right light bulb and capacitance), that enabled students to examine the direct influence of the components in the simulation in greater detail.

Both simulations enabled students to engage in the processes of hypothesis formation, experimentation, and evaluating evidence. Students could *hypothesize* about the effect of the resistance on the charging of the capacitor (e.g., I think that the resistor value influences the charge after loading). In order to test these hypotheses, students could *design and conduct experiments* by assigning resistance values in the simulation. An experiment to test the exemplary hypothesis would be to run the simulation twice: once with a low resistor value and once with a high resistor value. Following the experiment, students have to inspect the data (read from a table or a graph that the charge after loading was the same in both simulation runs) in order to draw conclusions (the resistor value does not influence the charge after loading), that can lead to new hypotheses (e.g., I think that the resistor value influences the capacitors' charging speed), which in turn leads to new experiments, new conclusions and so on until a full understanding of charging capacitors in an electrical circuit is reached.

Learning by modelling

Coll and Lajium (2011) state three principal purposes of modelling in the sciences as reported in the science education literature: (a) to produce simpler forms of objects or concepts; (b) to provide stimulation for learning or concept generation, and thereby support the visualization of some phenomenon; and (c) to provide explanations for scientific phenomena. Students benefit from modelling as it allows them to develop a deep understanding of difficult domain concepts, as well as a better understanding of science processes and the nature of science (Campbell, Zhang, & Neilson, 2011). Creating artefacts such as computer models is assumed to improve learning because students have to explicate their newly acquired understanding, which makes them aware of knowledge gaps they had not noticed before (Kafai & Resnick, 1996; Kolloffel, Eysink, & de Jong, 2010; Rocard et al., 2007). Besides computer models (hereafter: models), these artefacts can take several forms; examples include drawings, concept maps, physical objects, podcasts, and 3D-sketches. Yet models have the advantage of adding a temporal dimension to the constructed artefact, and thus form the natural counterpart of simulations that have a temporal dimension too. Furthermore, constructing models is in keeping with inquiry as creating and using models are common practice in authentic scientific inquiry.

Nowadays, several learning environments offer modelling platforms. Some of the more well-known examples include STELLA (Steed, 1992), Model-It (Jackson, Stratford, Krajcik, & Soloway, 1994), Co-Lab (van Joolingen et al., 2005) and, more recently, SCY-Lab (de Jong et al., 2010). The Co-Lab learning environment was used in the studies of this thesis. This choice was based on practical reasons: at the start of this thesis research project, Co-Lab was the only environment that combined simulations with modelling facilities.

The Co-Lab modelling tool makes use of the system dynamics modelling language (Forrester, 1961). As shown in Figure 1.2, system dynamics models consist of graphical elements that are linked by relation arrows. The model in this figure shows how salary and contribution determine monthly income and expenses respectively, which in turn influence the bank account balance.

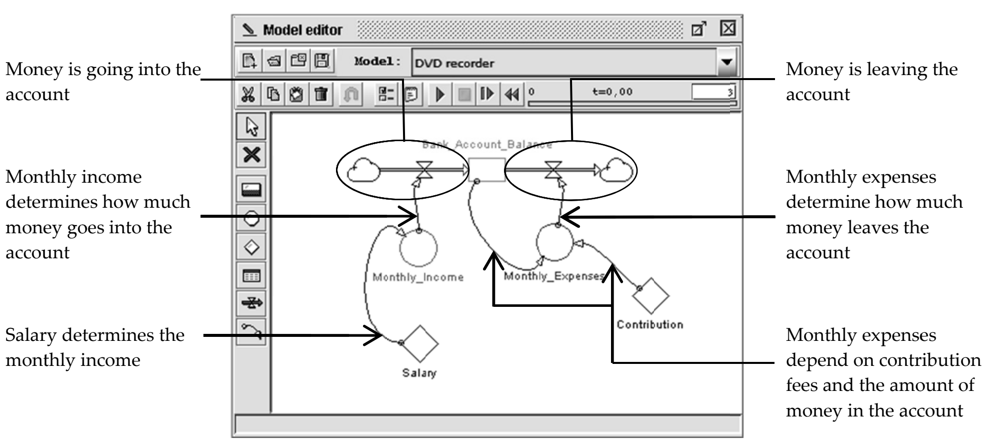

Figure 1.2. Annotated screen capture of the Co-Lab modelling tool.

Inquiry and modelling: the integrated approach to science learning

When involved in modelling, students ideally go through four distinguishable stages: (1) model sketching, (2) model specification, (3) data interpretation, and (4) model revision (cf. Hogan & Thomas, 2001). Combining these stages with the inquiry learning activities outlined in the previous section provides a description of the integrated approach to science learning (cf. van Joolingen et al., 2005). When students have no prior knowledge of the domain, they carry out exploratory experiments to gain an initial understanding of the phenomena. Students with prior knowledge can skip this step and immediately start sketching a model outline to express their understanding of the phenomena. Subsequently students form hypotheses which they can investigate through the simulation. The results of these experiments are then used to transform the model sketch into a runnable model by specifying the relations between the variables in the model. Accordingly, the model can be conceived of as a hypothesis. During data interpretation, learners compare their model to data from the simulation, which during the conclusion phase, feeds their decisions to revise the model.

However, in practice students have difficulty with both inquiry learning and modelling, which challenges the educational effectiveness of the integrated approach to science learning. For example, students are unable to infer hypotheses from (simulation) data, design inconclusive experiments, show inefficient experimentation behaviour, and ignore incompatible data (for extensive reviews, see de Jong & van Joolingen, 1998; Zimmerman, 2007). Regarding modelling, Hogan and Thomas (2001) noticed that students often fail to engage in dynamic iterations between examining output and revising models, and merely use output at the end of a session to check if the model's behaviour matches their expectations. A related problem concerns the students' lack of persistence in debugging their model to fine-tune its behaviour (Stratford, Krajcik, & Soloway, 1998).

These findings suggest that students' difficulties with inquiry and modelling both lie at a conceptual level. Most students manage to design and conduct experiments with a simulation; inferring knowledge from these experiments appears to be the major source of difficulty. Likewise, students are capable of building syntactically correct models, but often fail to relate their knowledge of phenomena to those models (Sins, Savelsbergh, & van Joolingen, 2005). As this ineffective behaviour is a serious obstacle to learning, students might benefit from additional support during their inquiry and modelling practices. The studies reported in this thesis sought to establish the need for and effects of various types of support. The general research question that guided these investigations was:

How can learning with computer simulations and models be improved by embedded support?

Thesis outline

The general research question was addressed in four empirical studies. The study in Chapter 2 concerned an empirical assessment of high school students' need for support. Toward this end, a target group of domain novices was compared to two more knowledgeable reference groups. Comparisons of the groups' behaviour and performance were conducted in order to determine which inquiry and modelling skills would require additional support. The studies reported in Chapter 3 and 4 investigated whether model progression (i.e., gradually increasing task complexity) could help compensate for these observed skill deficiencies. The study depicted in Chapter 3 aimed to offer empirical evidence regarding the instructional efficacy of model progression per se. Two types of model progression were examined and compared to a control group that received no additional support. Chapter 4 describes a study that aimed to further investigate the effects of model progression by examining the influence of learning path restrictions. In this study, the most effective type of model order progression from Chapter 3 was compared with two variants that had either more liberal or more strict requirements to progress to more complex subject matter. The study in Chapter 5 explored whether complementing model progression with worked examples would further enhance students' inquiry and modelling performance and learning. Finally, Chapter 6 gives a summary of the findings, presents conclusions drawn from the four studies, and discusses the theoretical and practical implications of the research.

References

Alfieri, L., Brooks, P. J., Aldrich, N. J., & Tenenbaum, H. R. (2011). Does discovery-based instruction enhance learning? *Journal of Educational Psychology, 103*, 1-18. doi: 10.1037/a0021017

Campbell, T., Zhang, D. H., & Neilson, D. (2011). Model based inquiry in the high school physics classroom: An exploratory study of implementation and outcomes. *Journal of Science Education and Technology, 20*, 258-269. doi: 10.1007/s10956-010-9251-6

Cobb, P. (1994). Constructivism. In T. Husén & T. N. Postlethwaite (Eds.), *The international encyclopedia of education* (2 ed.). Oxford: Pergamon.

Coll, R. K., & Lajium, D. (2011). Modeling and the Future of Science Learning. In M. S. Khine & I. M. Saleh (Eds.), *Models and modeling cognitive tools for scientific enquiry*. Dordrecht: Springer.

de Jong, T. (1991). Learning and instruction with computer simulations. *Education and Computing, 6*, 217-230.

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*, 179-201. doi: 10.3102/00346543068002179

de Jong, T., & van Joolingen, W. R. (2008). Model-facilitated learning. In M. Spector, M. D. Merrill, J. van Merriënboer & M. P. Driscoll (Eds.), *Handbook of research on educational communications and technology* (pp. 457-468). New York: Lawrence Erlbaum Associates.

de Jong, T., van Joolingen, W. R., Giemza, A., Girault, I., Hoppe, U., Kindermann, J., . . . van der Zanden, M. (2010). Learning by creating and exchanging objects: The SCY experience. *British Journal of Educational Technology, 41*, 909-921. doi: 10.1111/j.1467-8535.2010.01121.x

Dewey, J. (1938). *Logic: The theory of inquiry*. New York: Holt and Co.

Eysink, T. H. S., de Jong, T., Berthold, K., Kolloffel, B., Opfermann, M., & Wouters, P. (2009). Learner performance in multimedia learning arrangements: An analysis across instructional approaches. *American Educational Research Journal, 46*, 1107-1149. doi: 10.3102/0002831209340235

Forrester, J. (1961). *Industrial dynamics*. Waltham, Massachusetts: Pegasus Communications.

Hogan, K., & Thomas, D. (2001). Cognitive comparisons of students' systems modeling in ecology. *Journal of Science Education and Technology, 10*, 319-345. doi: 10.1023/A:1012243102249

Jackson, S. L., Stratford, S. J., Krajcik, J., & Soloway, E. (1994). Making dynamic modeling accessible to precollege science students. *Interactive Learning Environments, 4*, 233 - 257.

Kafai, Y. B., & Resnick, M. (Eds.). (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world.* Mawhaw, NJ: Lawrence Erlbaum Associates.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12*, 1-48. doi: 10.1207/s15516709cog1201_1

Kolloffel, B., Eysink, T. H. S., & de Jong, T. (2010). The influence of learner-generated domain representations on learning combinatorics and probability theory. *Computers in Human Behavior, 26*, 1-11. doi: 10.1016/j.chb.2009.07.008

Marušić, M., & Sliško, J. (2011). Influence of three different methods of teaching physics on the gain in students' development of reasoning. *International Journal of Science Education, 34*, 301-326. doi: 10.1080/09500693.2011.582522

National Science Foundation. (2000). *Inquiry: Thoughts, Views, and Strategies for the K-5 Classroom*. In Foundations (Ed.), (Vol. 2).

Osborne, J., & Dillon, J. (2008). *Science education in Europe: Critical reflections.* London: The Royal Society.

Riley, D. (1990). Learning about systems by making models. *Computers and Education, 15*, 255-263. doi: 10.1016/0360-1315(90)90155-Z

Rocard, M., Csermely, P., Jorde, D., Lenzen, D., Walberg-Henriksson, H., & Hemmo, V. (2007). *Science education now: A renewed pedagogy for the future of Europe*. Brussels: Directorate-general for research.

Rutten, N., van Joolingen, W. R., & van der Veen, J. T. (2012). The learning effects of computer simulations in science education. *Computers and Education, 58*, 136-153. doi: 10.1016/j.compedu.2011.07.017

Scalise, K., Timms, M., Moorjani, A., Clark, L., Holtermann, K., & Irvin, P. S. (2011). Student learning in science simulations: Design features that promote learning gains. *Journal of Research in Science Teaching, 48*, 1050-1078. doi: 10.1002/tea.20437

Sins, P. H. M., Savelsbergh, E. R., & van Joolingen, W. R. (2005). The difficult process of scientific modeling: An analysis of novices' reasoning during computer-based modeling. *International Journal of Science Education, 14*, 1695-1721. doi: 10.1080/09500690500206408

Smetana, L. K., & Bell, R. L. (2012). Computer simulations to support science instruction and learning: A critical review of the literature. *International Journal of Science Education*, Advance online publication. doi: 10.1080/09500693.2011.605182

Steed, M. (1992). Stella, a simulation construction kit: cognitive process and educational implications. *Journal of Computers in Mathematics and Science Teaching, 11*, 39-52.

Stratford, S. J., Krajcik, J., & Soloway, E. (1998). Secondary students' dynamic modeling processes: Analyzing, reasoning about, synthesizing, and testing models of stream ecosystems. *Journal of Science Education and Technology, 7*, 215. doi: 10.1023/A:1021840407112

van Joolingen, W. R., de Jong, T., Lazonder, A. W., Savelsbergh, E. R., & Manlove, S. (2005). Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning. *Computers in Human Behavior, 21*, 671-688. doi: 10.1016/j.chb.2004.10.039

White, B. Y., Shimoda, T. A., & Frederiksen, J. R. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. *International Journal of Artificial Intelligence in Education, 10*, 151-182.

Zimmerman, C. (2007). The development of scientific thinking skills in elementary and middle school. *Developmental Review, 27*, 172-223. doi: 10.1016/j.dr.2006.12.001

Chapter 2

Finding out how they find it out: An empirical analysis of inquiry learners' need for support[1]

Abstract

Inquiry learning environments increasingly incorporate modelling facilities for students to articulate their research hypotheses and (acquired) domain knowledge. This study compared performance success and scientific reasoning of university students with high prior knowledge (n = 11), students from senior high school (n = 10), and junior high school (n = 10) with intermediate and low prior knowledge respectively, in order to reveal domain novice's need for support in such environments. Results indicated that the scientific reasoning of both groups of high school students was comparable to that of the experts. As high school students achieved significantly lower performance success scores, their expert-like behaviour was rather ineffective; qualitative analyses substantiated this conclusion. Based on these findings, implications for supporting domain novices in inquiry learning environments are advanced.

[1] Mulder, Y. G., Lazonder, A. W., & de Jong, T. (2010). Finding out how they find it out: An empirical analysis of inquiry learners' need for support. *International Journal of Science Education*, *32*, 2033-2053. doi: 10.1080/09500690903289993 (with minor modifications).

Introduction

Computer-supported inquiry learning environments essentially enable students to learn science by doing science, offering resources to develop a deep understanding of a domain by engaging in scientific reasoning processes such as hypothesis generation, experimentation, and evidence evaluation. The central aim of this investigative learning mode is twofold: students should develop domain knowledge and proficiency in scientific inquiry (cf. Gobert & Pallant, 2004). Unfortunately the educational advantages of inquiry learning are often challenged by students' poor inquiry skills (e.g., de Jong & van Joolingen, 1998). Researchers and designers therefore often attempt to compensate for students' skill deficiencies by offering support such as proposition tables to help generate hypotheses (Shute, Glaser, & Raghavan, 1989), adaptive advice for extrapolating knowledge from simulations (Leutner, 1993), or regulative scaffolds to assist students in planning, monitoring, and evaluating their inquiry (Davis & Linn, 2000; Manlove, Lazonder, & de Jong, 2006)

Although much has been learned from these approaches, the empirical foundations underlying the contents of these support tools often remain hidden to the public eye. The work of Quintana et al. (2004) forms a notable exception. They argued that more insight into the specific problems students face is called for, and accordingly based their scaffolding framework on a descriptive analysis of students' inquiry learning problems. Yet even this well-documented framework lacks a specific frame of reference: if anything, there is an implicit reference to expert behaviour as yardstick of proficiency.

This study therefore sought to gain insight into students' scientific reasoning skill deficiencies by contrasting domain novices' inquiry behaviour and performance to that of a considerably more knowledgeable reference group (hereafter: experts). A group of students with intermediate levels of prior knowledge was included in this comparison to shed more light on the developmental trajectories of students' scientific reasoning and domain knowledge. Before elaborating the design of the study, a brief overview of the literature is given in order to contextualize the design rationale. This overview starts from classic novice-expert literature and results in a descriptive framework of the core scientific reasoning processes.

Theoretical background

Novice-expert differences have been studied extensively in the field of problem solving. This research has identified key characteristics of expert performance, some of which were found to be robust and generalizable across domains. In short, problem solving research has shown that people who have developed expertise in a certain area mainly excel within that area, perceive large meaningful patterns in their domain of expertise, perform fast (even though they spend a great deal of time analysing a problem), and have superior short-term and long-term memory. Experts also represent a problem in their domain at a deeper, more principled level than novices do and have strong self-monitoring skills (Bransford, Brown, & Cocking, 2002; Chi, Glaser, & Farr, 1988).

These general characteristics, although informative, are not specific enough to guide instructional designers and science educators in determining what exactly their support should focus on. A further complicating issue is that novice-expert differences in problem solving do not necessarily generalize to inquiry learning. According to Batra and Davis (1992), most problem solving tasks require participants to find a unique correct solution. In inquiry learning this search for a single optimal outcome (often referred to as an engineering approach) is generally considered less effective in facilitating students' understanding of a domain than a so-called science model of experimentation (Schauble, Klopfer, & Raghavan, 1991). Performing an inquiry task effectively and efficiently might thus require different skills and strategies than proficient problem solving does. As a result, the general instructional implications from problem solving research should be substantiated by, or supplemented with, insights gleaned from novice-expert differences in inquiry learning.

Inquiry learning attempts to mimic authentic scientific inquiry by engaging students in processes of orientation, hypothesis generation, experiment design, and data interpretation to reach conclusions (Shrager & Klahr, 1986; Zimmerman, 2007). While some have argued that the inquiry tasks given to students in schools evoke different cognitive processes than the ones employed in real scientific research (Chinn & Malhotra, 2002), the advancement of computer technology has significantly narrowed this gap. Contemporary electronic learning environments offer a platform for students to examine scientific phenomena through computer simulations. These environments increasingly provide opportunities for students to build computer models of the phenomena they are investigating. As in authentic scientific inquiry, modelling is considered an integral part of the inquiry learning process. Students can use models to express their understanding of a relation between variables (Jackson, Stratford, Krajcik, & Soloway, 1994; White, Shimoda, & Frederiksen, 1999); these propositions can be tested by running the model; evidence evaluation then occurs by weighting model output against prior knowledge or the data from the simulation. These comparisons yield further insight into the phenomenon and assist students in generating new hypotheses.

The effectiveness and efficiency with which students perform these processes can be expected to differ as function of their level of domain expertise. In the present research, Klahr and Dunbar's (1988) SDDS model was used to describe and explain these differences. This descriptive framework captures the core scientific reasoning processes and is sensitive to students' evolving domain knowledge. SDDS conceives of scientific reasoning as a search in two problem spaces (hence its name: Scientific Discovery as Dual Search): the hypothesis space and the experiment space. The former space comprises the hypotheses a learner can generate during the inquiry process; the latter consists of all possible experiments that can be conducted with the equipment at hand. Search in the hypothesis space is guided by either prior knowledge or experimental results. Search in the experiment space can be guided by the current hypothesis; in case learners do not have a hypothesis they can search the experiment space for exploratory experiments that will help them formulate new hypotheses.

According to the SDDS model, inquiry learning consists of three iterative processes: hypothesizing, experimenting, and evaluating evidence. The way students perform these processes is assumed to depend on their knowledge of the task domain. Students with domain expertise can generate hypotheses from prior knowledge and then test their hypotheses by conducting experiments (i.e., a 'theory-driven' approach). After experimenting, students can evaluate their hypotheses against the cumulative experimental results and prior knowledge. Evaluation has three possible outcomes: the current hypothesis can either be accepted, rejected, or considered further. Depending on this evaluation the student may start a new search for hypotheses, continue investigating the current hypothesis (which generally involves some alteration), or end the inquiry. Students without domain expertise cannot generate initial hypotheses from prior knowledge. They have to search the experiment space for a series of exploratory experiments (i.e., a 'data-driven' approach). Once performed and evaluated, these experiments may help students to formulate an initial hypothesis, which can then be tested through experimentation.

Research has generally confirmed the alleged influence of domain knowledge on scientific reasoning. The original study by Klahr and Dunbar (1988) provides evidence that prior knowledge reduces time on task and the number of experiments conducted. Performance success was independent of prior knowledge: all participants succeeded in discovering how an unknown function of an electronic device worked. Klahr and Dunbar also identified two distinct investigative strategies, a Theorist approach and an Experimenter approach. One of the key differences between the two was that Experimenters conduct more experiments than Theorists and that this extra experimentation is conducted without an explicit hypothesis statement (Klahr & Dunbar, 1988).

However, these results could not be replicated under more controlled circumstances. Wilhelm and Beishuizen (2003) for instance compared learning activities and outcomes across a concrete and abstract inquiry task. These tasks were designed so that participants had no prior knowledge of the abstract task and ample prior knowledge of the concrete task. Participants were found to perform better when their task was embedded in a concrete context. Compared to the students in the concrete condition, students in the abstract condition stated fewer hypotheses, but performed as many experiments (time on task was not assessed). Lazonder, Wilhelm, and Hagemans (2008) replicated these findings in a within-subject comparison. They too found that participants perform better on a concrete task with familiar content. Results also confirmed that participants generate more, and more specific hypotheses on the concrete task. The number of experiments was again comparable on both tasks. Lazonder et al. (2008) also confirmed the existence of two distinct investigative strategies. They argued that as individuals have little domain knowledge they are presumed to start off in a data-driven approach, meaning that they start experimenting without having formulated specific hypotheses, but gradually switch to a more theory-driven mode of experimentation. Individuals who do posses domain knowledge, in contrast, approach the task by generating and testing specific hypotheses, which is the Theorist approach.

These findings suggest that, although prior knowledge does not reduce the number of experiments per se, it does reduce the number of experiments not guided by a hypothesis. Students with prior knowledge thus engage in more theory-driven experimentation which leads to superior task performance. The latter part of this conclusion was corroborated by Lazonder, Wilhelm, and van Lieburg (2009), who found that the number of hypotheses stated by participants was a strong predictor of performance success. This study further showed that students learning by inquiry benefit little from knowledge of the meaning of variables per se, but it is the knowledge of the relations of the variables that is of pivotal importance.

In line with the previously mentioned studies, the research reported here investigated how prior domain knowledge influences students' scientific reasoning and performance in an inquiry task. In contrast to the previous studies, this study was designed as a novice-expert comparison that aimed to replicate and extend previous findings under more ecologically valid conditions. Toward this end the study utilized a genuine physics task that was situated in a realistic setting, and performed with an inquiry learning environment designed for secondary education –which stands in marked contrast to the fictitious small-scale inquiry tasks used in laboratory studies cited above. Another key difference with prior research is that modelling was treated as integral part of the inquiry process.

Toward this end the learning environment housed a modelling tool students could use to articulate their hypotheses and (acquired) domain knowledge.

Research design and hypotheses

This study compared scientific reasoning and performance success of low-level novices, high-level novices and experts on an inquiry task that involved modelling a charging capacitor. Low-level novices had no prior knowledge of the task content, but could induce this knowledge by interacting with a computer simulation so as to build a model of the capacitor. High-level novices were familiar with the physics laws that govern the behaviour of a charging capacitor, whereas the experts' knowledge of capacitors was well beyond the requirements for successful task completion.

In line with previous findings participants' prior domain knowledge was expected to influence their performance success and scientific reasoning. As participants could infer all knowledge by interacting with the learning environment, the quality of their final models was expected to be comparable and therefore independent of prior domain knowledge. However, it was expected that novices would need more time to create their models than experts.

Scientific reasoning was expected to differ as function of participants' prior domain knowledge. Low-level novices, in absence of prior domain knowledge, were expected to start off in a data-driven mode of inquiry and gradually shift to a more theory-driven approach, resulting in increasingly domain-specific hypotheses. High-level novices possessed some prior domain knowledge, and were therefore expected to approach the beginning of the task more theory driven than low-level novice. Still, high-level novices were expected to show an increase in their hypotheses' domain specificity. Experts on the other hand, were predicted to engage in theory-driven experimentation throughout their inquiry, expressing highly domain-specific hypotheses. As participants engaging in a data-driven approach will conduct more experiments than participants engaging in a theory-driven approach, a negative relationship was expected between prior domain knowledge and the number of conducted experiments.

Relatively many studies have been conducted investigating learners' evidence evaluation. This kind of research generally focuses on developmental differences and reasoning errors people make during evidence evaluation (for an extensive overview see Zimmerman, 2000). However, as the influence of prior domain knowledge on evidence evaluation has remained unexplored, this study does not start from an assumption regarding the process of evaluating evidence, and addressed this scientific reasoning process in an explorative way.

Method

Participants

Thirty-one Dutch students participated in this study. They were selected for their levels of prior domain knowledge and classified as either low-level novice, high-level novice, or expert. Low-level novices (n = 10) were junior high school students (aged 14 - 15) who had no prior domain knowledge: as capacitors were not part of their curriculum they were unfamiliar with the relevant formulas. However, they did have modelling experience, as they had recently attended an 8-hour modelling unit in which they built system dynamics models of several phenomena (i.e., influenza, fluid dynamics, and greenhouse gasses). High-level novices (n = 10) were senior high school students (aged 18 - 20) from the science track with some prior domain knowledge (capacitors had been taught in their curriculum and all relevant formulas were addressed), and modelling experience. One year prior to the experiment they had attended the same modelling unit as the low-level novices. Additionally, they had just finished a modelling refreshment course that, among other things, involved modelling a capacitor. Experts (n = 11) were university students (aged 20 - 27) who had finished their first year in electrical engineering. They thus had extensive prior domain knowledge (their curriculum involved knowledge about capacitors well beyond the scope of the task), as well as ample modelling experience.

Materials

Participants engaged in an inquiry task in a modified standalone version of the Co-Lab learning environment (van Joolingen, de Jong, Lazonder, Savelsbergh, & Manlove, 2005). The task was to replace parts of the electrical circuit of a speed control camera so it would match new specifications. The cover story told participants that a modification to speed control cameras (adding a transmitter that activates a matrix board) caused too long recharging times of the capacitor in the electrical circuit. Participants were told that by replacing the resistor in the electrical circuit the recharging times could be influenced. They had to suggest a possible resistance value which would lead to smaller capacitor recharging times.

In order to tackle the problem, participants first had to investigate how resistance affects the time to charge a capacitor. The behaviour of a charging capacitor could be studied by running experiments with a simulation (see Figure 2.1). The simulation represented an electrical circuit containing a power source, a resistor, a device that activates a matrix board (which has resistance), and a capacitor. Experiments could be conducted with this electrical circuit to examine the influence of the resistance on the charging of the capacitor. In the simulation the

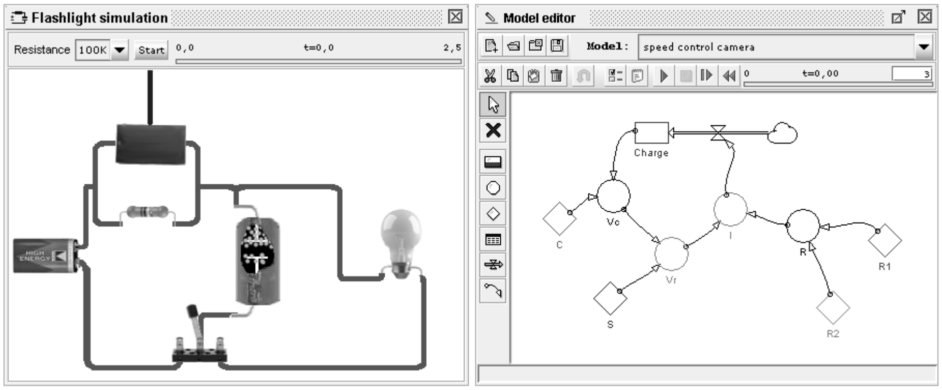

Figure 2.1. Screen capture of the simulation (left pane) and model editor tool (right pane). Pressing the start button in the simulation started an animation of moving green dots representing current, a flow of charge over time (see Equation 1). The charging of the capacitor was visualized by green dots piling up on the top plate of the capacitor. The model editor shows the reference model students had to build from their prior knowledge and/or insights gained through experimenting with the simulation.

resistor value could be manipulated (five possible values), which changed the current in the circuit. Simulation output of all variables could be inspected through a table and graph.

Participants could infer knowledge by interacting with the learning environment. Four knowledge components about electrical circuits can be distinguished: Ohms Law, Kirchhoff's law (including its two rules: the junction rule, and the loop rule), and the behaviour of capacitors. Students who are unfamiliar in the domain can generate this knowledge by conducting experiments with the simulation. For instance, from viewing the animation students can grasp the notion that a capacitor is a device where charge is stored (hence the animation was designed including a "peeled off" capacitor, so students could see a potential difference arising across the plates). Furthermore, the knowledge components could be inferred through (systematic) inspection of the results generated from these experiments (in a graph or table). For instance, students can plot the potential difference across the capacitor during charging in a graph. From inspection of this graph it can be hypothesized that as the potential difference across the capacitor increases, the charging speed decreases. Therefore, the increase in potential difference across the capacitor should be dependent (among other things) on the potential difference across the capacitor itself. Such reasoning concerns knowledge about the behaviour of capacitors and the loop rule.

The model editor (see Figure 2.1) enabled participants to build and test a model that represents their conceptions of the charging behaviour. (A reference voltage of 0 Volts at the negative battery pole was assumed so that absolute voltages could be used in the model.) The syntax of this system dynamics model makes use of 'stocks', 'auxiliaries', 'constants', 'flows' and 'relations arrows'. A model consists of several components: basic elements (i.e., elements that represent the model 'input': constants and stocks), auxiliary elements (i.e., elements that specify the integration of elements) and connecting arrows. An example looks like this: A basic element that changes over time and has an initial value (Charge) is represented in a stock. Connected to a stock are flows, indicating the changes in the stock. These changes are specified from the basic elements that remain constant (i.e., constants) (e.g., capacitance (C), power source (S), resistance (R_1 and R_2)) and auxiliary elements (i.e., auxiliaries) (e.g., potential difference across the capacitor (V_c), potential difference across the resistances (V_r), current (I), resistance total (R)) which are connected by relation arrows.

As explained in van Joolingen et al. (2005), participants could build their initial model early on by selecting pre-specified, qualitative relations from a drop-down menu (not shown in Figure 2.1). During the later stages, when participants' knowledge of the capacitor had increased, qualitative relations could gradually be replaced by quantitative ones using scientific formulas. Thus participants could use their models to express propositions about a relation between variables. Hence, students' modifications to a model were considered hypotheses that could be tested by running the model and analyzing its output through the table and graph. These tools further allowed students to compare model and simulation output in a single window.

The Co-Lab learning environment stored participants' actions in a log file; Camtasia Studio ("Camtasia Studio", 2003) was used to record participants' actions and verbalizations in real time.

Procedure

Students participated in the experiment one at a time. As experts had no prior experience with the syntax of the modelling tool, they completed a brief tutorial prior to the assignment. All other instructions and procedures were identical for the three groups of participants.

At the beginning of a session, the experimenter explained the experimental procedures. Participants were then presented with the cover story that introduced them to the inquiry task. Next, the experimenter demonstrated the procedural operation of the simulation, the model editor, and the graph and table tool. During this demonstration, the experimenter handed out a paper instruction manual on the modelling syntax participants could consult at any time during the task. All participants were familiar with this manual: both novices groups used it during their modelling unit and the experts studied the manual during their modelling tutorial prior to the assignment.

Participants were asked to think aloud during the task. Thinking aloud was practiced on a simple task (tying a bowline knot). After this final instruction, participants received the problem statement and started their inquiry. They had 1.5 hours maximum to complete the task.

During task performance the experimenter prompted the participants to think aloud when necessary. Thinking aloud was further encouraged by asking participants to state their hypotheses upon running the simulation and to verbalize their evaluation of evidence upon inspecting experimental results in the table or a graph. Towards this end the experimenter used non-directive probes to elicit the factor under investigation ("What are you going to investigate?") and its alleged effect on the output variable ("What do you think will be the outcome?") that have been shown to have no disruptive influence on participants' inquiry learning processes (Wilhelm & Beishuizen, 2004).

Coding and scoring

Variables under investigation in the study were time on task, performance success, and the three scientific reasoning processes of hypothesising, experimentation, and evidence evaluation. Time on task was assessed from the log-files. Performance success was scored from the participants' final models. Both a model content and a model structure score were calculated. The model content score represented participants' understanding of the four distinct knowledge components about electrical circuits within the task (i.e., Ohms Law: $I = V/R$, resistances connected in parallel: $1/R_t = 1/R_1 + 1/R_2$, the potential difference in the circuit depends on the power source and the potential difference across the capacitor: $\Delta V = V_s - V_c$, and the relationship between the potential difference across the capacitor and the amount of charge that gathers on the capacitor: $C = Q/V_c$). In a correct, fully specified model these components are correctly integrated and meet Equation 1. One point was awarded for each correctly specified component, leading to a four-point maximum score. Two raters scored the models of three randomly selected low-level novices, three randomly selected high-level novices and three randomly selected experts. Inter-rater reliability estimate was 1.0 (Cohen's κ).

$$(dQ/dt) = (V_s - Q/C) * (1/R_1 + 1/R_2) \qquad (1)$$ [2]

The model structure score was scored in accordance with Manlove et al.'s (2006) model coding rubric. This score represented the number of correctly specified variables and relations in the models. "Correct" was judged from the reference model shown in Figure 2.1. One point was awarded for each correctly named variable; an additional point was given if that variable was of the correct type. Concerning relations, one point was awarded for each correct link between two variables and one point was awarded for the direction. The maximum model structure score was 38. Two raters coded the models of three randomly selected low-level novices, three randomly selected high-level novices and three randomly selected experts. Inter-rater reliability estimates were .74 (variables) and .92 (relations) (Cohen's κ).

Participants' *simulation hypotheses* concerned statements about variables and relations accompanying simulation runs, and were assessed from the think-aloud protocols. Each hypothesis was classified according to the level of domain specificity using a hierarchical rubric consisting of fully-specified, partially-specified, and unspecified hypotheses (as did Lazonder et al., 2009). A fully-specified hypothesis comprised a prediction of the direction and magnitude of the effect ("I think a 10 times larger resistance will extend the capacitors' recharging period by 10"). Partially-specified hypotheses predicted the direction of effect ("I think increasing the resistance will increase the capacitors' recharging period"). Unspecified hypotheses merely denoted the existence of an effect ("I think the resistance influences the capacitors' recharging period"). Statements of ignorance or experimentation plans ("I'll just see what happens") were not considered hypotheses. Two raters coded the simulation hypotheses of three randomly selected low-level novices, three randomly selected high-level novices, and three randomly selected experts (in total 74 hypotheses). Inter-rater agreement was .77 (Cohen's κ).

In accordance with van Joolingen et al. (2005), model changes were also considered hypotheses. A *model hypothesis* was operationally defined as the changes in a

[2] Equation 1 can also be written as $dQ/dt = (V/R) \exp[-t/RC]$, with R being the total resistance of the parallel resistors. The formula used here was preferred because it is consistent with the system dynamics formalism.

participant's model between subsequent runs. Model hypotheses were coded based on the same hierarchical rubric as simulation hypotheses. Any change to a quantitatively specified relationship between two elements in the model was coded as fully-specified hypothesis. Changes in qualitative relationships were coded as partially-specified hypothesis, and changes to relation arrows not accompanied by a qualitative or quantitative specification was coded as unspecified hypothesis. Two raters coded the models of three randomly selected low-level novices, three randomly selected high-level novices and three randomly selected experts (in total 145 models). Inter-rater agreement was .85 (Cohen's κ).

The number of conducted *experiments* with the simulation and the number of model runs were retrieved from the log files. Every time participants clicked the 'Start' button in the simulation window was considered a simulation experiment. Experiments that were not accompanied by a hypothesis were considered exploratory experiments. Simulation experiments were further classified as unique or duplicated depending on whether the experiment had been previously run with the same resistance value. As the learning environment enabled participants to choose from 5 different resistance values, a maximum of 5 unique experiments could be conducted. Every time participants clicked the 'Start' button in the model editor was considered a model run. If the model had been conceptually altered since the previous run, this run was considered an experiment.

The results of participants' *evidence evaluation* was assessed from the progression of participants' models during their session. This evaluation of evidence process was coded based on participants' subsequent models. Based on cumulative evidence resulting from experimenting (and prior knowledge) participants could decide to (temporarily) accept, reject, or alter their current hypothesis (contrary to Klahr and Dunbar's (1988) study, further consideration of the current hypothesis with different experiments is conceptually not possible when a model is considered an hypothesis). Modifications to the previous version of the model were considered 'alterations', except when these modifications were deletions or additions that were not related to the previous hypothesis. Deletions of elements in prior models were considered 'rejections', as they reject the hypothesis in the prior model specified by this element. Additions of elements in models signalled 'acceptations', as the prior model was (temporarily) accepted as it was, and now a new hypothesis is considered by addition of this new element.

Results

Both groups of novices needed more than 80 minutes to complete the task (low level novices: M = 81.80, SD = 11.39; high-level novices: M = 81.30, SD = 19.61); experts took about 20 minutes less time (M = 63.36, SD = 22.12). Univariate analysis of variance (ANOVA) showed this difference to be statistically significant, $F(2,28)$ = 3.45, p = .050. Planned contrasts indicated that experts needed less time on task than novices, $t(28)$ = -18.19, p <.001, whereas the high-level novices and low-level novices needed as much time to complete the task, $t(28)$ = -.50, p = .310.

Table 2.1 presents a summary of participants' performance. Performance success was assessed from participants' final models. Multivariate analysis of variance (MANOVA) showed that the quality of the participants' models differed as function of their prior knowledge, $F(4,56)$ = 9.50, p < .001. Subsequent univariate ANOVA's indicated that prior knowledge influenced both model content, $F(2,28)$ = 59.11, p < .001, and model structure score, $F(2,28)$ = 8.28, p = .001. Planned contrasts revealed that experts achieved significantly higher model content, $t(28)$ = 3.09, p <.001, and model structure scores, $t(28)$ = 9.05, p = .001, than novices. The comparison among both groups of novices showed that high-level novices had higher model content scores than low-level novices, $t(28)$ = 1.10, p = .004. However, the model structure score indicated no significant difference between both novice groups, $t(28)$ = 3.30, p = .244.

From Table 2.1 it can be seen that participants differed in the number of hypotheses they generated. Although MANOVA with the number of simulation and model hypotheses as dependent variables did not reach significance, $F(4,56)$= 2.01, p = .105, the large standard deviations indicate a considerable variation in scores. Therefore, the content of these hypotheses was analysed using the percentages of all stated hypotheses as measure.

As few participants (4 low-level novices, 3 high-level novices, and 7 experts) stated hypotheses with both the simulation and the models, data were analysed with non-parametric Kruskal-Wallis' ranks tests. Results indicated that the groups neither differed in mean model hypothesis' specificity, $\chi^2(2, N = 20)$ = 5.59, p = .061, nor on their mean simulation hypothesis specificity, $\chi^2(2, N = 20)$ = .72, p = .699.

Table 2.1
Summary of participants' performance

	Low-level novices		High-level novices		Experts	
	M	*SD*	*M*	*SD*	*M*	*SD*
Performance success						
Model content[a]	0.00	0.00	1.10	1.20	3.64	0.67
Model structure[b]	13.30	5.74	16.60	6.40	24.00	6.40
Hypothesizing						
Simulation hypotheses	2.10	2.73	3.70	4.35	2.10	1.70
Model hypotheses	6.00	5.42	1.30	2.11	5.91	5.49
Domain specificity simulation hypotheses	1.80	0.57	1.84	0.24	1.89	0.37
Domain specificity model hypotheses	2.10	0.65	2.75	0.50	2.58	0.54
Experimenting						
Unique simulation experiments	1.80	1.87	2.50	1.90	2.45	1.21
Duplicated simulation experiments	2.60	3.69	4.90	5.92	1.91	1.64
Exploratory simulation experiments (%)	58.06	36.52	58.69	31.19	55.52	32.97
Model experiments	7.11	4.60	3.50	3.02	4.91	3.83
Exploratory model experiments (%)	10.89	22.99	7.87	12.22	0.00	0.00
Evaluating evidence						
Accepted hypotheses (%)	32.18	15.60	37.50	47.87	29.00	22.42
Reject hypotheses (%)	20.96	14.95	4.17	8.33	5.18	8.38
Altered hypotheses (%)	46.86	21.18	58.33	50.00	65.82	23.18

[a] Maximum score = 4. [b] Maximum score = 38

Figure 2.2 depicts the specificity of participants' hypotheses through time (as time on task differed between groups, it was standardized using quartiles). An increase in domain specificity was expected for both novice groups, whereas experts were expected to generate highly domain specific hypotheses throughout the task. Contrary to expectations however, the mean domain specificity of participants' hypotheses remained relatively stable through time. One noticeable finding is that low-level novices had substantially more domain specific simulation hypotheses in the fourth quartile. Yet the domain specificity of their model hypotheses failed to follow this trend.

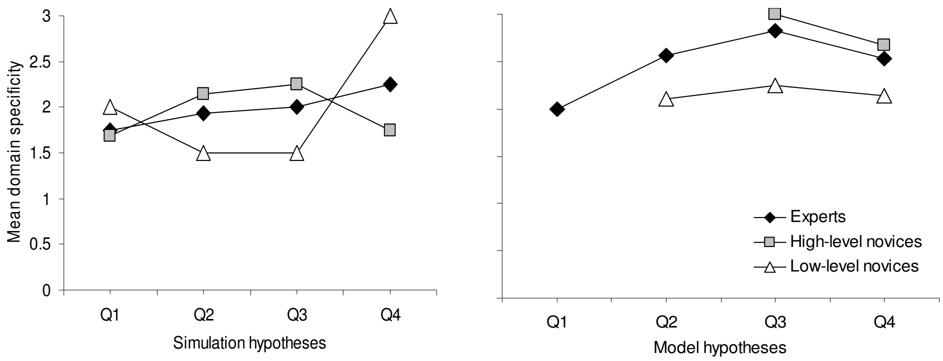

Figure 2.2. Mean specificity of participants' hypotheses accompanying simulation experiments (left pane) and model experiments (right pane) over time and by group.

Participants could experiment either by running the simulation or their models. MANOVA with the number of unique and duplicated simulation experiments as dependent variables produced no significant differences, $F(4,56) = 1.63$, $p = .179$. ANOVA of the number of model experiments was not significant either, $F(2,23) = 1.61$, $p = .218$, and nor was the percentage of these experiments that was exploratory (simulation experiments: $F(2,28) = 0.62$, $p = .545$; model experiments: $F(2,23) = 1.25$, $p = .\ 305$). These results indicate that participants with varying levels of prior knowledge performed as many experiments, and used these experiments as often to test hypotheses.

Participants could perform these experiments during the task as they deemed necessary, resulting in large inter-individual differences in experimenting behaviour over time. Figure 2.3 depicts the spread of the number of experiments conducted with the simulation and the models over time (as with hypotheses, time was divided in quartiles). As can be seen, in general the number of experiments with the simulation decreased over time, whereas the number of experiments with the models tended to increase. There was also a decline in the number of participants who experimented with the simulation. Even though an initial knowledge base could be acquired by experimenting with the simulation, seven low-level novices chose not to experiment with the simulation in the first quartile. Actually, three low-level novices did not experiment with the simulation at all. Even more participants did not make use of the modelling tool to experiment with, one low-level novice and four high-level novices never executed one of their own models.

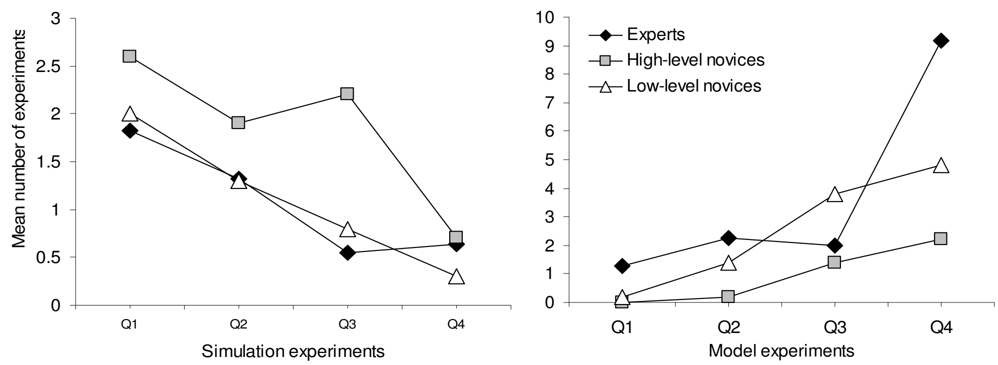

Figure 2.3. Mean number of experiments conducted with the simulation (left pane) and with the model (right pane) over time and by group.

For subsequent models, results of participants' evidence evaluating processes were analysed in light of the number of hypotheses. Therefore comparable to hypotheses' data, these data were also converted to percentages and analysed with non-parametric Kruskal-Wallis' ranks test. From Table 2.1 it can be seen that groups did not differ in percentage of evidence evaluation resulting in accepting, χ^2 (2, N = 20) = 0.10, p = .951, and alteration, χ^2 (2, N = 20) = 2.61, p = .271. However, prior knowledge affected the percentage of evidence evaluation processes resulting in rejection, χ^2 (2, N = 20) = 6.72, p = .035. Low-level novices rejected more model hypotheses than high-level novices and experts.

Qualitative analyses

From these statistical analyses it appears that novices predominantly followed the same approach as experts. Performance success scores suggest that this approach suited experts better than novices. Qualitative analyses of participants' modelling activities were performed to reveal why novices' behaviour was less effective.

When looking at participants' initial models (i.e., the first model they tried to run), it appeared that participants with domain knowledge were only a fraction better at deciding which components to include in their model. Experts' initial models contained nearly all basic elements from the target model (i.e., 1 stock and 4 constants) (M = 4.45, Range = 3-5), indicating that they could oversee the entire problem and correctly identified the relevant pieces of information from the problem statement. Novices included as many elements in their first model (low-level novices: M = 4.33, Range = 2-6; high-level novices: M = 4.00, Range = 3-5). However, low-level novices' initial models contained a few erroneous elements such as 'loading time' and 'switch' (M = 0.89, Range = 0-2), whereas high-level novices and experts' models had no such elements. The low-level novices' final models contained a comparable number of incorrect elements (M = 1.22, Range = 0-4).

Although low-level novices had a pretty good sense of which elements to include in their initial models, they were probably ignorant of the relationships between model elements. The modelling tool in Co-Lab anticipated this by offering participants the possibility to specify relationships qualitatively. Participants could thus specify relationships before they fully grasped the mathematical formula governing the relation between two variables. Surprisingly however, only two low-level novices and one expert made use of this feature. While this may seem a defendable choice for the experts and high-level novices, it may not be a wise decision for the low-level novices. Yet they generally ignored, and sometimes even deliberately rejected qualitative modelling by saying that it produced a less specific model that would not help them to discover the capacitor's behaviour.

These findings support the idea that low-level novices tried to build their models in an expert manner. But due to their lack of prior knowledge, low-level novices could only base their modelling efforts on insights gained through experimentation, or engage in trial and error activities. Therefore, participants' think-aloud protocols were analysed to reveal the reasoning behind subsequent model changes (i.e., model hypotheses). Results indicated that low-level novices hardly reasoned at all. Nine low-level novices utilized the modelling tool to experiment with their models, eight of them also experimented with adjusted models. These eight low-level novices did not motivate 87% of the changes they made to their models at all. The changes to models that were guided by reasoning could be considered 'data-driven'; this is illustrated in Excerpt 1.

Excerpt 1 (low-level novice)

"They [the resistances] ought to be 4.4 Volts.

[Participant inspects model output in the table]

Hmmz, 410 kilo Ohm, so with every kilo Ohm there will be approximately 0.1 Volts resisted. Thus this resistance resists 3 Volts and the other 1.1 Volts.

The experts, in contrast, relied heavily on their prior knowledge for their model changes. Eight experts performed more than one model experiment, and 83% of their model changes were motivated from prior knowledge; a typical example is shown in Excerpt 2. Of the remaining model changes, 12% was 'data-driven', often involving statements about previous model runs, 2% was based on logical reasoning, and 3% was not motivated.

Excerpt 2 (expert)

"Now I have the, ehm, source power I've got let's say to the…the source power is influenced by the resistances, from that I've made this current. That is the current behind the parallel resistances. As that is necessary to charge the capacitor. The formula to charge the capacitor is: the value of the capacitor times the current time derivative. So now I'm going, ehm, then you have the current over there…"

Only four high-level novices performed more than one model experiment. In the think-aloud protocols of the four high-level novices who found subsequent experimenting worthwhile, 89% percent of the changes made to the model were motivated. This reasoning was based on prior domain knowledge (28%), data from prior experiments (33%), information found in the assignment (28%; see Excerpt 3), or logical reasoning (11%).

Excerpt 3 (high-level novice)

"With these [the arrows connecting elements in the model] I want to indicate that there is a charge directly towards the capacitor…and that it goes through the sender or the resistance let's say…and then again through the capacitor, like in that circuit [the circuit depicted in the assignment paper]."

Discussion

The aim of this study was to reveal domain novices' need for support by comparing their scientific reasoning and performance success to that of students with higher levels of domain knowledge. The experts' task performance served as standard against which the scientific reasoning and knowledge acquisition of low-level novices and high-level novices were compared. The first comparison in particular elucidates the issues support for students without prior domain knowledge should address. The discussion concludes with implications for the design of such support.

Consistent with problem-solving research, the experts required less time for task completion than both groups of novices. Other findings suggest that these time differences were attributable to the experts' rich knowledge base. That is, experts needed only a few simulation experiments to create comprehensive initial models that generally contained all basic elements from the target model. Their model runs were always intended to test a hypothesis, and nearly all changes to the model were motivated from prior knowledge.

Low-level novices were predicted to perform these scientific reasoning processes in a different way. Contrary to expectations, however, their hypothesizing and experimenting did not differ from that of experts. Although the latter result is consistent with previous laboratory studies (Lazonder et al., 2008; Lazonder et al., 2009; Wilhelm & Beishuizen, 2003), the higher proportion of exploratory experiments found in these studies could not be confirmed. Together these findings suggest that low-level novices based their rather specific hypotheses on mere guesswork. The qualitative analyses bore this out: most low-level novices did not engage in qualitative modelling, and very few of the changes to their models (i.e., model hypotheses) were guided by reasoning. Therefore, many of these hypotheses inevitably were incorrect and should be rejected. This is indeed what appears to have happened since low-level novices rejected a larger proportion of their model hypotheses than experts did.

Performance success scores reflect to what extent participants' scientific reasoning was effective. Based on Klahr and Dunbar (1988), performance success was assumed to be independent of participants' prior knowledge because, contrary to most problem solving tasks, low prior knowledge participants could infer all knowledge by interacting with the learning environment. Results indicate that they did not: the quality of the experts' models was higher compared to that of the high-level novices' models, whereas high-level novices built better models than low-level novices. A closer look at these results shows that the experts achieved an almost perfect model content score; a few minor inaccuracies caused that not every expert produced a fully correct model. Low-level novices, in contrast, had rather low performance success scores. The magnitude of their model content scores indicates that they did not acquire complete understanding of any of the four formulas that governed the behaviour of the charging capacitor. Although the learning environment provided them with all necessary tools to induce this knowledge, low-level novices did not succeed in doing so –which suggests that their scientific reasoning was rather ineffective.

From these findings it can be concluded that low-level novices predominantly exhibit expert-like behaviour during an unsupported inquiry task, and that this approach apparently does not suit them that well. This conclusion is consistent with the findings of Lazonder et al. (2008). Their within-subject comparison revealed that students generally adopt a similar approach to inquiry tasks in familiar and unfamiliar domains, but perform better on tasks they possess prior knowledge of. Therefore, it can be concluded that the current results complement existing evidence on the influence of prior knowledge on inquiry behaviour. Findings from prior laboratory studies in which prior knowledge was manipulated by differences in task design, can now be generalized to more ecologically valid classroom situations.

This study added an intermediate group (i.e., high-level novices) to the novice-expert comparison. Insight into high-level novices' inquiry behaviour and difficulties is of interest for the design of support because low-level novices will probably encounter the same problems once they have gained some knowledge of the topic they are investigating. As high-level novices' prior knowledge was higher than the low-level novices' and lower than the experts', they were expected to perform better than the low-level novices, though possibly not as good as experts. Contrary to expectations, however, their hypothesizing and experimenting neither differed from that of experts, nor from that of low-level novices. The qualitative analyses suggest that this expert-like behaviour suits the high-level novices as there appeared to be sound reasoning behind the high-level novices' highly specific hypotheses. Consequently, most of their experiments resulted in either acceptation or alteration of the hypotheses, which was comparable to experts' evidence evaluation results.

The high-level novices' performance success scores were higher than low-level novices'. Yet these scores were still fairly low, considering that the high-level novices were familiar with all relevant domain knowledge. It appears that, despite their prior knowledge, performance on this task was difficult for the high-level novices, suggesting that they were unable to effectively apply their knowledge. These findings lead to the conclusion that learners who are somewhat familiar in the domain also need support in order to help them manage their knowledge to effectively perform an inquiry task.

However, there was one slightly a-typical finding. Several high-level novices were found not to perform any model experiment. This could be a result of the task difficulty. If high-level novices had difficulty expressing their knowledge in a model during the task, they probably did have enough domain knowledge to realize that the model was not good enough yet. As such, it would make sense not to run that model as they knew it to be incorrect. Future research might give more insight on this problem and how it can be overcome.

These conclusions lead to implications for support. Bearing in mind that what constitutes effective and efficient inquiry behaviour is dependent on domain knowledge, it can be argued that novices' (having no prior knowledge) unsupported inquiry behaviour was not effective on this task, but could be effective if they were familiar in the domain and would apply and expand this knowledge through iterative cycles of model testing. Conclusions for support for inquiry learning can therefore go into two directions, either providing domain support in order to increase the effectiveness of their students' natural inquiry behaviour, or process support to better attune students' inquiry behaviour to their level of domain knowledge. These two directions correspond with what Quintana et al. (2004) called content support and process support respectively.

In a literature review, de Jong and van Joolingen (1998) conclude that providing direct access to domain information seems effective as long as the information is presented concurrently with the simulation, so that the information is available at the appropriate moment. Lazonder, Hagemans, and de Jong (2010) found that offering domain support before and during the task is even more effective. Students who received domain information before and during the task not only inferred more knowledge from their investigations, but also exhibited more sophisticated scientific reasoning. This confirms the notion that providing domain knowledge to students is an effective type of support. However, as our low-level novices already exhibited quite sophisticated scientific reasoning, while still being rather unsuccessful on the task, providing domain knowledge appears not to be the most appropriate type of support. Moreover, as Lazonder et al. (2010) also mention, providing domain knowledge is somewhat at odds with the concept of inquiry learning, where learners have to discover domain knowledge themselves.

Therefore, it seems more appropriate to support students' inquiry behaviour by better attuning students' inquiry behaviour to their level of domain knowledge. Directions for such process support can be derived from this study's results. The bottleneck for novice learners was found *not* to be the identification of relevant elements, as it was the inquiry of the nature of the relationship between these elements that caused problems. Novice learners knew quite well which elements to include in the model (even their initial model contained nearly all correct elements and few erroneous elements). However, novice learners attempted to infer the relationships between those elements by means of testing hypotheses that were very specific in nature. Moreover, novices most likely based these hypotheses on guesswork, as there was hardly any underlying reasoning. As such inferring the correct relationships becomes very difficult and it is no surprise that they hardly succeeded in inferring these relationships.

The modelling tool in the learning environment aims to support learners' hypotheses construction in a graphical way (van Joolingen et al., 2005). Learners in this study were given a choice as to how detailed they wanted to specify relationships. They could opt for a self-generated, full-fletched scientific formula (i.e., quantitative relations), or select less detailed pre-specified, qualitative relations from a drop-down menu (i.e., qualitative relations). Qualitatively specified relations are more appropriate at the beginning of the modelling process when learners do not yet have a clear idea about the model they are making (Löhner, van Joolingen, & Savelsbergh, 2003; Sins, Savelsbergh, & van Joolingen, 2005). Therefore it is surprising that in the present study, where participants did not receive any kind of support, only 2 low-level novices made use of the possibility to state qualitative relations.

In view of these findings it might be fruitful to restrain domain novices' natural tendency to engage in quantitative modelling from scratch by first having them create models that are qualitatively specified, and then enabling them to transfer these qualitative relations into quantitative ones. This type of support is in line with the model progression approach described by White and Frederiksen (1990). Model progression was found to lead to higher performance (Rieber & Parmley, 1995; Swaak, van Joolingen, & de Jong, 1998). However, these authors interpret model progression as a type of support where the model at first is not offered in its full complexity, but variables are gradually introduced (or, in terms of White and Frederiksen (1990), a model progression where the degree of elaboration of a model is increased). Our proposed support, as suggested by Gobert and Clement (1999), can be considered a more fine-grained kind of model progression, where the specificity of the models is increased. This kind of model progression resembles what White and Frederiksen (1990) call model progression where the *order* of a model is increased.

To conclude, we propose to support learners on an inquiry learning task with model progression, where the model is progressed in specificity. In line with the coding of the model hypotheses, three increasingly specific stages of modelling can be identified: a stage in which relationships between elements are unspecified, a stage in which relationships between elements are specified qualitatively, and a stage in which these relationships are specified quantitatively. In the first stage of model progression, students investigate a phenomenon (e.g., an electrical circuit containing a capacitor) and have to make a model structure of that phenomenon without having to specify the relationships in the model. In the second stage, students continue to investigate the phenomenon in order to specify the relationships in their model qualitatively. In the third stage, students finalize their investigation of the phenomenon by replacing the qualitatively specified relationships with quantitatively specified relationships.

One important condition for this form of model progression to be effective is that students should have enough opportunity to build and test hypotheses. The simulation that was used in the present study might not satisfy this requirement: although output of all elements in the simulation interface could be inspected in the table or graph, only one element (the resistor) could be manipulated. Allowing students to change the values of the other elements as well extends the possibilities for students to validate the hypotheses they generate from interacting with the simulation and running their own model. Model progression could then be an effective way to support students' inquiry and modelling process. Validating this assumption in science classrooms is an important topic for future research.

References

Batra, D., & Davis, J. G. (1992). Conceptual data modeling in database design: Similarities and differences between expert and novice designers. *International Journal of Man-Machine Studies, 37*, 83-101. doi: 10.1016/0020-7373(92)90092-Y

Bransford, J. D., Brown, A. L., & Cocking, R. R. (Eds.). (2002). *How people learn: Brain, mind, experience, and school*. Washington, DC: National Academy Press.

Camtasia Studio (Version 2.0). (2003). TechSmith.

Chi, M. T. H., Glaser, R., & Farr, M. (1988). *The nature of expertise*. New Jersey: Lawrence Erlbaum Associates.

Chinn, C. A., & Malhotra, B. A. (2002). Epistemologically authentic inquiry in schools: A theoretical framework for evaluating inquiry tasks. *Science Education, 86*, 175-218. doi: 10.1002/sce.10001

Davis, E. A., & Linn, M. C. (2000). Scaffolding students' knowledge integration:Prompts for reflection in KIE. *International Journal of Science Education, 22*, 819-837. doi: 10.1080/095006900412293

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*, 179-201. doi: 10.3102/00346543068002179

Gobert, J. D., & Clement, J. J. (1999). Effects of student-generated diagrams versus student-generated summaries on conceptual understanding of causal and dynamic knowledge in plate tectonics. *Journal of Research in Science Teaching, 36*, 39-53. doi: 10.1002/(SICI)1098-2736(199901)36:1<39::AID-TEA4>3.0.CO;2-I

Gobert, J. D., & Pallant, A. (2004). Fostering students' epistemologies of models via authentic model-based tasks. *Journal of Science Education and Technology, 13*, 7-22. doi: 10.1023/B:JOST.0000019635.70068.6F

Jackson, S. L., Stratford, S. J., Krajcik, J., & Soloway, E. (1994). Making dynamic modeling accessible to precollege science students. *Interactive Learning Environments, 4*, 233 - 257.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12*, 1-48. doi: 10.1207/s15516709cog1201_1

Lazonder, A. W., Hagemans, M. G., & de Jong, T. (2010). Offering and discovering domain information in simulation-based inquiry learning. *Learning and Instruction*, doi:10.1016/j.learninstruc.2009.08.001.

Lazonder, A. W., Wilhelm, P., & Hagemans, M. G. (2008). The influence of domain knowledge on strategy use during simulation-based inquiry learning. *Learning and Instruction, 18*, 580-592. doi: 10.1016/j.learninstruc.2007.12.001

Lazonder, A. W., Wilhelm, P., & van Lieburg, E. (2009). Unraveling the influence of domain knowledge during simulation-based inquiry learning. *Instructional Science*, doi: 10.1007/s11251-008-9055-8. doi: 10.1007/s11251-008-9055-8

Leutner, D. (1993). Guided discovery learning with computer-based simulation games: Effects of adaptive and non-adaptive instructional support. *Learning and Instruction, 3*, 113-132. doi: 10.1016/0959-4752(93)90011-N

Löhner, S., van Joolingen, W. R., & Savelsbergh, E. R. (2003). The effect of external representation on constructing computer models of complex phenomena. *Instructional Science, 31*, 395-418. doi: 10.1023/A:1025746813683

Manlove, S., Lazonder, A. W., & de Jong, T. (2006). Regulative support for collaborative scientific inquiry learning. *Journal of Computer Assisted Learning, 22*, 87-98. doi: 10.1111/j.1365-2729.2006.00162.x

Quintana, C., Reiser, B. J., Davis, E. A., Krajcik, J., Fretz, E., Duncan, R. G., Kyza, E., Edelson, D., & Soloway, E. (2004). A scaffolding design framework for software to support science inquiry. *The Journal of the Learning Sciences, 13*, 337-386. doi: 10.1207/s15327809jls1303_4

Rieber, L. P., & Parmley, M. W. (1995). To teach or not to teach? Comparing the use of computer-based simulations in deductive versus inductive approaches to learning with adults in science. *Journal of Educational Computing Research, 13*, 359-374. doi: 10.2190/M8VX-68BC-1TU2-B6DV

Schauble, L., Klopfer, L., & Raghavan, K. (1991). Students' transitions from an engineering to a science model of experimentation. *Journal of Research in Science Teaching, 28*, 859-882. doi: 10.1002/tea.3660280910

Shrager, J., & Klahr, D. (1986). Instructionless learning about a complex device: the paradigm and observations. *International Journal of Man-Machine Studies, 25*, 153-189. doi: 10.1016/S0020-7373(86)80075-X

Shute, V. J., Glaser, R., & Raghavan, K. (1989). Inference and discovery in an exploratory laboratory. In P. L. Ackerman, R. J. Sternberg & R. Glaser (Eds.), *Learning and individual differences* (pp. 279-326). New York: Freeman.

Sins, P. H. M., Savelsbergh, E. R., & van Joolingen, W. R. (2005). The difficult process of scientific modeling: An analysis of novices' reasoning during computer-based modeling. *International Journal of Science Education, 14*, 1695-1721. doi: 10.1080/09500690500206408

Swaak, J., van Joolingen, W. R., & de Jong, T. (1998). Supporting simulation-based learning; the effects of model progression and assignments on definitional and intuitive knowledge. *Learning and Instruction, 8*, 235-252. doi: 10.1016/s0959-4752(98)00018-8 van Joolingen, W. R., de Jong, T., Lazonder, A. W., Savelsbergh, E. R., & Manlove, S. (2005). Co-Lab: Research and development of an online learning environment for collaborative scientific discovery learning. *Computers in Human Behavior, 21*, 671-688. doi: 10.1016/j.chb.2004.10.039

White, B. Y., & Frederiksen, J. R. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 42*, 99-157. doi: 10.1016/0004-3702(90)90095-h

White, B. Y., Shimoda, T. A., & Frederiksen, J. R. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. *International Journal of Artificial Intelligence in Education, 10*, 151-182.

Wilhelm, P., & Beishuizen, J. (2003). Content effects in self-directed inductive learning. *Learning and Instruction, 13*, 381-402. doi: 10.1016/S0959-4752(02)00013-0

Wilhelm, P., & Beishuizen, J. J. (2004). Asking questions during self-directed inductive learning: Effects on learning outcome and learning processes. *Interactive Learning Environments, 12*, 251 - 264. doi: 10.1080/10494820512331383449

Zimmerman, C. (2000). The development of scientific reasoning skills. *Developmental Review, 20*, 99-149. doi: 10.1006/drev.1999.0497

Zimmerman, C. (2007). The development of scientific thinking skills in elementary and middle school. *Developmental Review, 27*, 172-223. doi: 10.1016/j.dr.2006.12.001

Chapter 3

Comparing two types of model progression in an inquiry learning environment with modelling facilities[1]

Abstract

The educational advantages of inquiry learning environments that incorporate modelling facilities are often challenged by students' poor inquiry skills. This study examined two types of model progression as means to compensate for these skill deficiencies. Model order progression (MOP), the predicted optimal variant, gradually increases the specificity of the relations between variables, whereas model elaboration progression (MEP) gradually expands the number of variables in the task. The study utilized a between-subject design with three conditions: a MOP condition (n = 28), a MEP condition (n = 26), and a control condition without model progression (n = 30). Consistent with expectations, model progression enhanced students' task performance; a comparison among the two model progression conditions confirmed the predicted superiority of the MOP condition. These results are discussed in relation to the inconsistent findings from prior research, and ways to optimize model order progression are advanced.

[1]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. doi: 10.1016/j.learninstruc.2011.01.003 (with modifications).

Introduction

Computer-supported inquiry learning environments essentially enable students to learn science by doing science, offering resources to develop a deep understanding of a domain by engaging in scientific reasoning processes such as hypothesis generation, experimentation, and evaluating evidence. Computer simulations have long been incorporated in these environments, and are today increasingly being supplemented with opportunities for students to build computer models of the phenomena they are investigating via the simulation. As in authentic scientific inquiry, modelling is considered an integral part of the inquiry learning process as students can build models to express their understanding of the relation between variables (van Joolingen, de Jong, Lazonder, Savelsbergh, & Manlove, 2005; White, Shimoda, & Frederiksen, 1999). Students can check their understanding by running the model; evaluating evidence then occurs by weighting model output against prior knowledge or the data from the simulation. These comparisons yield further insight into the phenomenon and assist students in generating new hypotheses.

The educational advantages of inquiry learning are often challenged by the students' poor inquiry skills. De Jong and van Joolingen's (1998) review showed that many students experience difficulties during simulation-based inquiry learning. For example, students are unable to infer hypotheses from data, design inconclusive experiments, show inefficient experimentation behaviour, and ignore incompatible data. Students also experience difficulties during modelling. Hogan and Thomas (2001) noticed that students often fail to engage in dynamic iterations between examining output and revising models, and merely use output at the end of a session to check if the model's behaviour matches their expectations. A related problem concerns the students' lack of persistence in debugging their model to fine-tune its behaviour (Stratford, Krajcik, & Soloway, 1998).

These findings suggest that students' difficulties with inquiry and modelling both lie at a conceptual level. When provided with a simulation, most students manage to design and execute experiments; inferring knowledge from these experiments appears to be the major source of difficulty. Likewise, students are capable of building syntactically correct models, but often fail to relate knowledge about phenomena to those models (Sins, Savelsbergh, & van Joolingen, 2005). As this ineffective behaviour is a serious obstacle to learning, additional support is needed in order for inquiry learning and modelling to be effective.

Mulder, Lazonder, and de Jong (2010) were among the first to identify guidelines for supporting learners during inquiry learning with modelling. Their study compared domain novices' unsupported inquiry behaviour and performance to that of a considerably more knowledgeable reference group (referred to as 'experts'). Using Klahr and Dunbar's (1988) SDDS model as framework, the analyses focused on the processes of hypothesis generation, experimentation, and evaluating evidence. Results indicated that novices and experts were quite comparable with regard to these processes, suggesting that novices predominantly exhibited expert-like behaviour. However, as the novices received no support whatsoever, they induced virtually no knowledge from their inquiry and modelling activities (as also indicated by Dean & Kuhn, 2007; Klahr & Nigam, 2004; Mayer, 2004). Subsequent qualitative analyses provided starting points for the design of learner support. Contrary to expectations, novice learners knew quite well which elements to include in their models, and even their initial models contained nearly all correct elements. Generating the relationships between those elements appeared to be considerably more problematic. Novices generated and tested hypotheses about relations that were so specific that it is highly unlikely that these hypotheses originated from their inquiry or modelling activities, suggesting that they were merely based on guesswork. From these findings, Mulder et al. (2010) concluded that learner support should assist students in identifying the relations between the elements in their models.

Some learning environments such as Co-Lab (van Joolingen et al., 2005), LinkIt (Ogborn, 1998) and Model-it (Krajcik et al. Blumenfeld, Markx, Bass, Fredericks, and Soloway, 2000) already offer this support in a rather unobtrusive way by giving learners a choice as to how detailed they want to specify relationships. Learners can opt for a self-generated, full-fledged scientific formula (i.e., quantitative relations), or select less detailed pre-specified, qualitative relations from a drop-down menu (i.e., qualitative relations). According to Löhner, van Joolingen, and Savelsbergh (2003) such qualitatively specified models are more appropriate at the beginning of the modelling process when learners do not yet have a clear idea about the model they are making. It is therefore surprising that participants in the Mulder et al. (2010) study, who received no support, hardly used the possibility to state qualitative relations.

In view of these findings, it might be more fruitful to restrain domain novices' natural tendency to engage in quantitative modelling from scratch by first having them create models that are qualitatively specified, and then enabling them to transfer these qualitative relations into quantitative ones. This form of model progression is generally assumed to be beneficial to novice learners (de Jong, 2005; White & Frederiksen, 1990). The goal of the present study was to empirically assess the effectiveness of this type of scaffolding.

Theoretical and empirical background

The idea of model progression was coined by White and Frederiksen (1990) who used it to create problem sets that motivate successive refinements to the students' mental models. They distinguished three dimensions on which models may vary: their perspective, their degree of elaboration, and their order. Lateral progressions that represent alternative means of understanding the domain involve changes in model *perspective*. In the domain of electrical circuits, for instance, models describing Kirchhoff's Voltage Law use a different perspective from models describing Coulomb's Law. Upward progressions to more sophisticated models involve changes in a model's degree of elaboration and order. The *degree of elaboration* is determined by the number of variables and relations in a model. The core idea of model elaboration progression is therefore to let students start off with a simplified version of the phenomena; additional variables (and their relations) are introduced step by step over the course of the session so as to expose the students gradually to the full complex model. The *order* of a model concerns what type of reasoning it supports (i.e., qualitative reasoning or quantitative reasoning). White and Frederiksen postulated that a qualitative understanding needs to be developed before a quantitative understanding should occur. A further distinction is made in the qualitative understanding; the focus should initially be on the students' reasoning about the presence or absence of elements from the phenomena under investigation, and subsequently change to reasoning on the basis of incremental changes of these elements.

Model order progression resembles the type of scaffolding that was advocated on the basis of the Mulder et al. (2010) study. Model elaboration progression represents a viable alternative to provide learners with increasingly sophisticated models about a domain, and a comparison among these two forms of model progression could validate the alleged benefits of the former. Model perspective progression was not included in this comparison because there is no inherent increasing complexity associated with offering different perspectives of the same phenomena. As model perspective progression is not relevant to the purpose of the present study, the remaining part of this section discusses research on the simple-to-complex organization of learning materials.

Gradually introducing learners to increasingly more sophisticated or comprehensive subject matter has long since been recognized as powerful instructional strategy (e.g., Gagné, 1977; Reigeluth & Stein, 1983). Early attempts to reducing complexity during initial learning with computers can be found in the field of software training. Carroll and Carrithers (1984) provided novice learners with a so-called training-wheels system for learning to use a word processor. The key characteristic of this system is that features of the word processor new users typically do not need, but which can be springboards for errors and confusions, were disabled. Carroll and Carrithers reasoned that in a reduced interface learners are prevented from getting caught up in tangles of error and confusion, and as such will spend less time on errors. In their first experiment, the participants were ‘learning by doing’, they were given an example letter which they had to reproduce with the word processor. Participants who were provided with the training-wheels version of the word processor performed faster and more successful overall than participants who worked with the complete version of the program.

Research on combining a training-wheels system with other support has produced mixed results. Carroll and Carrithers (1984; experiment 2) demonstrated that training-wheels have added value to an instructional manual: learners working with an instructional manual in a simplified version of the domain performed faster and more successful compared to learners working with a manual in the full-complex domain. Results further showed that the training-wheels reduced learners’ time spent recovering from errors, which most likely accounts for the instructional efficacy of the training-wheels system (cf. Lazonder & van der Meij, 1995). However, Spannagel Girwids, Löthe, Zendler, and Schroeder (2008) found that learning to use a spreadsheet program with animated instructions, predominantly led to better performance than learning with text manuals, and that a training-wheels interface did not yield better results for students who learned with animations.

In a more recent study, Löhner et al. (2003) replicated the training-wheels findings in the domain of modelling the temperature inside a house. They compared the performance of a textual modelling group and a graphical modelling group. Participants in the textual modelling group had to build a full-complex model by specifying the relations between variables in precise, quantitative form. Participants in the graphical modelling group, in contrast, only had to indicate whether relations were positive or negative (i.e., specifying the relations qualitatively); the underlying mathematical specifications of the relations were handled by the system. Students using the graphical representation were found to switch quickly from one relation to the next, and try every idea that came up, which might be a viable strategy for the initial stages of a modelling process. Löhner et al. therefore concluded that at the beginning of an inquiry process, novice learners benefit from building qualitatively specified models compared to building quantitative models.

Although these studies demonstrate that starting off in a simple form can be beneficial to learning, they did not take the progression to higher levels of complexity into consideration. Supportive evidence on this matter can be found in the literature on learning with simulations. In a study by Alessi (1995), model progression pertained to the fidelity of the simulation; the general idea behind this form of progression was to go from 'simplified' to more 'realistic' simulations. Alessi assumed that fidelity progression would enhance learning because a simplified simulation supposedly facilitates initial learning whereas high fidelity is expected to be better from a transfer point of view. To validate this claim, three groups were compared that learned procedural knowledge about how to use a multimeter either with a low-fidelity simulation, fidelity progression simulations, or a high-fidelity simulation. The results confirmed some of Alessi's expectations: even though the three groups did not differ on measures of learning while working with the simulations, the high fidelity and progression simulations were found to enhance performance on a transfer task, thus supporting the notion that high fidelity simulations are superior in transfer. More recently, Zacharia and Olympiou (2011) were unable to replicate these findings. Participants who progressed from experimenting with a simulation to experimenting with the real equipment were found to learn as much as those who experimented with the real equipment only. However, participants who only worked with the simulation performed as well as participants in the other two conditions.

Other studies investigated model elaboration progression. Rieber and Parmley (1995) compared performance of students who were presented with either a structured or an unstructured simulation regarding the physics principles of Newtonian mechanics. The structured simulation consisted of a series of four activities in which students were given increasing levels of control over a simulated, free-floating object. This simulation was considered 'structured' because each activity included a controlled number of new subskills, and each successive activity incorporated the subskills of the preceding activity. The unstructured simulation consisted of an open-ended and unstructured activity in which subjects assumed full control over the floating object from the very beginning. Results indicated that students in the structured condition outperformed students in the unstructured condition.

Swaak, van Joolingen, and de Jong (1998) replicated these findings in the domain of oscillatory motion. The type of motion in their study depended on the presence of friction and/or an external force. In case both are absent, the motion is free; if only friction is present, the oscillation is damped; and if both are present, there is forced oscillatory motion. Model progression pertained to the degree of elaboration in that learners were first given a simulation about free oscillatory motion, then a simulation of damped motion, and finally a simulation on forced oscillatory motion. The study's main finding was that students in the model progression condition developed more intuitive knowledge about oscillatory motion than students from a control group who received no model progression.

Results further indicated that adding assignments to the model progression has no significant facilitating or deteriorating effect.

Quinn and Alessi (1994) were unable to replicate the facilitative effects of model progression. They compared groups that differed in whether the simulation was presented in its most complex form initially or whether it was presented in sections of increasing complexity. The computer simulation was a model of the spread of an influenza epidemic in which the number of people ill with influenza depended on four variables: the number of contacts per person per week, the time to illness, the duration of illness, and the length of the immune period. One group worked with a simulation in which all four variables were present, whereas an other (model elaboration progression) group worked on a simulation in which the variables were introduced gradually. Students in the latter group initially performed better than students who worked with the full-complex simulation, but this effect faded out upon completion of the task.

De Jong et al. (1999) combined model perspective and elaboration progression in a simulation on collisions. They divided the domain into five progression levels. The model's degree of elaboration was progressed in the first three levels; the last two levels offered two alternative perspectives on collisions. Contrary to Swaak et al. (1998), this study showed no main effect of model progression, which was allegedly due to the level of task complexity. Collisions is a relatively straightforward domain that might not be complicated enough for the effects of model progression to show. This explanation was substantiated by the fact that participants in this study had considerable prior knowledge, and therefore might not have needed the first three level of model progression.

To conclude, model progression has been investigated in slightly different configurations, task domains, and for different types of knowledge. These cross-study variations could be the reason why results on the effectiveness of model elaboration progression are inconclusive. Another, perhaps more plausible explanation is that model elaboration progression is a suboptimal way to arrange learning tasks in a simple-to-complex sequence. Various authors have postulated that model *order* progression better meets the learning needs of domain novices (Löhner et al., 2003, Mulder et al., 2010, Veenman & Elshout, 1995, White & Frederikisen, 1990), but the effectiveness of this type of scaffolding has neither been assessed nor compared to that of model elaboration progression. Both issues were central to the research reported below.

Research design

This study aimed to scaffold domain novices on an inquiry learning task with model progression in which the model order is progressed. The study utilized a between-subject design with three conditions. Students in the *model order progression (MOP)* condition had to build increasingly more specific models. They received a full simulation with four variables, and had to model its behaviour in three consecutive phases. Modelling thus progressed from indicating the presence of all elements and relations in the model, through a qualitative specification of these relations, to a quantitative specification. Performance in this condition was compared to two reference groups. In one group, the *model elaboration progression (MEP)* condition, students had to build increasingly more comprehensive models. Their version of the simulation contained four variables that were introduced one at a time. Students' task was to build the model underlying each simulation quantitatively from scratch. The second reference group was not scaffolded by model progression. Students in this *control* condition received the full-complex simulation and had to infer and build the quantitative model that governed its behaviour from scratch and without any externally-imposed structuring.

Comparing performance success of both model progression conditions to the control condition will demonstrate the instructional efficacy of this type of scaffolding. A comparison among both model progression groups served to validate the predicted superiority of MOP (Hypothesis 1). As MOP was intended to scaffold learners' relation construction –a key problem to domain novices– MOP students were expected to outperform MEP students on the construction of relations in their models (Hypothesis 2).

Method

Participants and design

The initial sample consisted of 90 Dutch high school students from the science track, aged 15-17. However, as 6 students were absent due to illness during one of the sessions, analyses were performed with 84 participants. A review of school curricula and teacher statements showed that the charging of capacitors, which was the topic of inquiry, had not been taught yet in the students' physics classes. A pretest was administered to substantiate that participants were indeed domain novices; class-ranked pretest scores were used to assign students to either the MOP condition (n = 28), the MEP condition (n = 26), or the control condition (n = 30).

Materials

Inquiry task and learning environments

All participants worked on an inquiry task about the charging of a capacitor. Their assignment was to examine an electrical circuit in which a capacitor was embedded, and create a computer model that mirrors the capacitor's charging behaviour. Participants performed this task within a modified stand-alone version of the Co-Lab learning environment (van Joolingen et al., 2005) that stored all participants' actions in a logfile.

The learning environment housed a *simulation* of an electrical circuit containing a voltage source, two light bulbs, and a capacitor. Through systematic experimentation with this simulation, participants could induce four physics equations: (1) Ohms law, (2) the junction rule of Kirchoff's law, (3) the loop rule of Kirchoff' law, and (4) the behaviour of capacitors.

The learning environment also contained a *model editor* tool that enabled participants to represent their knowledge of the four physics equations in an executable computer model. As can be seen from Figure 3.1, such models have a graphical structure that consists of variables and relations. Variables are the constituent elements of a model and can be of three different types: variables that do not change over time (i.e., constants), variables that specify the integration of other variables (i.e., auxiliaries), and variables that accumulate over time (i.e., stocks). Relations define how two or more variables interact. Each relation is visualized by an arrow connector to indicate the causal link between model elements, and specified by a quantitative formula to indicate the exact nature of this relationship. The model editor also enabled participants to test their understanding by running the model and analysing its output through the *table* and *graph* tool. These tools further allowed students to compare model and simulation output in a single window. Students could use the results of this comparison to adjust or fine-tune their model and thus build an increasingly elaborate understanding of a charging capacitor.

An embedded *help file* tool contained the assignment and offered explanations of the operation of the tools in the learning environment. The help files also informed participants about the specifics of their condition by indicating whether and how their session was divided into phases. The help files contained no domain information on electrical circuits and capacitors as this knowledge should be inferred from interacting with the simulation.

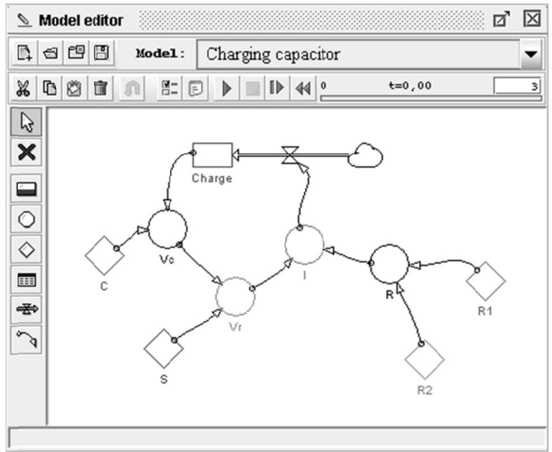

Figure 3.1. Screen capture of the model editor tool, which shows the reference model students had to build from their prior knowledge and/or insights gained through experimenting with the simulation. The reference model depicts the charging of a capacitor. The stock element (Charge), which changes over time, is influenced by four constant elements (capacitance (C), power source (S), and two resistances (R1 and R2)) and four mediating auxiliary elements (potential difference across the capacitor (Vc), potential difference across the resistances (Vr), current (I), and resistance total (Rt)).

Variants of the learning environment for the different conditions

All conditions used the same instructional content (i.e., electrical circuits), but differed with regard to the scaffolding mechanisms (see Figure 3.2). Participants in the *control* condition worked with the standard configuration of the environment (as described above) and thus received no scaffolding.

Participants in the *model order progression* (MOP) condition received a full-complex version of the simulation, and were asked to induce and build increasingly specific models. Specificity pertained to the relations in the model and progressed in three phases from identifying a relation to quantitatively specifying that relation (cf. Lazonder, Wilhelm, & van Lieburg, 2009; Mulder et al., 2010). In Phase 1, students just had to indicate the model elements (variables) and which ones affected which others (relationships) – but not *how* they affected them. In Phase 2, students had to provide a qualitative specification of each relationship so as to indicate the general direction of effect (e.g., if resistance increases, then current decreases). In Phase 3, students had to specify each relationship quantitatively in the form of an actual equation (e.g., I = V / R).

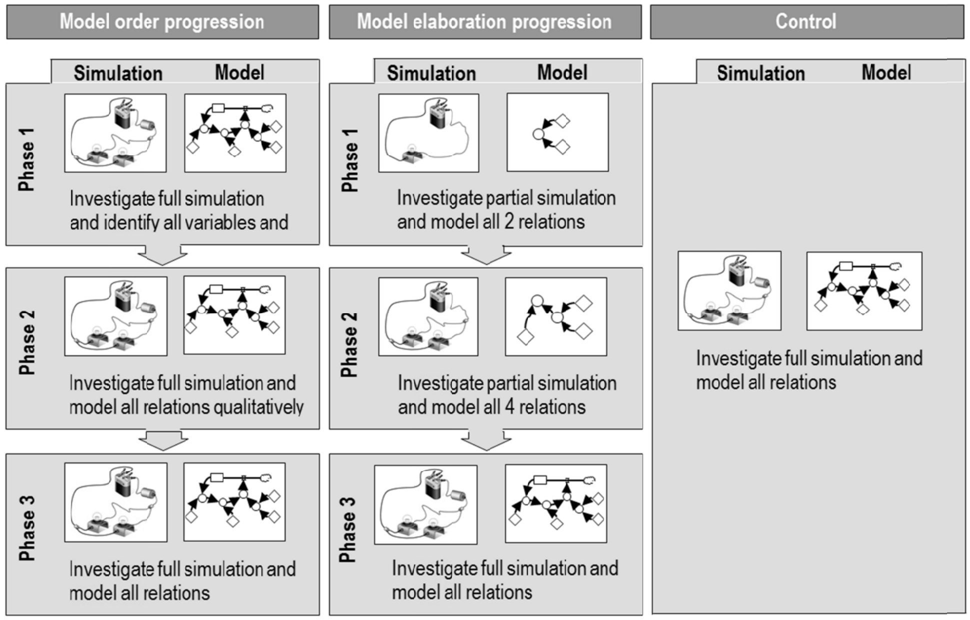

Figure 3.2. Schematic overview of the three experimental conditions. Students in the model order progression (MOP) condition investigated a full-complex simulation in all three phases and had to induce and build increasingly specific models. Students in the model elaboration progression (MEP) condition were given an increasingly elaborate simulation in each phase which they had to model quantitatively. Students in the control condition worked with a full-complex simulation and had to induce and build a full quantitative model.

As the first MOP phase had students identify relationships without specifying them, it was technically impossible for the model editor to execute these models. Model runs in this phase therefore activated a software agent that assessed the correctness of the students' model by comparing its elements and relations to a reference model. Results were presented in a bar chart tool with stacked columns that showed the number of correct, incorrect, and incomplete elements as well as the number of correct and incorrect relations. The bar chart tool was available in Phase 1 only. However, students could still add and delete elements and relations during subsequent phases.

In the *model elaboration progression* (MEP) condition, the complexity of the simulation was gradually increased by adding components to the electrical circuit. The simulation in Phase 1 contained a circuit with a voltage source and one light bulb, enabling discovery of Ohm's law. A second light bulb was added to the electrical circuit in the simulation in Phase 2, now introducing the junction rule of Kirchoff's law. The capacitor was added to the simulation in Phase 3, introducing both the loop rule of Kirchoff's law and the behaviour of capacitors. Participants had to induce and build a quantitative model of the circuit in each simulation. Over phases, participants could extend their model to incorporate the new elements. Both the bar chart tool and the possibility to engage in qualitative modelling were disabled in this condition.

Pretest

A pretest consisting of eight open-ended questions assessed participants' prior knowledge of electrical circuits. Four questions addressed the meaning of key domain concepts (i.e., voltage source, resistance, capacitor, capacitance), the other four items addressed the knowledge about the charging of a capacitor in an electrical circuit (i.e., Ohms law, Kirchoff's law (including its two rules: the junction rule and the loop rule), and the behaviour of capacitors). As performance on the test was expected to be low, three simple filler items on the interpretation of numerical data were added to sustain students' motivation during the test. These filler items were left out of the analysis. A rubric was developed to score participants' answers to the eight questions, and one point was allocated to each correct response. Two raters used this rubric to score a randomly selected set of 24 pretests; inter-rater reliability was .89 (Cohen's κ).

Procedure

All participants engaged in two sessions: a 50-min introduction and a 100-min experimental session. The time between sessions was one week maximum. To control for differences of the duration of this break, the allocation to condition occurred within each class, so that participants with different inter-session time gaps were equally spread across conditions. During the introductory session, participants first filled out the pretest, then received a guided tour of the Co-Lab learning environment, and finally completed a brief tutorial that familiarized them with the system dynamics modelling language and the operation of the modelling tool.

The experimental session started with a brief reminder that some participants would work in a learning environment where the assignment was split into phases (i.e., the model progression conditions), whereas others would receive a non-divided assignment (i.e., the control condition). Participants were instructed to open the help file tool upon entering the learning environment to find out the specifics of the condition they were assigned to. Consequently, participants were only aware of the details of their own condition. Participants in both model progression conditions were further told that they were free to progress through the phases, but could not return to a previous phase. Toward this end, phase changes were password protected, and participants had to ask the experimenter to unlock the next level. The experimenter did so only if a participant was certain about the phase change and had saved his/her model. After these instructions participants started the assignment. They worked individually and could ask the experimenter for technical assistance only. Participants could stop ahead of time if they had completed the assignment.

Coding and scoring

All data were assessed from the logfiles. Variables under investigation were time on task and performance success. Time on task concerned the duration of the experimental session.

Performance success scores were assessed from participants' final models. For both model progression conditions, intermediate performance success scores were assessed at the end of each phase. A model structure score was calculated in accordance with Manlove, Lazonder, and de Jong's (2006) model coding rubric. This score represented the number of correct variables and relations in the models. 'Correct' was judged from the reference model shown in Figure 3.1. One point was awarded for each correctly named element; an additional point was given if that variable was of the correct type (i.e., constant, auxiliary, or stock). Concerning relations, one point was awarded for each correct link between two variables and one point was awarded for the direction. The maximum model structure score was 38. A previous study (Mulder et al., 2010) found inter-rater reliability estimates of .74 (variables) and .92 (relations) (Cohen's κ).

As the model structure score leaves the quantitative aspects of the model unaddressed, a complementary final model content score was calculated. This score represented participants' understanding of the physics equations that gouvern the behaviour of a charging capacitor (i.e., Ohms Law: $I = V / R$; resistances connected in parallel: $1 / R_t = 1 / R_1 + 1 / R_2$; the potential difference in the circuit depends on the power source and the potential difference across the capacitor: $\Delta V = V_s - V_c$; and the relationship between the potential difference across the capacitor and the amount of charge that gathers on the capacitor: $C = Q / V_c$). In a correct, fully-specified model these components are correctly integrated as represented in Equation 1:

$$(dQ / dt) = (V_s - Q / C) * (1 / R_1 + 1 / R_2) \qquad (1)$$

One point was awarded for each correctly specified part, leading to a four-point maximum score. A prior study (Mulder et al., 2010) found the inter-rater reliability to be 1.0 (Cohen's κ).

Results

Preliminary analyses were performed to check whether the matching of participants had lead to comparable levels of prior knowledge across conditions and to validate the selection of participants. The entire-sample mean pretest score was 1.35 (SD = 1.10), which was deemed sufficiently low to assume that participants can be considered domain novices. The mean pretest scores for each condition are presented in Table 3.1. Univariate analysis of variance (ANOVA) showed that there were no significant differences in prior knowledge among the three experimental conditions, $F(2, 81) = 0.24$, $p = .787$. The time on task scores from Table 3.1 further show that, on average, participants in each condition spent over 90 min working on the assignment. Univariate ANOVA showed that the minor cross-condition differences in time were not statistically significant, $F(2, 81) = 1.34$, $p = .268$.

Performance success was assessed from the participants' final models (see Table 3.1). A distinction was made between model content and model structure scores. As the model content scores failed to meet the normality assumption, this data was analysed by a non-parametric Kruskal-Wallis test. Results showed a significant effect for experimental condition on the model content scores of participants' final models, $H(2) = 13.16$, $p = .001$. Post hoc comparisons, using Mann-Whitney U tests with Bonferroni correction ($\alpha = .0167$) revealed no differences in model content scores between the MOP condition and either the MEP condition, $U = 114$, or the control condition, $U = 174$. Comparison among the latter two conditions revealed a

Table 3.1
Summary of participants' performance

	MOP (n=28)		MEP (n=26)		Control (n=30)	
	M	SD	M	SD	M	SD
Pretest score	1.39	1.20	1.42	1.18	1.23	1.01
Time on task (min)	98.41	5.82	93.60	13.82	95.58	11.66
Performance success						
Model content score [a,b]	0.00	0.00	0.50	0.91	0.07	0.37
Model structure score (variables) [c]	7.29	2.57	6.58	2.50	6.63	1.63
Model structure score (relations) [d]	6.86	3.79	3.42	3.60	3.17	3.24

[a]Maximum score = 4. [b]As only 12 MOP participants progressed through all phases, a model content score of 0 can often be explained by slow progressing through phases. [c]Maximum score = 18. [d]Maximum score = 20.

significant difference in favour of the MEP condition, $U = 298$, $r = .33$. In interpreting these results, it should be noted that few MOP participants ($n = 12$) reached the third phase where they could specify their model quantitatively. The remaining 16 MOP participants obtained a model content score of zero, which, as will be discussed below, can often be explained by slow progressing through phases.

Participants' model structure scores were analyzed by MANOVA with both model structure aspects (i.e., variables and relations) as dependent variables. Using Pillai's trace, this analysis produced a significant effect for experimental condition, $V = 0.21$, $F(4, 162) = 4.74$, $p = .001$. Subsequent univariate ANOVAs validated the conjecture that model progression has no effect on the number of correct variables in the students' model, $F(2, 81) = 0.85$, $p = .431$, but does enhance the quality of the relations between these variables, $F(2, 81) = 9.53$, $p < .001$. Helmert planned contrasts revealed that the model progression conditions combined had significantly higher scores for relations than the control condition, $t(81) = 2.45$, $p = .006$, $r = .26$, and that the MOP condition outperformed the MEP condition on this measure, $t(81) = 3.56$, $p = .001$, $r = .37$.

Performance success within both model progression conditions was assessed at three points in time. Table 3.2 reports descriptive results for each assessment, indicating how the quality of the participants' models developed through time. For statistical analysis of this data it needs to be taken into account that not all participants progressed through all phases. Based on their progression through phases they were classified as either double phase changers (i.e., participants who worked in all three phases, $n = 29$), single phase changers (i.e., participants who

Table 3.2
Mean performance success scores in both model progression conditions by phase

	MOP			MEP		
	n	*M*	*SD*	*n*	*M*	*SD*
Model structure score (variables)						
Phase 1	28	6.46	2.50	26	4.08	1.55
Phase 2	26	7.04	2.68	20	5.25	1.92
Phase 3	12	6.92	2.47	17	6.82	2.94
Model structure score (relations)						
Phase 1	28	5.57	3.63	26	1.85	2.13
Phase 2	26	6.04	3.68	20	2.20	2.80
Phase 3	12	6.42	3.90	17	2.71	3.47

worked in Phase 1 and 2, $n = 17$) and no phase changers (i.e., participants who only worked in Phase 1, $n = 8$). As previously mentioned, model quality could not be assessed from the model content score for most of the MOP participants; therefore for the remainder of this results section the model structure score will be the only measure of performance success.

Prior to analysing how the quality of the participants' models developed over time and across conditions, it was examined whether the subset of participants who progressed to subsequent phases were representative of the entire sample in their experimental condition. Logistic regression analyses (using the Enter method) were conducted to determine whether transition to the second and third phase depended on the type of model progression and performance success. The models generated by the logistic regression predicting the phase changes are reported in Table 3.3. This data indicates that phase change was related to the type of model progression but not to performance success, indicating that the type of model progression influenced the speed of progression through phases. Furthermore, performance success was not a significant predictor of phase change, suggesting comparable performance success slopes for double, single, and no phase changers. This means that the students who progressed to higher phases performed as well

Table 3.3
Logistic regression-analyses (enter method) testing the dependence of progression speed on condition and performance success

Predictor	*B*	*SE*	Wald	d.f.	sig.	Exp(B)
Dropout phase 2[a]						
Condition	–2.37	1.24	3.64	1	.056	0.09
Model structure (variables aspect) phase 1	–0.05	0.26	0.04	1	.835	0.95
Model structure (relations aspect) phase 1	–0.18	0.17	1.12	1	.289	0.83
Constant	4.18	1.82	5.28	1	.022	65.09
Dropout phase 3[b]						
Condition	2.01	0.91	4.86	1	.028	7.46
Model structure (variables aspect) phase 1	0.12	0.27	0.21	1	.649	1.13
Model structure (relations aspect) phase 1	0.44	0.28	2.51	1	.113	1.55
Model structure (variables aspect) phase 2	–0.03	0.20	0.02	1	.882	0.97
Model structure (relations aspect) phase 2	–0.50	0.27	3.46	2	.063	0.61
Constant	–0.12	1.19	0.01	1	.919	0.89

[a] R^2 = .12 (Hosmer & Lemeshow), .09 (Cox & Snell), .15 (Nagelkerke). Model $\chi^2(3) = 4.86$, $p = .182$.
[b] R^2 = .21 (Hosmer & Lemeshow), .24 (Cox & Snell), .33 (Nagelkerke). Model $\chi^2(5) = 12.63$, $p = .027$.

as the students who remained in a lower phase. Consequently, analysis of performance success slopes of the double phase changers is likely to be representative for single phase changers' performance success slopes over Phase 1 and 2, and for the no phase changers' performance success over Phase 1.

A mixed-design MANOVA was performed to analyse how model structure scores for variables and relations evolved over the phases in each condition. Using Pillai's trace, MANOVA showed significant multivariate main effects for the between-subjects factor condition, $V = 0.27$, $F(2, 26) = 4.88$, $p = .016$, the within-subject factor phase, $V = 0.49$, $F(4, 24) = 5.76$, $p = .002$, and a significant Condition × Phase interaction, $V = 0.34$, $F(4, 24) = 3.09$, $p = .035$.

Subsequent univariate ANOVAs with the variables aspect of the model structure score as dependent variable indicated a non-significant main effect for condition, $F(1, 27) = 2.32$, $p = .139$, a significant main effect for phase, $F(2, 54) = 14.74$, $p < .001$, and a significant Condition × Phase interaction, $F(2, 54) = 5.28$, $p = .008$. The latter result indicates that the increase in model quality (variables aspect) over phases differed among MOP and MEP participants. Planned contrasts were performed to break down this interaction. The first contrast, comparing intermediate to initial scores, was not significant, $F(1, 27) = 0.24$, $p = .628$, meaning that both MOP and MEP participants' variables aspect of the model structure score slightly increased during Phase 2. From Table 2 it can be seen that scores in the MEP condition increased during Phase 3, whereas scores in the MOP condition remained relatively constant during this phase, Planned contrasts of the scores at the end of Phase 2 and 3 showed this difference to be statistically significant, $F(1, 27) = 7.71$, $p = .010$. Upon interpreting these results, it needs to be taken into account that participants in the MEP condition started with a partial simulation that progressed over phases to the full-complex simulation in Phase 3. Therefore, an increase in performance success is inherent to the experimental manipulation in the MEP condition, whereas in the MOP condition it is not.

Univariate ANOVAs with the relations aspect of the model structure score as dependent variable indicated both main effects to be significant (condition: $F(1, 27) = 9.86$, $p = .004$; phases: $F(1.69, 45.56) = 5.54$, $p = .010$), whereas their interaction was not, $F(1.69, 45.56) = 0.14$, $p = .836$. (As the sphericity assumption was violated, the Huynh-Feldt corrected degrees of freedom are reported). This indicates that the relation aspect of the model structure score in the MOP condition was generally higher than in the MEP condition. Furthermore, as shown in Table 3.2, there was a gradual increase in the relations' aspect of the model structure score over phases. Planned contrasts revealed that this increase was not significant during Phase 2, $F(1, 27) = 0.47$, $p = .497$, whereas it was significant during Phase 3, $F(1, 27) = 7.12$, $p = .013$. It is interesting to see that, although an increase is inherent to the manipulation in the MEP condition, the model structure score regarding the relations in the model does not show a different slope for the MEP condition compared to the MOP condition (as was the case for the variables aspect of model structure score).

Discussion

This study investigated the effects of model progression on students' performance during an inquiry learning task. Model progression in general was predicted to lead to higher performance success (Hypothesis 1). Furthermore, as model order progression was assumed to be more in keeping with domain novices' learning needs, participants in the MOP condition were expected to outperform those from the MEP condition (Hypothesis 2). Both predictions were generally supported by the results.

Evidence for Hypothesis 1 comes from the comparison of the two model progression conditions together with the control condition. Participants from both model progression conditions created more comprehensive models–as indicated by their model structure scores–than their control counterparts. Participants' model content scores further indicate that students in the MEP condition created more sophisticated models than students from the control condition. The predicted superiority of the MOP condition on this measure could not be shown, which is likely due to their slow progressing through phases.

Differences in performance success were also found between both model progression conditions. Consistent with Hypothesis 2, students from the MOP condition outperformed students in the MEP condition. Comparison of their final models indicated that MOP and MEP students were equally proficient in identifying which elements are relevant to their models (i.e., voltage source, light bulbs, and capacitor), whereas MOP participants more accurately modelled the relations between those elements. However, the predicted superiority of the MOP condition could not be shown on the model content score.

Existing research on the effectiveness of model progression paint a mixed picture, and the present findings could help explain why this is so. De Jong et al. (1999) previously proposed two conditions for model progression to be effective: a high level of task complexity and low levels of prior domain knowledge. The present study suggests that the type of model progression constitutes a third condition: in the physics domain of electrical circuits, model order progression was found to be more effective than model elaboration progression. The inconsistent results from prior research might therefore be attributable to the application of other, less effective types of model progression such as model fidelity progression (i.e., simulations going from 'simplified' to more 'realistic') and model elaboration progression (Alessi, 1995; Jackson, Stratford, Krajcik, & Soloway, 1994; de Jong et al., 1999; Swaak et al., 1998; Quinn & Alessi, 1994).

Even though all cited studies attempted to scaffold students on a science task, there is some evidence that the effectiveness of model order progression extends to different domains. In a recent study, Slof, Erkens, and Kirschner (2010) successfully applied model order progression to a business-economics task. They distinguished three models (a conceptual, causal, and simulation model) that essentially resemble the current study's model order progression phases. Students who consecutively received the three models performed better than students who only worked with one of these models throughout the entire session.

But do students who perform better also learn more? A knowledge posttest might have answered this question, but could not be included in the present study for practical reasons. Yet theoretical and empirical evidence suggests that the performance measures (i.e., model quality scores) that assessed the instructional effects of model progression are indicative of the knowledge students acquired during the experiment. Our students built a system dynamics model that was assumed to represent their knowledge of a charging capacitor. This assumption is based on constructionism, an instructional paradigm in which learning is considered synonymous to the knowledge construction that takes place when learners are engaged in building objects (Kafai & Resnick, 1996). Research has confirmed that the construction of models is associated with cognitive learning (e.g., van Borkulo, 2005), and that the quality of students' models is associated with their reasoning processes (Sins et al., 2005). It thus seems plausible that the superior performance of the MOP participants mirrors higher knowledge acquisition. Still, future research is needed to validate this claim, preferably through an independent measure of learning to supplement performance success measures.

Learning performance in the present study, although significant, was quite modest. The average model structure scores indicate that even the models created by participants in the MOP condition only partially reflected the contents of the domain. Future research might assess students additional support needs by addressing learning behaviour. A closer look at the quality of students' simulation and model experiments could provide more detailed information as to why elements and relations in the model are correct or not. From the present study it seems plausible that, given that relatively few participants reached the final progression phase, time on task was too short for students to create a full-fledged model. Task performance could accordingly be enhanced by either increasing time on task, or by promoting participants efficiency during the task.

Prior attempts to increase efficiency have tried to accompany model progression with assignments (de Jong et al., 1999; Swaak et al., 1998). These efforts turned out to be unsuccessful; future research should either continue along these lines or explore the effect of other types of additional support. One example would be to embed domain information in the learning environment. Lazonder, Hagemans, and de Jong (2010) found that this type of content support significantly enhances students' inquiry learning performance. Offering designated pieces of domain information in each model progression phase might accordingly improve the efficiency of students' modelling performance.

Efficiency could also be increased by fine-tuning the way model order progression is implemented. Model progression in this study followed the students' learning pace: if learners comprehended a phase, they could progress to the next phase. However, students progressed to consecutive phases with suboptimal models, suggesting that they progressed without full comprehension of the previous phase. This might have compromised the effectiveness of model progression which aims to keep the learning environment manageable by not introducing too many ideas at the same time (Swaak et al., 1998). For students who progressed to phases with suboptimal knowledge, the new phase is unlikely to be manageable. This might also have accounted for the large number of students who never progressed beyond Phase 2 in this study. As such, the instructional effectiveness of model order progression could be further enhanced by prohibiting phase-change until a minimal comprehension level is reached. This could be implemented by a restrictive software agent that functions as a gatekeeper at the phase-change points based on a model quality benchmark.

Restricting phase-changing nevertheless appears (and probably is) a counterintuitive way to help students progress through all phases. An alternative approach might be to further reduce phase change restrictions. Based on Klahr and Dunbar (1988), inquiry learning is often defined as consisting of three iterative processes: hypothesizing, experimenting, and evaluating evidence. The way students perform these processes is assumed to depend on their knowledge of the task at hand. Model order progression as implemented in this study only enabled students to iterate these processes within each phase. This suggests that model order progression might conflict with the iterative nature of the inquiry learning process; directions confirming this conflict can be found in the results. As model structure scores in the MOP condition were found to increase even in Phase 2 and 3, students apparently generated new or adjusted hypotheses on a Phase 1 level in subsequent phases. Therefore, an adjustment to model order progression, to the extent that students can freely navigate through the order dimension both forwards and backwards (i.e., iterative model order progression), might better suit the iterative nature of the inquiry learning process.

To conclude, this study points to the idea that model progression can foster learning performance, and that learners benefit most from model order progression. Future research is needed to investigate how model order progression can be further optimized, and two alternative approaches were proposed. One is to introduce a restrictive software agent that functions as gatekeeper at the phase-change points, the second alternative is to be less restrictive and allow students to wander across phases in any order they see fit.

References

Alessi, S. (1995). *Dynamic versus static fidelity in a procedural simulation*. Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco.

van Borkulo, S. P. (2009). *The assessment of learning outcomes of computer modelling in secondary science education*. Unpublished doctoral dissertation, University of Twente, Enschede.

Carroll, J. M., & Carrithers, C. (1984). Blocking learners error states in a training-wheels system. *Human Factors, 26*, 377-389.

Dean, D., & Kuhn, D. (2007). Direct instruction vs. discovery: The long view. *Science Education,* 91, 384-97. doi: 10.1002/sce.20194

Gagné, R. M. (1977). *The conditions of learning* (3rd ed.). New York: Holt, Rinehart & Winston.

Hogan, K., & Thomas, D. (2001). Cognitive comparisons of students' systems modeling in ecology. *Journal of Science Education and Technology, 10*, 319-345. doi:10.1023/A:1012243102249

de Jong, T. (2005). The guided discovery principle in multimedia learning. In R. E. Mayer (Ed.), *Cambridge handbook of multimedia learning* (pp. 215-229). Cambridge: Cambridge University Press. doi: 10.2277/0521838738

de Jong, T., Martin, E., Zamarro, J. M., Esquembre, F., Swaak, J., & van Joolingen, W. R. (1999). The integration of computer simulation and learning support: An example from the physics domain of collisions. *Journal of Research in Science Teaching, 36*, 597-615. doi: 10.1002/(SICI)1098-2736(199905)36:5<597::AID-TEA6>3.0.CO;2-6 de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*, 179-201. doi: 10.3102/00346543068002179

van Joolingen, W. R., de Jong, T., Lazonder, A. W., Savelsbergh, E. R., & Manlove, S. (2005). Co-Lab: Research and development of an online learning environment for collaborative scientific discovery learning. *Computers in Human Behavior, 21*, 671-688. doi: 10.1016/j.chb.2004.10.039

Kafai, Y. B., & Resnick, M. (Eds.). (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Mawhaw, NJ: Erlbaum.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12*, 1-48. doi: 10.1207/s15516709cog1201_1

Klahr, D., & Nigam, M. (2004). The equivalence of learning paths in early science instruction: Effects of direct instruction and discovery learning. *Psychological Science, 15*, 661-667. doi: 10.1111/j.0956-7976.2004.00737.x

Krajcik, J., Blumenfeld, P. C., Marx, R. W., Bass, K. M., Fredericks, J., & Soloway, E. (2000). Instructional, curricular, and technological supports for inquiry in science classrooms. In J. Minstrell & E. H. van Zee (Eds.), *Inquiring into inquiry learning and teaching in science* (pp. 283-315). Washington, DC: AAAS.

Lazonder, A. W., & van der Meij, H. (1995). Error-information in tutorial documentation: Supporting users' errors to facilitate initial skill learning. *International Journal of Human-Computer Studies, 42*, 185-206. doi: 10.1006/ijhc.1995.1009

Lazonder, A. W., Wilhelm, P., & van Lieburg, E. (2009). Unraveling the influence of domain knowledge during simulation-based inquiry learning. *Instructional Science*, *37*, 437-451. doi: 10.1007/s11251-008-9055-8

Lazonder, A. W., Hagemans, M. G., & de Jong, T. (2010). Offering and discovering domain information in simulation-based inquiry learning. *Learning and Instruction, 20,* 511-520. doi: 10.1016/j.learninstruc.2009.08.001

Löhner, S., van Joolingen, W. R., & Savelsbergh, E. R. (2003). The effect of external representation on constructing computer models of complex phenomena. *Instructional Science, 31*, 395-418. doi: 10.1023/A:1025746813683

Manlove, S., Lazonder, A. W., & de Jong, T. (2006). Regulative support for collaborative scientific inquiry learning. *Journal of Computer Assisted Learning, 22,* 87-98. doi: 10.1111/j.1365-2729.2006.00162.x

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American Psychologist, 59*, 14-19. doi: 10.1037/0003-066X.59.1.14

Mulder, Y. G., Lazonder, A. W., & de Jong, T. (2010). Finding out how they find it out: An empirical analysis of inquiry learners' need for support. *International Journal of Science Education, 32, 2033-2053,* doi: 10.1080/09500690903289993.

Ogborn, J. (1998). Cognitive development and qualitative modelling. *Journal of Computer Assisted Learning, 14,* 292-307. doi: 10.1046/j.1365-2729.1998.144069.x

Quinn, J., & Alessi, S. (1994). The effects of simulation complexity and hypothesis-generation strategy on learning. *Journal of Research on Computing in Education, 27*, 75-91.

Reigeluth, C., & Stein, F. S. (1983). The elaboration theory of instruction. In C. Reigeluth (Ed.), *Instructional design theories and models: An overview of their current status* (pp. 335-383). Hillsdale, NJ: Erlbaum.

Rieber, L. P., & Parmley, M. W. (1995). To teach or not to teach? Comparing the use of computer-based simulations in deductive versus inductive approaches to learning with adults in science. *Journal of Educational Computing Research, 13*, 359-374. doi: 10.2190/M8VX-68BC-1TU2-B6DV

Sins, P. H. M., Savelsbergh, E. R., & van Joolingen, W. R. (2005). The difficult process of scientific modeling: An analysis of novices' reasoning during computer-based modeling. *International Journal of Science Education, 14*, 1695-1721. doi: 10.1080/09500690500206408

Slof, B., Erkens, G., Kirschner, P. A., & Jaspers, J. G. M. (2010). Design and effects of representational scripting on group performance. *Educational Technology Research and Development, 58*, 589-608. doi: 10.1007/s11423-010-9148-3

Spannagel, C., Girwids, R., Löthe, H., Zendler, A., & Schroeder, U. (2008). Animated demonstrations and training wheels interfaces in a complex learning environment. *Interacting with Computers, 20*, 97-111. doi: 10.1016/j.intcom.2007.08.002

Stratford, S. J., Krajcik, J., & Soloway, E. (1998). Secondary students' dynamic modeling processes: Analyzing, reasoning about, synthesizing, and testing models of stream ecosystems. *Journal of Science Education and Technology, 7*, 215. doi: 10.1023/A:1021840407112

Swaak, J., van Joolingen, W. R., & de Jong, T. (1998). Supporting simulation-based learning: The effects of model progression and assignments on definitional and intuitive knowledge. *Learning and Instruction, 8*, 235-252. doi: 10.1016/s0959-4752(98)00018-8

Veenman, M. V. J., & Elshout, J. J. (1995). Differential effects of instructional support on learning in simulation environments. *Instructional Science, 22*, 363-383. doi: 10.1007/BF00891961

White, B. Y., & Frederiksen, J. R. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 42,* 99-157. doi: 10.1016/0004-3702(90)90095-h

White, B. Y., Shimoda, T. A., & Frederiksen, J. R. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. *International Journal of Artificial Intelligence in Education, 10*, 151-182.

Zacharia, Z.C., & Olympiou, G., (2011). Physical versus virtual manipulative experimentation in physics learning. *Learning and Instruction*, *21,* 317-331. doi: 10.1016/j.learninstruc.2010.03.001.

Chapter 4

Model progression: The influence of phase change restrictions[1]

Abstract

Model progression denotes the organization of a learning task in increasingly sophisticated phases. This study examined whether and how the conditions for changing model progression phases affects learning and performance in an inquiry learning task combined with modelling. Students in the 'standard' model progression condition ($n = 19$) could enter subsequent phases at will, and indeed did so more often than students from the *restricted* condition ($n = 20$) who could progress to subsequent phases only if sufficient knowledge had been acquired. Still, restricted participants who managed to progress to a subsequent phase did so with better models. Students from the *unrestricted* condition ($n = 22$) were free to enter subsequent and previous phases. They used these navigation options thoughtfully, and progressed to subsequent phases more often than students from the 'standard' condition. Even though both model progression variants influenced the learning process, they did not enhance students' performance success. It thus seems that neither more liberal nor more strict requirements to change model progression phases are sufficient to further improve the effectiveness of model progression. Additional support is needed for students to reach a full understanding of the learning domain.

[1] This chapter is based on: Mulder, Y. G., Lazonder, A. W., de Jong, T., Anjewierden, A., & Bollen, L. (2012). Validating and optimizing the effects of model progression in simulation-based inquiry learning. *Journal of Science Education and Technology*. Advance online publication. doi: 10.1007/s10956-011-9360-x

Introduction

Computer-based inquiry learning environments are known to foster a deeper understanding of a domain. These environments increasingly offer computer simulations and modelling facilities for students to explore science phenomena. As in authentic scientific inquiry, modelling is considered an integral part of inquiry learning in that students have to build models to express their understanding (van Joolingen, de Jong, Lazonder, Savelsbergh, & Manlove, 2005; White, Shimoda, & Frederiksen, 1999). This understanding can be tested by running the model; model data can then be compared with prior knowledge or data from the simulation, and these comparisons yield further insight into the phenomenon.

Most students, however, experience difficulties with this mode of learning, and accordingly exhibit suboptimal performance (Dean & Kuhn, 2007; Klahr & Nigam, 2004; Mayer, 2004; Mulder, Lazonder, & de Jong, 2010). A review by de Jong and van Joolingen (1998) gives an account of the intrinsic problems students face during simulation-based inquiry learning. For example, students cannot infer hypotheses from data, design inconclusive experiments, show inefficient experimentation behaviour, and ignore incompatible data. Similar problems arise during modelling. Hogan and Thomas (2001), for instance, noticed that students often fail to engage in dynamic iterations between examining output and revising models, and Stratford, Krajcik, and Soloway (1998) observed a lack of persistence in debugging models to fine-tune their behaviour.

These findings suggest that additional support is needed in order for inquiry learning and modelling to be more effective. This support can take several forms, such as content explanations (e.g., Lazonder, Hagemans, & de Jong, 2010), process prompts (e.g., Lin & Lehman, 1999) and direct instruction in inquiry skills (e.g., Klahr & Nigam, 2004). Model progression (White & Frederiksen, 1990) is probably the least intrusive form of support that aims to pave students' way through an inquiry by carefully structuring the task content according to a simple-to-complex sequence. Because of its unobtrusive nature, model progression complies with the self-directed way of learning that is pivotal to ingegrated approach to science education (see Chapter1). Model progression would entail that students start working on a simpler version of a task, and gradually progress to the more complex versions of that task. Such progressions aim to keep the learning environment manageable (Swaak, van Joolingen, & de Jong, 1998) and reduce the possibility that students get overwhelmed. As a result, students' inquiry efforts are more likely to be effective.

Model progression was indeed found to lead to higher performance success by learners in some studies (Alessi, 1995; Eseryel & Law, 2010; Rieber & Parmley, 1995; Swaak et al., 1998), but other studies report less favourable results (de Jong et al., 1999; Quinn & Alessi, 1994). Mulder, Lazonder, and de Jong (2011) reasoned that these differential effects are attributable to the way in which task complexity is increased. They compared two types of model progression: model order progression, which gradually increases the specificity of the relations between variables, and model elaboration progression, which gradually increases the number of variables in the task. The model order progression students generally performed better on the task than the model elaboration students.

These benefits notwithstanding, performance in the model order progression condition in the Mulder et al. (2011) study, left room for improvement. In terms of model quality, the students' final models showed just approximately 40 % understanding of the domain. Furthermore, few students progressed through all three model order progression phases: about 90% of students continued to the second phase, but only about 40 % of students progressed to the third –and final– phase. Analysis of students' intermediate models further showed that progressions from the first to the second phase often occurred with little knowledge. This might have challenged the effectiveness of model progression, which aims to keep the learning environment manageable by not introducing too many new ideas at the same time. If students progress to a subsequent phase with insufficient knowledge, the new phase is less manageable and can cause students to get stuck. This might have accounted for the large number of students in the Mulder et al. (2011) study who never progressed beyond the second phase.

In view of these problems, the effectiveness of model order progression could be further increased by restricting phase changes until sufficient knowledge has been acquired. Restricting phase-changing nevertheless appears (and probably is) a counter-intuitive way to help students progress through all phases. An alternative solution might be to broaden phase-change possibilities so as to allow students who get stuck in a particular phase to return to previous phases to remediate knowledge deficiencies. Inquiry learning is often defined as consisting of three *iterative* processes (hypothesising, experimenting, and evaluating evidence) that are shaped on the basis of the student's knowledge of the task (Klahr & Dunbar, 1988; Lazonder, Wilhelm, & Hagemans, 2008). Model order progression in the Mulder et al. (2011) study required students to iterate these processes within each phase, and this could be at odds with the iterative nature of inquiry learning. Therefore, an adjustment to model order progression where students can freely navigate forward and backward through the phases, seems more in keeping with the iterative nature of the inquiry learning process.

The present study aimed to explore the effects of both ways to improve model order progression. The basic premise underlying this research was that model order progression could be improved by enhancing students' performance within the phases. Both restricting phase changes until students have acquired sufficient domain knowledge, and removing phase-change restrictions by allowing free navigation through the phases –both forward and backward– were hypothesized to enhance students' performance. This assumption was investigated in a between-subjects design with three conditions. Model order progression was implemented in all three conditions, and divided the inquiry learning task in three consecutive phases. Students in the *semi-restricted* condition could progress to subsequent phases as they pleased (i.e., there were no performance benchmarks for progressing), but could not return to a previous phase. To investigate the influence of phase-change restrictions, performance in this condition was compared with two experimental groups. In one group, the *restricted* condition, students could only progress to the next phase if they had reached sufficient understanding. Returning to a previous phase was not possible in this condition. In the second group, the *unrestricted* condition, there were no phase-change restrictions at all, so students could navigate through phases both forwards and backwards as they pleased. Phase changes in this condition were allowed irrespective of participants' domain knowledge.

It was hypothesized that both restricting forward phase-changing based on acquired knowledge and reducing phase-change restrictions would increase participants' performance success and their chances of successfully progressing through all phases. Two sets of pairwise comparisons were made to assess the influence of phase-change restrictions on the effectiveness of model progression. A comparison of performance success among the restricted and the semi-restricted condition served to establish the effects of forward phase-change restrictions. The comparison of the semi-restricted condition with the unrestricted condition assessed the advantages of backward phase-change restrictions.

Method

Participants

The study's initial sample consisted of 72 Dutch high-school students from a science track, aged 15-17. However, as 11 students were absent due to illness and timetable difficulties during one of the sessions, analyses were performed with 61 participants. A review of school curricula and teacher statements showed that the charging of electrical capacitors, which was the topic of inquiry, had not yet been taught in these students' physics classes. A pretest was administered to confirm that participants were indeed domain novices; class-ranked pretest scores were used to assign students to either the restricted condition ($n = 20$), the semi-restricted condition ($n = 19$), or the unrestricted condition ($n = 22$).

Materials

Inquiry task and learning environment

All participants worked on an inquiry task about the charging of a capacitor in an electrical circuit. Their assignment was to examine and model the behaviour of each element in the electrical circuit presented in a simulation (i.e., a voltage source, two light bulbs, and a capacitor). Participants performed this task within a modified stand-alone version of the Co-Lab learning environment (van Joolingen et al., 2005) that stored all participants' actions in a log file.

Model order progression was implemented by dividing the task into three phases that involved increasingly specific reasoning. The initial phase only dealt with the model structure; students had to indicate the elements and the relationships between the elements (but not specify these relationships). The consecutive phases dealt with the model content; students had to examine the simulation and specify the relationships between the elements in their model qualitatively in Phase 2 and quantitatively in Phase 3. Changing the model structure by adding and deleting elements and relations (which was central to Phase 1) was still possible in these phases, although these relations now did need specification.

Participants worked on the model in the model editor tool (see Figure 4.1) which allowed for system dynamics modelling. Over phases, students worked on the

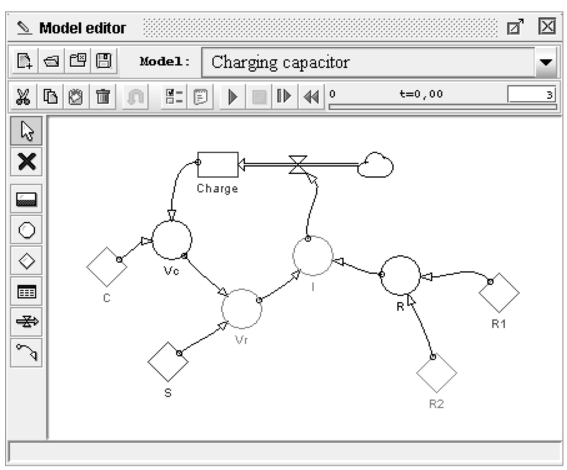

Figure 4.1. Screen capture of the model editor tool with the reference model.

same model, adding the new phase-specific demands to that model. As can be seen from Figure 4.1, such models have a graphical structure that consists of variables and relations. Variables are the constituent elements of a model and can be of three different types: variables that remain constant (i.e., constants), variables that specify the integration of other variables (i.e., auxiliaries), and variables that accumulate over time (i.e., stocks). Relations express propositions about how two or more variables interact. Participants' could express the content of these propositions qualitatively (Phase 2) by selecting a pre-specified, qualitative relation from a drop-down menu, or quantitatively (Phase 3) by inserting scientific equations.

Students could analyse their model output through a bar chart, table, or graph tool. As without specification of the relationships it is technically impossible to run the model in Phase 1, a bar chart tool was provided to analyse model structure. The bar chart displayed the number of correct and incorrect elements and relations in the model. The correctness of variables and relations was assessed by a software agent that compared the elements and relations in the students' models to a reference model. The bar chart tool was available in Phase 1 only. The table and graph tool allowed students to analyse model content by comparing model and simulation output in a single window. Students could use the results of this comparison to adjust or fine-tune their model and thus their understanding of electrical circuits.

An embedded help file tool contained the assignment and offered explanations of the operation of the tools in the learning environment. The help files contained no domain information on electrical circuits and capacitors as students should infer this knowledge from interacting with the simulation. The help files also informed participants about the specifics of the condition they were assigned to.

Variants of the learning environment for the different conditions

All conditions used the same instructional content (i.e., electrical circuits) that was divided into three phases according to the principles of model order progression. The three conditions differed only with regard to the restrictions to enter subsequent or previous phases. Table 4.1 presents an overview of the (im)possibilities of changing phases in each condition.

Participants in the *restricted* condition could not return to previous phases, and could progress to the next phase only if their model was of sufficient quality. Students were free to *attempt* to go to next phases whenever they liked but a

Table 4.1
Phase-change possibilities in each condition

Condition	Phase-change possibilities	
	To next phase	To previous phase
Restricted	Yes (but only if student's model is of sufficient quality)	No
Semi-restricted	Yes	No
Unrestricted	Yes	Yes

software agent functioned as a gatekeeper, restricting phase changes in case the students' model did not meet the requirements of a predefined rule set.

The rule set was based on the similarity between the student's model and the reference model. Minimal requirements for the transition from Phase 1 to Phase 2 were the presence of all but one of the constant and stock elements (C, S, R_1, R_2, and charge), one auxiliary element (V_c, V_r, I, or R), and all relationship arrows between these elements. For the transition from Phase 2 to Phase 3, this rule set was extended with the requirement to have a correct, qualitative specification for all but one of the relation arrows.

Participants in the *semi-restricted* condition were also prohibited to return to previous phases. However, they could progress to subsequent phases at will and without any restrictions imposed by the software agent. As such, the forward progression was completely learner controlled.

The *unrestricted* condition had no phase-change restrictions at all. Participants in this condition were free to go to subsequent and previous phases as they deemed fit. As such, both the forward *and* backward progressions were completely learner-controlled.

Pretest

A pretest consisting of eight open questions assessed participants' prior knowledge of electrical circuits. Four questions addressed the meaning of key domain concepts (i.e., voltage source, resistance, capacitor, and capacitance), the other four items addressed the knowledge about the charging of a capacitor in an electrical circuit (i.e., Ohms law, Kirchoff's law (including its two rules: the junction rule and the loop rule), and the behaviour of capacitors). As performance on the pretest was expected to be low, three simple filler items on the interpretation of numerical data were added to sustain students' motivation during the test. These filler items were left out of the analysis. Participants' answers to the eight questions were scored using the rubric of Mulder et al. (2011), which allocates one point to each correct response. Inter-rater reliability reached .89 (Cohen's κ).

Procedure

All participants engaged in two sessions: a 50-minute introduction and a 100-minute experimental session. The time between sessions was one week maximum. During the introductory session, participants first completed the pretest, then received a guided tour of the Co-Lab learning environment, and finally completed a brief tutorial that familiarised them with the system dynamics modelling language and the operation of the modelling tool.

During the experimental session, participants were presented with the inquiry task about the charging of a capacitor in an electrical circuit. Students had to examine and model the behaviour of each element in the electrical circuit as presented in the simulation in Co-Lab. Students were directed to begin by reading the assignment and to work individually. They were instructed about the different variants of the learning environments and were told to look at the help files to find out more about the condition they were assigned to. During the assignment, they could ask the experimenter for technical assistance only. Participants could stop ahead of time if they had completed the assignment.

Coding and scoring

All data were assessed from the log files. Variables under investigation were time on task, learning activities, phase-changes, and performance success. Time on task concerned the duration of the experimental session. Learning activities were defined by the number of times participants clicked the "Start" button in the simulation (simulation experiment) or model editor (model experiment).

Regarding phase-changes, a distinction was made between phase-change attempts and actual phase-changes. Phase-change attempts were defined as the number of times participants expressed a desire to go to another phase, as indicated by a click on the "next phases" button in the learning environment. Actual phase-changes were the number of times these attempts were successful. Actual phase-changes were further classified as either forward or backward progressions.

To assess performance success, both a model structure and a model content score were calculated. For participants' final and intermediate (i.e., at phase-changes) models, a model structure score was computed in accordance with Manlove, Lazonder, and de Jong's (2006) model coding rubric. This score represents the number of correctly specified variables and relations in the models. "Correct" was judged from the reference model shown in Figure 4.1. One point was awarded for each correctly named variable; an additional point was given if that variable was of the correct type. Concerning relations, one point was awarded for each correct link between two variables and one point was awarded for the direction. The maximum model structure score was 38. The rubric's inter-rater reliability for variables (Cohen's $\kappa = .74$) and relations (Cohen's $\kappa = .92$) was considered to be sufficient.

As the model structure score leaves the quantitative aspects of the model unaddressed, a complementary final model content score was calculated. This score represented participants' understanding of the physics equations that govern the behaviour of a charging capacitor (i.e., Ohms Law: $I = V / R$; resistances connected in parallel: $1 / R_t = 1 / R_1 + 1 / R_2$; the potential difference in the circuit depends on the power source and the potential difference across the capacitor: $\Delta V = V_s - V_c$; and the relationship between the potential difference across the capacitor and the amount of charge that gathers on the capacitor: $C = Q / V_c$). In a correct, fully-specified model these components are correctly integrated as represented in Equation 1:

$$(dQ / dt) = (V_s - Q / C) * (1 / R_1 + 1 / R_2) \qquad (1)$$

One point was awarded for each correctly specified part, leading to a four-point maximum score. A prior study (Mulder et al., 2010) found a 1.0 inter-rater reliability (Cohen's κ).

Results

Table 4.2 summarizes the descriptive statistics for participants' performance by condition. Univariate analysis of variance (ANOVA) on the pretest scores revealed no significant differences in prior knowledge between the conditions, $F(2, 58) = 0.67$, $p = .517$. The time on task scores from Table 4.2 further show that, on average, participants in each condition spent 85 to 90 minutes working on the assignment. ANOVA indicated that the minor cross-condition differences in time were not statistically significant, $F(2, 58) = 1.65$, $p = .201$.

Table 4.2
Summary of participants' performance

	Restricted (n = 20)		Semi-restricted (n = 19)		Unrestricted (n = 22)	
	M	*SD*	*M*	*SD*	*M*	*SD*
Pretest score	1.65	1.46	1.21	1.13	1.27	1.28
Time on task (min.)	85.36	11.15	86.58	14.02	90.99	5.47
Performance success						
Model content score	0.00	0.00	0.00	0.00	0.00	0.00
Model structure score– variables[a]	9.85	4.06	8.00	3.33	9.86	2.77
Model structure score– relations[b]	6.50	4.95	6.74	4.40	7.73	3.41
Learning activities						
Number of simulation experiments	15.95	12.77	12.47	6.50	13.45	8.94
Number of model experiments	63.70	38.40	68.47	51.25	63.59	34.64

[a]Maximum score = 18. [b]Maximum score = 20.

Performances success was assessed from the participants' final models. The model content scores displayed in Table 4.2 implicate that none of the participants reached a correct, quantitative understanding of the physics equations. This could have been due to the low number of participants who reached phase 3 where the quantitative relations were addressed (see Table 4.3). Still, in absence of any variation in scores, the model content measure was left out of further analysis. Model structure scores were analysed by Multivariate Analyses of Variance (MANOVA). Results showed no significant difference between experimental conditions on this measure, $V = 0.10$, $F(4, 116) = 1.55$, $p = .192$. This means that, overall, participants in all three conditions performed equally successfully. To check the validity of these outcomes, the model structure scores in the semi-restricted condition were compared with those of participants in the Mulder et al. (2011) study who worked on the same task and were supported by the same form of model order progression. These students had an average variable score of 6.82 ($SD = 2.51$) and a mean relation score of 6.18 ($SD = 3.21$). MANOVA revealed no significant difference between these scores, $V = 0.04$, $F(2, 44) = 0.98$, $p = .384$, indicating that the quality of the students' models in both studies were comparable in terms of variables and relations.

Participants' learning activities were defined by the number of simulation experiments and model runs. MANOVA showed no significant difference

Table 4.3
Mean model structure score at phase change by condition

	Restricted			Semi-restricted			Unrestricted[a]		
	n	*M*	*SD*	*n*	*M*	*SD*	*n*	*M*	*SD*
	From Phase 1 to Phase 2								
Model structure score (variables)	6	13.17	2.99	15	8.20	3.76	20	8.85	3.30
Model structure score (relations)	6	11.17	3.55	15	6.47	4.26	20	6.05	4.49
	From Phase 2 to Phase 3								
Model structure score (variables)	0	–	–	5	6.80	3.56	9	9.67	3.24
Model structure score (relations)	0	–	–	5	4.00	2.83	9	7.89	3.41

[a] Scores represent the mean model structure scores upon first entering a subsequent phase.

between experimental conditions regarding these measures, V = 0.03, $F(4, 116) = 0.39$, $p = .813$. This means that the different phase change restrictions had no significant effect on the number of simulation experiments and model runs participants performed.

Analysis of the learning process addressed the issue of whether and how phase-change restrictions influenced participants' attempts to visit subsequent and previous phases. Participants in the *restricted* condition, who could only progress to a subsequent phase if sufficient knowledge had been acquired, tried to change phases 5.21 times on average ($SD = 7.31$; Range: 0-27). (One participant had 81 unsuccessful phase change attempts and was left out of this analysis). Access to Phase 2 was granted on six occasions, and no participants in this condition reached Phase 3. MANOVA confirmed that the six participants who progressed to Phase 2 (variables: $M = 13.17$, $SD = 3.00$; relations: $M = 11.17$; $SD = 3.55$) performed better than the participants who were not granted access to Phase 2 (variables: $M = 8.43$, $SD = 3.65$; relations: $M = 4.43$, $SD = 3.92$), V = 0.45, $F(2, 17) = 7.02$, $p = .006$. Subsequent univariate ANOVA's showed that the participants that did progress to Phase 2 performed better both on the variables aspect $F(1, 18) = 7.78$, $p = .012$ and on the relations aspect $F(1, 18) = 13.09$ $p = .002$.

Participants in the *semi-restricted* condition were free to progress to subsequent phases. Fifteen of the 19 participants in this condition entered Phase 2, and 5 of them went on to Phase 3. Participants in the *unrestricted* condition could enter subsequent and previous phases as they pleased. Two out of the 22 participants never progressed to Phase 2, and 4 out of the remaining 20 participants never made use of the opportunity to regress to a previous phase. This means that 16 participants utilized the unrestricted phase-change possibilities as intended. These 16 participants changed phases 7.94 times on average ($SD = 4.48$; Range: 2-19).

Approximately half of these attempts (54%) were progressions, and 46% were regressions. Most phase-changes (79%) occurred between Phases 1 and 2.

Table 4.3 summarizes the number of participants who progressed through phases. There was a significant association between the type of restriction and whether or not participants reached Phase 2, χ^2 (2, N = 61) = 19.36, p = .006. Participants from the unrestricted condition were most likely to enter Phase 2. Based on the odds ratio, the chance that participants would enter Phase 2 was 2.67 times higher if they were in the unrestricted condition than in the semi-restricted condition, and 8.75 times higher if they were in the semi-restricted condition than in the restricted condition. The transition from Phase 2 to Phase 3 was independent of the type of phase change restriction, χ^2 (2, N = 41) = 4.16, p = .125.

Table 4.3 also shows the mean model structure scores at each phase change. MANOVA with both model structure aspects (i.e., variables and relations) as dependent variables produced a significant effect of condition at the first phase-change point (Phase 1 to Phase 2), V = 0.23, $F(4, 76) = 2.50$, $p = .050$, but not at the second (Phase 2 to Phase 3), V = 0.28, $F(2, 11) = 2.15$, $p = .163$. Subsequent univariate ANOVAs indicated that, at the first phase-change point, phase-change restrictions significantly affected the number of correct variables in the students' model, $F(2, 38) = 4.72$, p = . 015, and the quality of the relations between these variables, $F(2, 38) = 3.43$, $p = .043$. Planned contrasts revealed that the restricted participants scored higher on both aspects (variables: $t(38) = 4.97$, $p = .005$; relations: $t(38) = 4.70$, $p = .029$) than the semi-restricted participants. A comparison of the semi-restricted participants with the unrestricted participants did not indicate significant differences (variables: $t(38) = 0.65$, $p = .583$; relations: $t(38) = -0.42$, $p = .778$).

Discussion

Prior research showed that model order progression promotes inquiry learning (Mulder et al., 2011). The aim of the present study was to determine whether the use of either more liberal or more strict requirements to enter model progression phases can further enhance this type of support. Two alternatives to the 'standard' form of model order progression (i.e., semi-restricted condition) were examined: restricting phase changing until students acquire sufficient domain knowledge (restricted condition), and allowing free navigation through the phases – both forwards and backwards – (unrestricted condition).

As in the Mulder et al. (2011) study, students in the semi-restricted condition were allowed to progress to subsequent phases at will, but could not return to a previous phase. Performance success in the semi-restricted condition was comparable to that in the Mulder et al. study, which validates the use of this condition as a comparison condition in the present research. The main conclusion of this study is that both alternative phase-change restrictions have no significant influence on participants' overall performance success as indicated by the models they created. Nonetheless, even though phase-change restrictions were not as beneficial as hypothesized, they did influence students' performance.

Students in the restricted condition were prohibited from progressing to the next phase until they had reached sufficient understanding, and could not return to a previous phase. Restricting phase-changing to subsequent phases based on intermediate performance success was presumed to enhance the effectiveness of model progression as only students with sufficient knowledge would progress to subsequent phases. This was partially confirmed by the results: the restricted students often attempted to progress to Phase 2 with little knowledge as judged by the software agent. This suggests that the students misconceived their knowledge as being sufficient to progress to the next phase. Their phase-change attempts were blocked by the software agent until a knowledge benchmark was reached. As a result, the restricted students who managed to progress to Phase 2 had more elaborate models than those who did not. A comparison with the semi-restricted participants further revealed that the restricted students were less likely to enter Phase 2. Still, restricted students who entered Phase 2 did so with more elaborate models than the semi-restricted participants. The alleged advantage of performance-based phase-change restrictions is that it increases new phases' manageability, as it restricts students from progressing until they have sufficient knowledge. Unfortunately however, none of the participants in the restricted condition reached Phase 3, thus this study could not confirm this advantage.

These findings seem to indicate that performance-based phase-change restrictions do not increase the effectiveness of model progression. An alternative interpretation, however, is that the effectiveness of these restrictions failed to show because of time constraints. Imposing performance standards inevitably increased students' time on task in Phase 1. This is actually a common problem in instructional approaches that restrict students based on their performance, see for example Bloom's (1968) work on mastery learning (Arlin, 1984). It thus seems plausible that performance-based restrictions can enhance model order progression if students are allowed more time to complete the task. Future research could continue along these lines and investigate whether without these time constraints, restricting phase changing based on performance will increase performance success.

A related question for future research would be how restrictive the software agent should be. The software agent in this study was rather strict: it granted access to subsequent phases only if all but one of the new phases' requirements were met. This appeared to delay progressions for quite a long time. A more permissive software agent might enable faster progression through the phases while maintaining manageability of the learning content. Striking a balance between these two issues is an interesting challenge for future studies.

Students in the unrestricted condition were allowed to progress to subsequent phases as well as regress to previous phases as they saw fit. This appeared to be a more effective way to ensure progression through all phases. Compared to the semi-restricted condition, students in the unrestricted condition were nearly three times more likely to reach Phase 2. Also, more students from the unrestricted condition reached Phase 3, but this difference was not significant. However, the quality of the unrestricted participants' intermediate models did not differ from the quality of the semi-restricted participants' intermediate models. This suggests that even for the unrestricted students –who knew that they could undo a phase-change any time– each phase change was a conscious choice.

On average, students navigated between the three phases approximately eight times, most often between Phases 1 and 2. Nearly half of these phase changes were regressions in phases, suggesting that the students acknowledged a knowledge gap for the higher phase and regressed to a previous phase to remediate it. Most students who used the freedom to regress to previous phases also changed their model structure, which most often lead to improvements to the model. However, these improvements were not substantial enough to yield an overall cross-condition effect on performance success.

The effects of free navigation has also been studied in hypermedia research regarding the effects of learner control. Among other things, this research lets learners determine the order in which they would like to access different information units and decide over the pace of information presentation, including returning to previous information units (alike the possibility to regress to previous phases for the unrestricted condition) (Scheiter & Gerjets, 2007). Even though increased learner control is considered a major advantage of hypermedia learning, Scheiter and Gerjets conclude that empirical evidence supporting this claim is still scant (cf. Dillon & Gabbard, 1998; Johnson, Perry, & Shamir, 2010). It can be hypothesized that novice learners lack the skills to navigate hypermedia environments. Although the present study provided the students with an inherent implicit learning sequence, the increased learner control also did not lead to higher performance success.

As for practical implications, the results of this study should be regarded in perspective with previous findings on model progression. The success of model progression depends on the way complexity is gradually increased (Mulder et al., 2011). However, even when model order progression is applied, students' performance on inquiry learning tasks combined with modelling leaves room for improvement. That is, only few students progressed through all phases, and their final models demonstrate an incomplete understanding of the task domain. Unfortunately, both alternative phase-change restrictions explored in this study did not adequately settle these problems for all students. It can therefore be concluded that model order progression should be supplemented with additional support.

This additional support could take several forms. Prior research failed to demonstrate a performance increase when model progression was complemented with assignments (de Jong et al., 1999; Swaak et al., 1998). Future research should investigate which additional support is synergistic rather than redundant when combined with model order progression. Students' spontaneous reactions during this study suggest that worked examples qualify as good candidate for this additional support. Students' appeared to lack confidence in their approach to the task and as a result they lost a lot of time figuring out what to do and applying unproductive strategies. Worked examples have long since been found effective in initial skill learning for problem-solving tasks (e.g. Atkinson, Derry, Renkl, & Wortham, 2000; Paas & van Merriënboer, 1994; Sweller & Cooper, 1985) and more recently the application of worked examples has effectively been expanded to more complex learning tasks (Hilbert & Renkl, 2009; Hilbert, Renkl, Kessler, & Reiss, 2008; van Gog, Paas, & van Merriënboer, 2008). It would be interesting to investigate the instructional efficacy of worked examples additional to model order progression in an inquiry learning task combined with modelling.

To conclude, the results of this study indicate that phase-change restrictions influence students' behaviour and performance on an inquiry learning task combined with modelling. However, this influence is too small to enhance overall performance for all students. Future research should explore the effects of additional support.

References

Alessi, S. (1995). *Dynamic versus static fidelity in a procedural simulation*. Paper presented at the Annual meeting of the American Educational Research Association, San Francisco.

Arlin, M. (1984). Time, equality, and mastery learning. *Review of Educational Research, 54*, 65-86.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research, 70*, 181-214. doi: 10.3102/00346543070002181

Bloom, B. S. (1968). Learning for mastery. *Evaluation Comment UCLA-CSIEP, 1*, 1-12.

de Jong, T., Martin, E., Zamarro, J. M., Esquembre, F., Swaak, J., & van Joolingen, W. R. (1999). The integration of computer simulation and learning support: An example from the physics domain of collisions. *Journal of Research in Science Teaching, 36*, 597-615. doi: 10.1002/(SICI)1098-2736(199905)36:5<597::AID-TEA6>3.0.CO;2-6

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*, 179-201. doi: 10.3102/00346543068002179

Dean, D., & Kuhn, D. (2007). Direct instruction vs. discovery: The long view. *Science Education, 91*, 384-397. doi: 10.1002/sce.20194

Dillon, A., & Gabbard, R. (1998). Hypermedia as an educational technology: A review of the quantitative research literature on learner comprehension, control, and style. *Review of Educational Research, 68*, 322-349. doi: 10.3102/00346543068003322

Eseryel, D., & Law, V. (2010). *Promoting learning in complex systems: Effect of question prompts versus system dynamics model progressions as a cognivie-regulation scaffold in a simulation-based inquiry-learning envornment*. Paper presented at the 9th International Conferences of the Learning Sciences, Chicago, IL.

Hilbert, T. S., & Renkl, A. (2009). Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps. *Computers in Human Behavior, 25*, 267-274. doi: 10.1016/j.chb.2008.12.006

Hilbert, T. S., Renkl, A., Kessler, S., & Reiss, K. (2008). Learning to prove in geometry: Learning from heuristic examples and how it can be supported. *Learning and Instruction, 18*, 54-65. doi: 10.1016/j.learninstruc.2006.10.008

Hogan, K., & Thomas, D. (2001). Cognitive comparisons of students' systems modeling in ecology. *Journal of Science Education and Technology, 10*, 319-345. doi: 10.1023/A:1012243102249

Johnson, E. P., Perry, J., & Shamir, H. (2010). Variability in reading ability gains as a function of computer-assisted instruction method of presentation. *Computers & Education, 55*, 209-217. doi: 10.1016/j.compedu.2010.01.006

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12*, 1-48. doi: 10.1207/s15516709cog1201_1

Klahr, D., & Nigam, M. (2004). The equivalence of learning paths in early science instruction. *Psychological Science, 15*, 661-667. doi: 10.1111/j.0956-7976.2004.00737.x

Lazonder, A. W., Hagemans, M. G., & de Jong, T. (2010). Offering and discovering domain information in simulation-based inquiry learning. *Learning and Instruction, 20*, 511-520. doi: 10.1016/j.learninstruc.2009.08.001

Lazonder, A. W., Wilhelm, P., & Hagemans, M. G. (2008). The influence of domain knowledge on strategy use during simulation-based inquiry learning. *Learning and Instruction, 18*, 580-592. doi: 10.1016/j.learninstruc.2007.12.001

Lin, X., & Lehman, J. D. (1999). Supporting learning of variable control in a computer-based biology environment: Effects of prompting college students to reflect on their own thinking. *Journal of Research in Science Teaching, 36*, 837-858. doi: 10.1002/(SICI)1098-2736(199909)36:7<837::AID-TEA6>3.0.CO;2-U

Manlove, S., Lazonder, A. W., & de Jong, T. (2006). Regulative support for collaborative scientific inquiry learning. *Journal of Computer Assisted Learning, 22*, 87-98. doi: 10.1111/j.1365-2729.2006.00162.x

Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction. *American Psychologist, 59*, 14-19. doi: 10.1037/0003-066X.59.1.14

Mulder, Y. G., Lazonder, A. W., & de Jong, T. (2010). Finding out how they find it out: An empirical analysis of inquiry learners' need for support. *International Journal of Science Education, 32*, 2033-2053. doi: 10.1080/09500690903289993

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. doi: 10.1016/j.learninstruc.2011.01.003

Paas, F. G. W. C., & van Merriënboer, J. J. G. (1994). Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive load approach. *Journal of Educational Psychology, 86*, 122-133.

Quinn, J., & Alessi, S. (1994). The effects of simulation complexity and hypothesis-generation strategy on learning. *Journal of Research on Computing in Education, 27*, 75-91.

Rieber, L. P., & Parmley, M. W. (1995). To teach or not to teach? Comparing the use of computer-based simulations in deductive versus inductive approaches to learning with adults in science. *Journal of Educational Computing Research, 13*, 359-374. doi: 10.2190/M8VX-68BC-1TU2-B6DV

Scheiter, K., & Gerjets, P. (2007). Learner control in hypermedia environments. *Educational Psychology Review, 19*, 285-307. doi: 10.1007/s10648-007-9046-3

Stratford, S. J., Krajcik, J., & Soloway, E. (1998). Secondary students' dynamic modeling processes: Analyzing, reasoning about, synthesizing, and testing models of stream ecosystems. *Journal of Science Education and Technology, 7*, 215. doi: 10.1023/A:1021840407112

Swaak, J., van Joolingen, W. R., & de Jong, T. (1998). Supporting simulation-based learning; the effects of model progression and assignments on definitional and intuitive knowledge. *Learning and Instruction, 8*, 235-252. doi: 10.1016/s0959-4752(98)00018-8

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2*, 59-89.

van Gog, T., Paas, F., & van Merriënboer, J. J. G. (2008). Effects of studying sequences of process-oriented and product-oriented worked examples on troubleshooting transfer efficiency. *Learning and Instruction, 18*, 211-222. doi: 10.1016/j.learninstruc.2007.03.003

van Joolingen, W. R., de Jong, T., Lazonder, A. W., Savelsbergh, E. R., & Manlove, S. (2005). Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning. *Computers in Human Behavior, 21*, 671-688. doi: 10.1016/j.chb.2004.10.039

White, B. Y., & Frederiksen, J. R. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 42*, 99-157. doi: 10.1016/0004-3702(90)90095-h

White, B. Y., Shimoda, T. A., & Frederiksen, J. R. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. *International Journal of Artificial Intelligence in Education, 10*, 151-182.

Chapter 5

The added value of worked examples to support students on an inquiry learning task combined with modelling[1]

Abstract

Recent studies on the effect of model progression (i.e., gradually increasing a tasks' complexity) have shown that students need additional support on an inquiry learning task combined with modelling. This study investigated the effect of heuristic worked examples as additional support. High-school students in the experimental condition (*n* = 46) could consult heuristic worked examples that explained what activities were needed and how they should be performed. Students in the control condition (*n* = 36) did not receive this support. Results showed that students in the experimental group exhibited more proficient inquiry and modelling behaviour, performed better, but did not learn more than their control counterparts.

[1] This chapter is based on: Mulder, Y. G., Lazonder, A. W., & de Jong, T. (2012). *The added value of worked examples to support students on an inquiry learning task combined with modelling*. Manuscript in preparation for publication.

Introduction

Recent meta-analyses conclude that inquiry learning can benefit students and lead to superior student performance than more direct forms of instruction (Alfieri, Brooks, Aldrich, & Tenenbaum, 2011; Minner, Levy, & Century, 2010). However, these meta-analyses also conclude that these benefits only hold when students are supported during their inquiry activities. Students need this support to compensate for their modest inquiry skills or prior knowledge deficits. De Jong and Van Joolingen's (1998) review revealed a broad variety of skill deficiencies in simulation-based inquiry learning. Among the most pertinent problems are students' inability to infer hypotheses from data, design conclusive experiments, engage in efficient experimentation behaviour, and attend to incompatible data. Similar problems arise when students create computer models of scientific phenomena. Hogan and Thomas (2001), for instance, noticed that students often fail to engage in dynamic iterations between examining output and revising models, and Stratford, Krajcik, and Soloway (1998) observed a lack of persistence in debugging models to fine-tune their behaviour.

Mulder, Lazonder, and de Jong (2010) examined whether these results generalize to a learning task where inquiry and modelling are combined. They had high school students infer the model underlying a simulation of an electrical circuit through systematic experimentation, and rebuild this model to express their understanding of the variables and relations in the simulation. Students could verify their knowledge of the electrical circuit by running their model and weighting its output against prior knowledge or data from the simulation. This study indicated that domain novices are quite capable of identifying relevant variables, but have difficulty inferring how these variables are related. Instead of gradually working toward a full-fledged scientific equation to specify a relationship, novices tried to induce and model these equations from scratch, which proved ineffective given their lack of prior domain knowledge. These findings suggest that students could benefit from support that prevents them from 'jumping the gun' and better attunes their inquiry and modelling activities to their level of domain knowledge (cf. Quintana et al., 2004).

This support can be offered in a non-intrusive way by organizing the learning task according to a simple-to-complex sequence that matches the students' increasing levels of domain understanding. This type of task structuring was first introduced by White and Frederiksen (1990), who termed it 'model progression'. Model progression was found to lead to higher performance success in some studies (Alessi, 1995; Eseryel & Law, 2010; Rieber & Parmley, 1995; Swaak, van Joolingen, & de Jong, 1998), but other studies report less favourable results (de Jong et al., 1999; Quinn & Alessi, 1994). These differential effects might be attributable to the slightly different configurations of the simple-to-complex sequencing. Some studies had students engage in increasingly specific reasoning about the task content (i.e., model order progression) whereas students in other studies engaged in specific reasoning about increasingly more elaborate task content (i.e., model elaboration progression). Mulder, Lazonder, and de Jong (2011) compared both types of model progression on inquiry learning task combined with modelling. Students who were supported by either type of model progression outperformed students from a unsupported control condition. A comparison among the two model progression variants further showed that students who investigated and created increasingly more specific models outperformed students who investigated and created increasingly more elaborate models.

However, even students in the best-performing model progression group produced mediocre models. One reason could be that few students completed all three phases of the task sequence. Analysis of the students' learning activities and models revealed that many students progressed from the first to the second phase, but few went on to the third -and final- phase. Those who got stuck in the second phase entered this phase with a rather simple model, which probably provided an insufficient basis for the complex task at hand. In an attempt to optimize model progression, the study described in Chapter 4 explored the effects of either broadening or narrowing students' possibilities to choose their own learning paths through the pre-defined task sequence. Neither of these adjustments enhanced the quality of the students' models, and observation during the lessons revealed that many students performed the inquiry learning task ineffectively and inefficiently. These findings suggest that students might need a more explicit account of what the activities in each model progression phase entail and how they should be performed.

Such support could take the form of worked examples which are a proven fruitful means to enhance problem-solving performance (e.g., Paas & van Merriënboer, 1994; Sweller & Cooper, 1985). Worked examples essentially include a problem statement, a step-by-step account of the procedure to solve the problem, and the final solution. Worked examples have traditionally been applied to learn to solve well-structured problems that have a straightforward algorithmic solution process. Research has shown that studying a series of worked examples, either as preparation to or substitute of problem-solving practice, is more effective than conventional, unsupported problem solving (see, for a review, Atkinson, Derry, Renkl, & Wortham, 2000; Sweller, Ayres, & Kalyuga, 2011).

However, the effectiveness of problem-solving support methods does not necessarily generalize to inquiry learning tasks. Inquiry and modelling are recursive processes in which the scientific reasoning skills of hypothesizing, experimenting, and evidence evaluation are performed repeatedly. The nature of the hypotheses, the way they are examined, and the outcomes of these investigations all determine what would be the next logical step to induce and model the characteristics of the topic at hand (Klahr & Dunbar, 1988; White, Shimoda, & Frederiksen, 1999). Capturing this complex cognitive activity in a fixed, algorithmic sequence of action steps would neither be possible nor do justice to the true nature of the inquiry and modelling process, and presumably cause students to develop a limited understanding of the task content.

Hilbert and colleagues acknowledged this limitation of traditional worked examples, and proposed a variant that can be applied in non-algorithmic problem-solving situations (Hilbert & Renkl, 2009; Hilbert, Renkl, Kessler, & Reiss, 2008). These so-called heuristic worked examples do not emphasize the specific action sequence students should follow to solve a problem, but display the heuristic reasoning underlying the choice and application of this action sequence. This shift in focus has broadened the application of worked examples from well-structured, algorithmic problem-solving tasks to more ill-structured, and hence complex learning tasks. Recent reviews of worked-examples research have demonstrated that heuristic worked examples can effectively be applied in a variety of domains such as mathematical proving, concept mapping, and second language learning (Renkl, Hilbert, & Schworm, 2009; Sweller et al., 2011).

Heuristic worked examples also hold promise to support students' inquiry and modelling activities. Both processes are iterative by nature and require students to consider previously performed activities and their results to decide which actions to perform next. These decisions were found to be problematic because students have an insufficient understanding of the inquiry and modelling process (Mulder et al., 2011). Heuristic worked examples could help alleviate this problem by exemplifying how students can move in iterative cycles from hypothesis generation through experimentation to evidence evaluation within each model progression phase.

Research design

The purpose of the present study was to establish the instructional efficacy of heuristic worked examples in an inquiry learning environment with modelling facilities. This study utilized a between-subject design with two conditions. The learning environment in both conditions contained model order progression so that students had to build increasingly more specific models. In the experimental condition, heuristic worked examples were available for each of the three model progression phases. For each phase, the worked examples demonstrated the heuristic strategies students should apply to choose and perform their actions. Students in the control condition received no such support.

The following research hypotheses were investigated:

1. Students who are supported with heuristic worked examples exhibit a more appropriate sequence of learning activities than students who are not supported with worked examples.
2. Students who are supported with heuristic worked examples perform better (i.e., create better models) than students who are not supported with heuristic worked examples.
3. Students who are supported with heuristic worked examples learn more than students who are not supported with heuristic worked examples.

Method

Participants

Participants were 82 Dutch high school students from a science track, aged 15-17. A review of school curricula showed that the charging of capacitors, which was the topic of inquiry, was not yet taught in these students' physics classes. The students' teachers confirmed that this was the case. A prior knowledge test (see the section on knowledge tests) was administered to substantiate that participants were indeed domain novices; class-ranked prior knowledge test scores were used to assign students to either the worked example condition (n = 46) or the control condition (n = 36).

Materials

Inquiry task and learning environment

All participants worked on an inquiry task about the charging of a capacitor in an electrical circuit. Their assignment was to examine and model the influence and interactions of each element in the electrical circuit presented in a simulation. Participants performed this task within a modified stand-alone version of the Co-Lab learning environment (van Joolingen, de Jong, Lazonder, Savelsbergh, & Manlove, 2005).

The learning environment housed a simulation tool containing an electrical circuit, a voltage source, two light bulbs, and a capacitor (see Figure 5.1, left pane). Participants could experiment with this simulation to find out how these

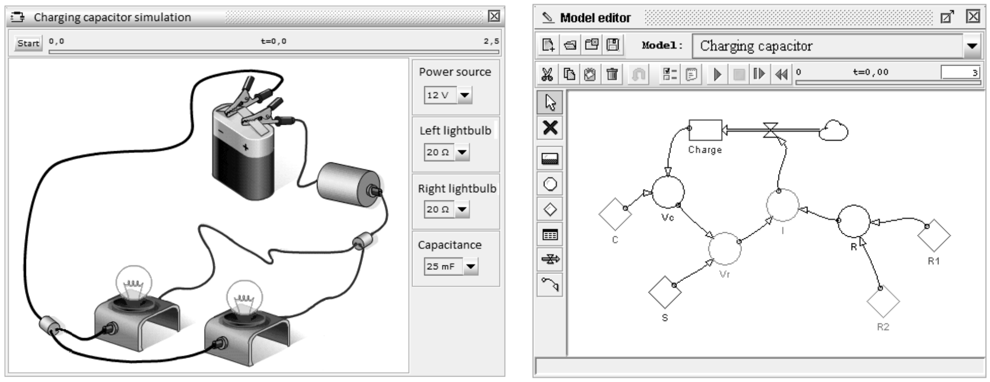

Figure 5.1. Screen capture of simulation tool (left pane) and the model editor tool with the reference model (right pane).

components behave. When they had unravelled the model underlying the simulation, participants could use the model editor tool to represent their acquired understanding in an runnable system dynamics model. As shown in Figure 5.1 (right pane), these models have a graphical structure that consists of variables and relations. Variables are the constituent elements of a model and relations define how two or more variables interact. Students could experiment with both the simulation and the model editor. The output of these experiments could be analysed through a bar chart, table, or graph tool.

An embedded help file tool contained the assignment and offered explanations of the operation of the tools in the learning environment. The help files contained no domain information on electrical circuits and capacitors.

Model order progression was implemented by dividing the modelling task into three subsequent phases. In Phase 1, students had to indicate the model elements (variables) and which ones affected which others (relationships) – but not *how* they affected them. In Phase 2, students had to provide a qualitative specification of each relationship (e.g., if resistance increases, then current decreases). In Phase 3, students had to specify each relationship quantitatively in the form of an equation (e.g., I = V / R). A more elaborate description of the model is given in Chapter 2.

Worked examples

Heuristic worked examples (hereafter: worked examples) were designed for each model progression phase. Research on the presentation format of worked examples advocates presenting examples as an action sequence over time to foster student learning (Lewis & Barron, 2009; Lusk & Atkinson, 2007). The worked examples in the present study therefore came in the form of an annotated streaming video that contained a dynamic screen capture of an anonymous person performing an inquiry and modelling task. This task was situated in a different, yet familiar context: the inflow and outflow of money. This context was chosen because it was known to the students (it was also used in the introductory session to familiarize them with the learning environment), and familiarity with the exemplifying domain was found pivotal to skill acquisition from worked examples (Renkl et al., 2009).

Seven worked examples were created: one general introductory example to reacquaint students with the exemplifying domain, and two specific examples for each model progression phase. The latter examples demonstrated the heuristic strategies students should apply to cycle effectively through the processes of hypothesis generation, experimentation, and evidence evaluation. In each model progression phase, one worked example displayed these strategies for the students' inquiry activities with the simulation; a second example concerned the use of these strategies during modelling. Both worked examples together showed how to coordinate simulation, model, and data-inspection activities.

More specifically, the simulation worked example for Phase 1 demonstrated how students could experiment with the simulation to identify relevant variables and find out which variables are related. The modelling example for this phase build on this information by demonstrating how variables can be created and linked in a model sketch, and how feedback on this sketch leads to new simulation experiments, new data, and refinements to the model. Likewise, the simulation example in Phase 2 displayed how students could induce the nature of the relationship in their model from simulation experiments; the modelling example explained how the newly-discovered relationships are incorporated in a qualitative model. In Phase 3, the simulation example demonstrated the reasoning involved in inferring physics equations from simulation data, and the modelling example displayed how these equations can be included and tested in a quantitative model.

The worked examples were presented on a website that was available only to students in the experimental condition. All seven worked examples were accessible during the entire experimental session, regardless of the model progression phase a student was in. The names of the worked examples reflected their content (e.g., "Phase 1, simulation") so as to indicate to students which worked example was relevant to them at that moment. Participants' interaction with the website's movie player that showed the worked example videos (e.g., pressing the play and stop button) were stored in a log file.

Knowledge tests

Two tests were used to assess participants' knowledge of electrical circuits: a prior knowledge test and a posttest that contained 8, respectively 14 items.

In the prior knowledge test, four open-ended questions addressed the meaning of key domain concepts (i.e., voltage source, resistance, capacitor, and capacitance), and four open-ended questions addressed the physics equations that govern the behaviour of the charging of a capacitor in an electrical circuit (i.e., Ohms law, Kirchoff's law (including its two rules: the junction rule and the loop rule), and the behaviour of capacitors). As performance on the prior knowledge test was expected to be low, three simple filler items on the interpretation of numerical data were added to sustain students' motivation during the test. These filler items were left out of the analysis. Participants' answers to the questions were scored using a rubric that allocated one point to each correct response. The Cohen's κ inter-rater reliability of this rubric was assessed by Mulder et al. (2011) and reached .89.

The posttest aimed to assess learning outcomes, and differed from the prior knowledge test in two respects. First, all eight items from the prior knowledge test were maintained, but rephrased in modelling terms in order to establish maximum resemblance with the learning task. Second, to ensure that the posttest covered the contents of all three model progression phases, six multiple-choice items were added to gauge students' qualitative understanding of the task. A rubric was developed to score participants' answers to the 14 posttest questions, and one point was allocated to each correct response. Two raters used this rubric to score the open-ended questions of a randomly selected set of 20 students: the inter-rater reliability was .96 (Cohen's κ).

Procedure

All participants engaged in three sessions that were scheduled within one week. However, due to organizational difficulties, one class (15 students) had a three-week break between the first and second session. Their second session was therefore preceded by a 10-minute (extra) recapitulation of the first session's activities. As student allocation to experimental conditions occurred within each class, this should not have influenced the study's results.

During the introductory session, participants first filled out the prior knowledge test, then received a guided tour of the Co-Lab learning environment, and finally completed a brief tutorial that familiarized them with the system dynamics modelling language and the operation of the modelling tool.

The second session started with a brief reminder that the students would work in a learning environment where the assignment was split into phases. The students were told that they could progress through these phases at their own pace, but could not return to a previous phase. They were encouraged to progress through all three phases. Furthermore, students in the experimental condition were instructed to access the website where they could watch the worked example videos. After the instructions, students worked on the assignment for approximately 90 minutes. They could stop ahead of time if they had completed their assignment.

In the final session students filled out the posttest. Students were not told in advance that they were to take this test so as to increase the likelihood that the test scores would represent the knowledge gained during the experiment.

Coding and scoring

Variables under investigation were time on task, learning paths, learning activities, performance success, and learning outcomes. The first three measures were assessed from the log files. Time on task concerned the duration of the second session; learning paths were indicated by the students' advancement through the three model progression phases. To gain insight into students' learning activities, students' use of tools in the learning environment was examined. All students could engage in activities with the simulation, the model editor, the data inspection tools (i.e., bar chart, table, and graph), and the help file tools. Students in the experimental condition had the additional option of viewing the worked example videos. As each tool was represented in a separate window, the frequency and duration of these learning activities was assessed from the log files. A specific inquiry activity that was assessed was students' experimenting behaviour. This was defined by the number of times participants clicked the "Start" button in the simulation (simulation experiment) or model editor (model experiment).

Performance success scores were assessed from the participants' final models. Both a model content and a model structure score was calculated. The model content score represented participants' understanding of the four physics equations that define the behaviour of a charging capacitor (i.e., Ohms law, Kirchoff's law (including its two rules: the junction rule and the loop rule), and the behaviour of capacitors). One point was awarded for each correctly specified equation, leading to a four-point maximum score. A prior study (Mulder et al., 2010) found a 1.0 inter-rater reliability (Cohen's κ).

The model structure score was computed in accordance with Manlove, Lazonder, and de Jong's (2006) model coding rubric. This score represents the number of correctly specified variables and relations in the models. "Correct" was judged from the reference model (see Figure 5.1). One point was awarded for each correctly named variable; an additional point was given if that variable is of the correct type. Concerning relations, one point was awarded for each correct link between two variables, and one point was awarded for the direction of the effect. The maximum model structure score was 38. In a prior study (Mulder et al., 2010) the rubric's inter-rater reliability for variables (Cohen's κ = .74) and relations (Cohen's κ = .92) were assessed, and considered to be sufficient.

Learning outcomes were indicated by students' scores on the posttest; the maximum score was 14 points.

Results

Table 5.1 summarizes the descriptive statistics for participants' performance by condition. Univariate analysis of variance (ANOVA) revealed no significant differences in prior knowledge between the conditions, $F(1, 80) = 0.03$, $p = .866$.

Most students from the experimental condition ($n = 40$) viewed at least one worked example. These students started a worked-example video 10 times on average ($SD = 6.42$) and watched them for a total of 13 minutes ($SD = 12$). One-sample t-test

Table 5.1
Summary of participants' performance

	Worked examples ($n = 46$)		Control ($n = 36$)	
	M	*SD*	*M*	*SD*
Prior knowledge test score[a]	1.48	1.28	1.53	1.36
Posttest score[b,c]	2.84	2.02	2.97	1.70
Time on task (min.)	80.48	9.85	78.25	14.21
Performance success				
Model content score[d]	0.00	0.00	0.00	0.00
Model structure score–variables[e]	10.46	3.88	7.39	2.98
Model structure score–relations[f]	8.17	4.94	4.89	4.62
Learning activities				
Number of simulation experiments	26.11	17.56	14.39	10.39
Number of model experiments	67.09	33.91	60.06	48.51

[a] Maximum score = 8. [b] Maximum score = 14. [c] Three students from the worked examples condition and two from the control condition were absent during the posttest. [e] Maximum score = 18. [f] Maximum score = 20.

showed that the mean number of worked example views differed significantly from zero, $t(39) = 9.96$, $p < .001$, which confirmed the difference in treatment across the two conditions. Despite these additional efforts, the overall time on task was comparable in both conditions, $F(1, 80) = 0.70$, $p = .404$.

Inspection of students' learning paths showed that 72% of the worked example students progressed from the first to the second phase ($n = 33$) and 30% went on to the third phase ($n = 14$). In the control condition, 56% of the students progressed from the first to the second phase ($n = 20$) and 42% reached the third phase ($n = 15$). Binomial tests revealed that the percentage of students who entered Phase 2 was significantly higher in the worked example condition, $z = 2.06$, $p = .037$, whereas the percentage of students who continued to Phase 3 was higher in the control condition, $z = -4.12$, $p < .001$.

Table 5.2 shows the frequency and duration of the learning activities in both conditions. As these are inter-dependent measures, separate univariate analyses were performed. To control for the overall Type I error rate increase with multiple significance tests, a Bonferroni correction ($\alpha = .01$) was applied. The ANOVAs indicated that students from the worked example condition engaged more often in simulation activities, $F(1, 80) = 10.72$, $p = .002$, and data inspection activities, $F(1, 80) = 8.49$, $p = .005$. Both conditions did not differ in the number of model activities, $F(1, 80) = 4.45$, $p = .038$, nor in the number of times they consulted the help files, $F(1, 80) = 1.21$, $p = .274$.

Table 5.2
Mean frequency of and percentage of time spent on learning activities

	Worked example ($n = 46$)		Control ($n = 36$)	
	M	*SD*	*M*	*SD*
Frequency				
Simulation	24.20	15.27	14.53	10.13
Model editor	64.54	29.60	50.17	31.90
Data-inspection	61.85	31.22	41.50	31.61
Help file	2.91	2.91	4.06	6.23
Relative time spent (%)				
Simulation	10.79	6.14	8.77	5.21
Model editor	58.83	11.16	75.40	7.89
Data-inspection	18.36	8.22	13.34	6.13
Help file	2.03	2.28	2.49	3.25

Furthermore, differences between conditions were found on the relative time students spent on these inquiry and modelling activities. As these too are inter-dependent measures, separate univariate ANOVAs with a Bonferroni correction ($\alpha = .01$) were performed. Results indicated that the worked example students spent relatively more time on data inspection, $F(1, 80) = 9.37$, $p = .003$, whereas the control condition students spent more time with the model, $F(1, 80) = 57.00$, $p < .001$. No statistical differences were found for simulation activities, $F(1, 80) = 2.50$, $p = .118$, and help file seeking activities, $F(1, 80) = 0.58$, $p = .450$.

While using the simulation and model editor tool, students could perform experiments to test their hypotheses. Table 5.1 shows the number of simulation and model experiments students performed. Using Pillai's trace, MANOVA produced a significant effect for condition on the number of simulation and model experiments, $V = 0.14$, $F(2, 79) = 6.25$, $p = .003$. Subsequent univariate ANOVAs revealed that students in the worked example condition performed significantly more simulation experiments, $F(1, 80) = 12.57$, $p = .001$, but as many model experiments, $F(1, 80) = 0.60$, $p = .443$, as students from the control condition.

As the worked examples showed how to integrate simulation, model, and data-inspection activities, it is also interesting to examine successions of learning activities. Figure 5.2 depicts students' navigation among the simulation, the model, data-inspection, and the help seeking tools (i.e., help files and worked examples). It can be seen that the students who received worked example support navigated among the activities differently, as was confirmed by MANOVA, using Pillai's trace, $V = 0.57$, $F(6, 75) = 11.03$, $p < .001$. (As the control condition had no access to

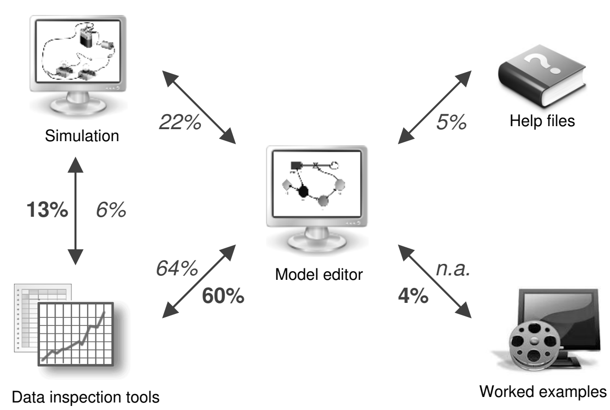

Figure 5.2. Navigation patterns among the tools in the learning environment. Data for the worked example condition appears in boldface; data for the control condition in italics. To enhance readability, only transition lines with a frequency of over 2 % are depicted.

the worked examples, this option was excluded from this analysis). Subsequent univariate ANOVAs revealed that worked example students navigated more often among the data-inspection and simulation tools, $F(1, 80) = 13.68$, $p < .001$, whereas the control students navigated more often among the help files and model editor tools, $F(1, 80) = 6.78$, $p = .011$ and between the model editor and simulation tools, $F(1 ,80) = 5.83$, $p = .018$. No differences were found for navigation between the data-inspection and help file tools, $F(1, 80) = 2.48$, $p = .119$, the help files and simulation tools, $F(1, 80) = 1.27$, $p = .263$, and the data-inspection and model editor tools, $F(1, 80) = 0.74$, $p = .392$.

Analysis of students' models suggests that the worked examples enhanced students' performance success. Using Pillai's trace, MANOVA showed a significant effect for condition on the variable and relations aspect of the model structure score, $V = 162$, $F(2, 79) = 7.65$, $p = .001$. Subsequent univariate ANOVAs revealed significant worked example effects on both the variables, $F(1, 80) = 15.38$, $p < .001$, and the relations aspect, $F(1, 80) = 9.45$, $p = .003$. The model content scores displayed in Table 5.1 implicate that none of the participants reached a correct, quantitative understanding of the physics equations. As there was no variation in scores, the model content measure was not analysed further.

The posttest was used to establish learning outcomes. Univariate ANOVA on the mean posttest scores from Table 5.1 revealed no significant difference between the two conditions, $F(1, 75) = 0.10$, $p = .759$.

Discussion

The present study addressed the effectiveness of worked examples in an inquiry learning environment with modelling facilities. The study compared the learning activities, performance, and learning outcomes among students who either were or were not supported by worked examples that provided an explicit account of what the activities in each model progression phase entail, and how they should be performed. The worked examples were expected to make the students' inquiry and modelling activities more effective and efficient, and hence lead to higher performance success and learning outcomes.

In general, results showed a positive effect of worked examples. The worked example students spent a substantial amount of their time on task viewing the worked examples, suggesting that students appreciated the additional instruction on how to coordinate and perform the inquiry and modelling activities. As predicted by the first research hypothesis, these instructions influenced the students' learning activities, as was indicated by the frequency, sequence, and duration of tool use.

Students could acquire an initial knowledge base by experimenting with the simulation tool. The worked example students used this tool more efficiently in that they conducted more experiments in the same amount of time. Furthermore, in order for experiments to lead to knowledge acquisition, it is imperative that data from these experiments is inspected in order to reach conclusions. This activity sequence was emphasized by and illustrated in the worked examples, and observed more often in the experimental condition. Compared to the control students, the worked example students more often navigated between the simulation and data-inspection tools, and spent more of their time on the data-inspection tools.

The newly acquired understanding could be tested by experimenting with the model editor tool. The worked example students used this tool more efficiently. They visited the model editor tool as often as the control students, conducted the same number of model experiments, but needed relatively less time for these experiments. The navigation patterns further show a difference in help seeking behaviour. The control students often navigated between the help-files and the model, whereas the worked example students sought help from the worked example videos. Additionally, compared to the worked example students, the control students more often navigated between the model editor and simulation. Whether this navigation pattern is sensible, depends on the direction of the navigation. Going from the model editor to the simulation can be helpful, for instance when students have finished a part of their model or experience a knowledge gap. However, navigating directly from the simulation to the model editor tool makes little sense because students would then skip the data inspection part where they can reach conclusions about their experiments and thus develop an understanding of the phenomenon.

Despite these differences in learning activities, the effects of worked examples on students' performance success and learning outcomes were less straightforward. The worked example students performed better during the task as they more accurately identified the relevant elements and their relations in the model. This confirmed the second research hypothesis. However, none of the participants acquired a complete understanding of any of the four formulas that governed the behaviour of the charging capacitor, nor were there cross-conditional differences on the posttest. This suggests that the worked examples only enhanced students' learning activities and performance but did not lead to higher learning outcomes. The procedure of this study could have brought this about, as the students were unaware that their knowledge of the task would be tested afterwards. Possibly the students only focused on performing the task well instead of learning about the phenomena. This could be prevented in future research by announcing to students that their topical knowledge will be assessed after the inquiry and modelling activity. Alternatively, learning outcomes might have been less positive due to the design of the worked examples. The worked example students spent approximately 10 percent of their time viewing the worked examples in which inquiry and modelling behaviour was demonstrated in different domain. Therefore the worked example students –compared to the control students– spent less time on the domain of the learning task from which their learning outcomes were evaluated.

Performance success and learning outcomes in the present study were quite modest. Superficial processing of the worked examples could explain this result. It could be that students merely used the worked examples to find out what was expected of them (i.e., what activities to perform), rather than to understand the rationale of these activities. Supplementing traditional worked examples with self-explanation prompts has been found to encourage learners to identify the underlying principles. Future research should investigate whether this facilitative effect generalizes to heuristic worked examples.

Alternatively, these results might be explained by the slow advancement through model progression phases. The posttest addressed the contents of all three phases, but (too) many students stayed in the first phase, and few reached the third phase. This means that relatively many students were tested on subject matter they had not been able to investigate during the session. It could therefore be that the low performance and learning outcomes are at least in part due to time constraints during the task. Similar problems arose during previous studies in which students were not supported by worked examples (Mulder et al., 2011; Mulder, Lazonder, & de Jong, 2012). It thus seems that the current worked examples did not help students progress through all three phases.

Future research should examine what is needed for students to advance through all model progression phases. The present findings suggest that students could either be given more time on task, or more appropriate support. Research on the latter option could go in several directions. One possibility is to replace worked examples by different, more fruitful forms of support. Scaffolding frameworks (e.g., Quintana et al., 2004) provide a good starting point for this line of research. A second option would be to improve the application of worked examples, for instance by using different example types (e.g., completion problems, process worked examples, partial solutions), or by supplementing worked examples with self-explanation prompts. A third possibility is to optimize the design of the heuristic worked examples. Prior work has extensively investigated the design of traditional worked examples, but the design of heuristic worked examples is relatively unexplored. Design principles from traditional worked examples (e.g., adding self-explanation prompts) do not necessarily apply to heuristic worked examples (Renkl et al., 2009). One issue in particular is that prior research leaves it somewhat unclear what should be the exemplifying domain of the heuristic worked examples. In some studies the examples were situated in more-or-less the same domain as the actual task (e.g. Hilbert & Renkl, 2009, experiment 2; Hilbert et al., 2008) whereas other studies used different contents for the heuristic examples and actual task (e.g. Hilbert & Renkl, 2009, experiment 1). As domain knowledge and inquiry skills are mutually dependent (e.g., Klahr & Dunbar, 1988), the effectiveness of the heuristic examples might depend on the subject matter it contains.

To conclude, the present study does not allow for a definitive conclusion on the added value of heuristic worked examples to support students on an inquiry task combined with modelling. Even though heuristic worked examples were found to enhance learning activities and performance success, they did not affect learning outcomes. As with any novel application of learning support, continued iterative rounds of design and evaluation are needed to discover its true potentials.

References

Alessi, S. (1995). *Dynamic versus static fidelity in a procedural simulation*. Paper presented at the Annual meeting of the American Educational Research Association, San Francisco.

Alfieri, L., Brooks, P. J., Aldrich, N. J., & Tenenbaum, H. R. (2011). Does discovery-based instruction enhance learning? *Journal of Educational Psychology, 103*, 1-18. doi: 10.1037/a0021017

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research, 70*, 181-214. doi: 10.3102/00346543070002181

de Jong, T., Martin, E., Zamarro, J. M., Esquembre, F., Swaak, J., & van Joolingen, W. R. (1999). The integration of computer simulation and learning support: An example from the physics domain of collisions. *Journal of Research in Science Teaching, 36*, 597-615. doi: 10.1002/(SICI)1098-2736(199905)36:5<597::AID-TEA6>3.0.CO;2-6

de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*, 179-201. doi: 10.3102/00346543068002179

Eseryel, D., & Law, V. (2010). *Promoting learning in complex systems: Effect of question prompts versus system dynamics model progressions as a cognivie-regulation scaffold in a simulation-based inquiry-learning envornment*. Paper presented at the 9th International Conferences of the Learning Sciences, Chicago, IL.

Hilbert, T. S., & Renkl, A. (2009). Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps. *Computers in Human Behavior, 25*, 267-274. doi: 10.1016/j.chb.2008.12.006

Hilbert, T. S., Renkl, A., Kessler, S., & Reiss, K. (2008). Learning to prove in geometry: Learning from heuristic examples and how it can be supported. *Learning and Instruction, 18*, 54-65. doi: 10.1016/j.learninstruc.2006.10.008

Hogan, K., & Thomas, D. (2001). Cognitive comparisons of students' systems modeling in ecology. *Journal of Science Education and Technology, 10*, 319-345. doi: 10.1023/A:1012243102249

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12*, 1-48. doi: 10.1207/s15516709cog1201_1

Lewis, D., & Barron, A. (2009). *Animated demonstrations: Evidence of improved performance efficiency and the worked example effect*. Paper presented at the 1st International Conference on Human Centered Design, San Diego, CA.

Lusk, M. M., & Atkinson, R. K. (2007). Animated pedagogical agents: does their degree of embodiment impact learning from static or animated worked examples? *Applied Cognitive Psychology, 21*, 747-764. doi: 10.1002/acp.1347

Minner, D. D., Levy, A. J., & Century, J. (2010). Inquiry-based science instruction- what is it and soes it matter? Results from a research synthesis years 1984 to 2002. *Journal of Research in Science Teaching, 47*, 474-496. doi: 10.1002/tea.20347

Mulder, Y. G., Lazonder, A. W., & de Jong, T. (2010). Finding out how they find it out: An empirical analysis of inquiry learners' need for support. *International Journal of Science Education, 32*, 2033-2053. doi: 10.1080/09500690903289993

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. doi: 10.1016/j.learninstruc.2011.01.003

Mulder, Y. G., Lazonder, A. W., de Jong, T., Anjewierden, A., & Bollen, L. (2012). Validating and optimizing the effects of model progression in simulation-based inquiry learning. *Journal of Science Education and Technology*. Advance online publication. doi: 10.1007/s10956-011-9360-x

Paas, F. G. W. C., & van Merriënboer, J. J. G. (1994). Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive load approach. *Journal of Educational Psychology, 86*, 122-133.

Quinn, J., & Alessi, S. (1994). The effects of simulation complexity and hypothesis-generation strategy on learning. *Journal of Research on Computing in Education, 27*, 75-91.

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

Renkl, A., Hilbert, T., & Schworm, S. (2009). Example-based learning in heuristic domains: a cognitive load theory account. *Educational Psychology Review, 21*, 67-78. doi: 10.1007/s10648-008-9093-4

Rieber, L. P., & Parmley, M. W. (1995). To teach or not to teach? Comparing the use of computer-based simulations in deductive versus inductive approaches to learning with adults in science. *Journal of Educational Computing Research, 13*, 359-374. doi: 10.2190/M8VX-68BC-1TU2-B6DV

Stratford, S. J., Krajcik, J., & Soloway, E. (1998). Secondary students' dynamic modeling processes: Analyzing, reasoning about, synthesizing, and testing models of stream ecosystems. *Journal of Science Education and Technology, 7*, 215. doi: 10.1023/A:1021840407112

Swaak, J., van Joolingen, W. R., & de Jong, T. (1998). Supporting simulation-based learning; the effects of model progression and assignments on definitional and intuitive knowledge. *Learning and Instruction, 8*, 235-252. doi: 10.1016/s0959-4752(98)00018-8

Sweller, J., Ayres, P., & Kalyuga, S. (2011). *Cognitive load theory: Explorations in the learning sciences, instructional systems and performance technologies*. New York, NY: Springer.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2*, 59-89.

van Joolingen, W. R., de Jong, T., Lazonder, A. W., Savelsbergh, E. R., & Manlove, S. (2005). Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning. *Computers in Human Behavior, 21*, 671-688. doi: 10.1016/j.chb.2004.10.039

White, B. Y., & Frederiksen, J. R. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 42*, 99-157. doi: 10.1016/0004-3702(90)90095-h

White, B. Y., Shimoda, T. A., & Frederiksen, J. R. (1999). Enabling students to construct theories of collaborative inquiry and reflective learning: Computer support for metacognitive development. *International Journal of Artificial Intelligence in Education, 10*, 151-182.

Chapter 6

Summary and general discussion

Introduction

Inquiry learning environments based on simulations essentially enable students to learn science by doing science: they offer interactive resources so students can develop a deep understanding of a domain by engaging in scientific reasoning processes such as hypothesis generation, experiment design, and evaluating evidence. Inquiry learning environments also increasingly incorporate modelling facilities for students to articulate their research hypotheses and (acquired) domain knowledge. These technological advancements have led to the development of the integrated approach to science learning that was pivotal to the research in this thesis. This approach, in short, entails that students without prior domain knowledge first carry out some exploratory experiments with a simulation to gain an initial understanding of the phenomena under investigation. Students who have some prior knowledge can skip this step and immediately start sketching a model outline to express their (initial/acquired) understanding of the phenomena. Subsequently, students form hypotheses which they can explore with the simulation in order to transform their model sketch into a runnable model by specifying the relations between the variables in the model. During data interpretation, learners compare their model to data from the simulation, which during the conclusion phase, feeds their decisions to revise the model.

As the orchestration of these inquiry and modelling activities is both new and demanding to students, the general research question of this thesis was:

How can learning with computer simulations and models be improved by embedded support?

This general research question was addressed in the four empirical studies depicted in Chapters 2 through 5. The first empirical study concerned an assessment of students' need for support by comparing the inquiry and modelling activities of junior high school students (aged 14-15), senior high school students (aged 18-20), and university students (aged 20-27). The second and third study investigated whether model progression (i.e., gradually increasing task complexity) could help compensate for the skill deficiencies observed in the first study. The final study explored whether complementing model progression with worked examples would further enhance students' inquiry and modelling performance and learning. The three intervention studies involved 15 to 17-year old high school students from the science track. All participants worked on an inquiry learning task in the Co-Lab learning environment (van Joolingen, de Jong, Lazonder, Savelsbergh, & Manlove, 2005) which combines learning from simulations with learning by modelling. The topic of inquiry was a charging capacitor, which students had to investigate through a simulation and then create a computer model of its behaviour. In the present chapter, the main findings from these studies are summarized and discussed.

Empirical studies

Study 1: Assessment of students' support needs

The first study, described in Chapter 2, attempted to reveal students' need for support during learning with computer simulations and models. This study sought to gain insight into students' scientific reasoning skill deficiencies by contrasting domain novices' inquiry behaviour and performance to that of a considerably more knowledgeable reference group and an intermediate reference group.

Klahr and Dunbar's (1988) SDDS model was used as analytical framework to predict why and how students' levels of domain knowledge influence their inquiry learning skills and outcomes. Three inquiry skills are central to this model: hypothesis generation, experimentation, and evaluating evidence. SDDS assumes that students will perform these skills more effectively and efficiently to the extent that they possess more domain knowledge. Such superior skills, in turn, are assumed to yield higher knowledge gains. In view of this reciprocal influence, it was predicted that high prior knowledge students would generate more -and more specific- hypotheses, and conduct fewer experiments to build a fully correct model than students with intermediate prior knowledge. The same differences were expected to occur when comparing intermediate and low prior knowledge students. Evaluating evidence, an often neglected inquiry skill, was studied in an exploratory fashion.

Thirty-one students volunteered to participate in this study. Participants were selected for their levels of prior domain knowledge and classified as either low-level novice (10 junior high school students without prior knowledge), high-level novice (10 senior high school students from the science track with some prior knowledge), or expert (11 university students in electrical engineering). All participants performed an unguided physics task within Co-Lab.

Results showed that experts needed less time to complete the task than high-level novices, who needed as much time as low-level novices. The quality of the students' models also differed in favour of the experts, whereas high-level novices built better models than low-level novices. Qualitative analyses of students' models revealed that even low-level novices had a pretty good sense of which elements to include in their models but largely failed to identify the relationships between the model elements.

Only few between-group differences were found regarding participants' learning behaviour. The three groups generated equally specific hypotheses, conducted as many experiments overall, and had an equal percentage of exploratory experiments. Surprisingly, few low-level novices started off by experimenting with the simulation to gather domain knowledge. Concerning modelling, experts evidenced more model runs than both groups of novices. Analysis of students' evidence evaluation showed that low-level novices rejected hypotheses more often than high-level novices and experts —which seems fitting as novices are likely to have more incorrect hypotheses. Accepting and modifying hypotheses occurred as often in all three groups. However, the hypotheses about relations that novices generated and tested were so specific (i.e., only quantitative in nature) that they were most likely based on guesswork rather than students' inquiry or modelling activities. Analysis of the reasoning behind model modifications bore this out: while experts and high-level novices typically motivated their changes, only few of the low-level novices' changes were guided by reasoning.

These findings point to a clear need for support. Low-level novices' model quality scores were very low, indicating that they acquired almost no knowledge from their inquiry. This seems due to the fact that these students generally exhibited the same inquiry behaviour as their more knowledgeable counterparts. However, without prior knowledge (and instructional support) this expert-like behaviour is probably less appropriate and certainly less effective. Support for inquiry learning should therefore try to better attune students' inquiry activities to their level of domain knowledge, or provide domain support in order to increase the effectiveness of their natural inquiry behaviour. From these findings it was concluded that it might be fruitful to restrain domain novices' natural tendency to engage in quantitative modelling from scratch by first having them create models that are qualitatively specified, and then enabling them to transfer these qualitative relations into quantitative ones.

Study 2: Model progression

Given the results of the previous study, the second study (described in Chapter 3) investigated the effects of model progression. Model progression (i.e., gradually introducing a tasks' complexity), was expected to enhance performance success as it better attunes students' inquiry behaviour to their level of domain knowledge. Following White and Frederiksen (1990), two types of model progression were distinguished. Model order progression (MOP) gradually increases the specificity of the relations between variables, whereas model elaboration progression (MEP) gradually expands the number of variables in the task. As MOP was intended to scaffold learners' relation construction –a key problem to domain novices– it was hypothesized that MOP would be the better means to support students.

Based on class-ranked pretest scores, participants were assigned to three conditions (MOP, n= 28; MEP, n =26; control condition, n = 30). All participants performed an inquiry learning physics task within Co-Lab, an inquiry learning environment in which participants could experiment through a computer simulation, and express hypotheses and (acquired) understanding in a system dynamics model.

All conditions used the same instructional content (i.e., charging capacitors), but differed with regard to the support mechanisms. Participants in the control condition worked with the standard configuration of the environment and thus received no support. As such, the specifics of this condition were comparable to those of the low-level novices in the first study. Participants in the model order progression (MOP) condition received a full-complex simulation and were asked to induce and build increasingly specific models. The initial phase only dealt with the model structure: students were asked to identify and sketch the model with its variables and relations. The consecutive phases deal with the model content. Students had to specify the relationships between the elements qualitatively (e.g., if resistance increases, then current decreases) in Phase 2 and quantitatively (e.g., I=V/R) in Phase 3. In the model elaboration progression (MEP) condition, the complexity of the simulation was gradually increased by adding components to the electrical circuit. Over phases, participants had to extend their model to incorporate the new elements presented in the simulation and their associated knowledge components.

Model progression in general was found to lead to higher performance success, and participants in the MOP condition outperformed those from the MEP condition. This result supports the hypothesis that model order progression is more in keeping with domain novices' learning needs. However, observed learning gains, although statistically significant, were quite modest as even the MOP-students obtained merely one third of the maximum score on average. One reason could be that few MOP students completed all three phases of the task sequence. Analysis of these students' learning activities and models revealed that many students progressed from the first to the second phase, but few went on to the third phase. From these findings it was concluded that additional support is needed.

Study 3: Model progression fine-tuned

The third study (described in Chapter 4) aimed to further examine model order progression. Presumably, the model order progression students in the second study (Chapter 3) got stuck in the second phase as they entered this phase with a rather simple model. This simple model probably provided them with an insufficient basis for the complex task at hand in the second phase. Such 'premature' progressions could be avoided by prohibiting students to enter subsequent phases until sufficient understanding has been acquired. An alternative solution might be to allow students who get stuck in a particular phase to return to previous phases to remediate knowledge gaps —an option that was unavailable to students in the previous study. Both alternatives were hypothesized to increase participants' performance success and their chances of successfully progressing through all phases.

Based on class-ranked pretest scores, sixty-one participants were assigned to one of three conditions (restricted, n = 19; semi-restricted, n = 19; unrestricted, n = 22). All participants performed the same inquiry learning physics task within Co-Lab, which was supported by model order progression (MOP) as used in the previous study, but they differed with regard to the requirements to cycle through model progression phases. Participants in the restricted condition could progress to subsequent phases only if sufficient knowledge had been acquired. Participants in the semi-restricted condition received the 'standard' model progression, they could enter subsequent phases at will (comparable to the MOP condition in the second study). Participants in the unrestricted condition were free to enter both subsequent and previous phases.

The conditions for changing model progression phases was found to affect the learning process. Compared to the semi-restricted condition, the restricted students reached the second phase less often, but the students who did had significantly better models. This seems due to the fact that the restricted students were simply prohibited to enter Phase 2 with insufficient knowledge. The unrestricted students, by contrast, progressed to subsequent phases more often than the semi-restricted students. However, the quality of the unrestricted participants' intermediate models did not differ from the quality of the semi-restricted participants' intermediate models, which suggests that the unrestricted participants used their navigation freedom wisely.

Even though both model progression variants influenced the learning process, they did not enhance students' performances success. It thus seems that neither more liberal nor more strict requirements to change model progression phases are sufficient to further improve the effectiveness of model progression. It thus seems that students need more explicit support in order to better understand what the activities in each model progression phase entail, and how they should be performed.

Study 4: Model progression complemented with worked examples

The fourth study conducted (described in Chapter 5) aimed to examine the effectiveness of heuristic worked examples as additional support. Heuristic worked examples were proposed by Hilbert and colleagues (Hilbert & Renkl, 2009; Hilbert, Renkl, Kessler, & Reiss, 2008) as means to extend the application of worked examples from well-structured problem solving tasks to more ill-structured, and hence more complex tasks. Unlike 'traditional' worked examples that provide students with a single algorithm to solve one particular type of problem, heuristic worked examples outline a series of problem solving strategies and demonstrate their usage in or across a range of related tasks. The heuristic worked examples were hypothesized to influence students' learning activities' sequence and improve students' performance and learning.

Based on class-ranked pretest scores eighty-two participants were assigned to either the worked examples condition (n = 46) or the control condition (n = 36). As in the previous studies, students in both conditions had to investigate a charging capacitor and create a computer model of its behaviour. This task was divided in three phases, comparable with the MOP and semi-restricted conditions in the second and third study, respectively. Students in the worked examples condition could consult two heuristic worked examples for each phase. These examples came in the form of annotated videos that showed the inquiry or modelling activities of an anonymous person on a comparable task in a different domain. Students in the control condition did not receive these worked examples.

Main findings indicate that students in both conditions had comparable and low pretest scores, needed quite the same amount of time on task, but spent this time differently. As instructed by the worked examples, students in this condition did more experiments with the simulation and took more time to analyse and interpret the outcomes. Control students, by contrast, largely ignored the simulation and spent most of their time creating and testing their model. This proved rather ineffective, as the models of students in the worked example condition contained more correct variables and relations. Despite this performance difference, however, posttest scores were comparable across conditions, suggesting that worked example students performed better, but did not learn more.

General discussion

After having summarized the experimental studies comprised in this thesis, this concluding section discusses some cross-study issues from a bird's-eye view.

Participants in the studies reported in this thesis mainly were high school students who were largely incognizant of the topic of charging capacitors. Literature suggests that domain knowledge influences students' inquiry process and thus how much they can pick up from an inquiry learning task (Hmelo, Nagarajan, & Day, 2000; Klahr & Dunbar, 1988; Lazonder, Wilhelm, & Hagemans, 2008; Schauble, Klopfer, & Raghavan, 1991). A review of school curricula and teacher statements showed that students were (or should be) familiar with electrical circuits and concepts such as power source, resistance and Ohm's law. This knowledge is prerequisite to the topic of charging capacitors which was *not* yet taught in these students' physics classes. To confirm both assumptions, a prior knowledge test was administered that addressed both the allegedly familiar knowledge about electrical circuits as well as the new and unfamiliar knowledge about charging capacitors. Students' performance on this test (a score of 1 or 2 out of 8) indicated that students could indeed be considered domain novices. In hindsight, however, this low score also suggests that they may have lacked that necessary prerequisite knowledge, which may have negatively impacted their inquiry process. To prevent the negative influence of too low entry levels of domain knowledge in future research, the prerequisite knowledge could be recapitulated before or during the studies (Lazonder, Wilhelm, & van Lieburg, 2009). Students then are likely to benefit even more from the support they receive.

For assessing the learning process, the three intervention studies reported in Chapters 3 to 5 relied almost exclusively on logfile analysis. The logfiles generated by the Co-Lab learning environment provided data on the frequency with which students' performed experiments and how they evaluated data from these experiments. For additional information regarding students' hypotheses, qualitative information was obtained through think aloud protocols in the first study (Chapter 2). Think aloud protocols offer information about what students are doing, and why they are doing it while they are doing it (Ericsson & Simon, 1993). However, whilst think alouds can provide rich data, some participants may find it difficult to think aloud during tasks that require cognitive processing (Branch, 2000). Therefore, and because the aim of successive studies was to reveal differences in performance success, only the non-obtrusive method of logfile analysis was used in these studies, which admittedly may have limited our view on what really was going on from a cognitive perspective.

From the first study onwards, the models were evaluated using two coding rubrics that were derived from Manlove, Lazonder, and de Jong's (2006) model coding rubric. A distinction was made between the structure and content of a model, and separate scores were computed for each aspect. The model structure score indicated the number of correctly specified variables and relations in a model; the model content score represented students' understanding of the four physics equations that define the behaviour of a charging capacitor. In later studies where model order progression was applied, the knowledge as addressed by these measures was closely related to the content of the first (model sketching) and third (quantitative relations) model progression phases, leaving the qualitative knowledge of Phase 2 unaddressed. For the second study (Chapter 3), adding a qualitative component to either of the coding rubrics would have been unfair in a comparison to the other two conditions who received no model order progression support and thus did not engage in qualitative modelling. For the third and fourth study, adding a qualitative component to either of the rubrics might have led to a more complete assessment of the models. This adjustment was made in a recent re-analysis of the third study's data (Mulder, Lazonder, de Jong, Anjewierden, & Bollen, 2012). Although the qualitative component increased the scores on the whole, the ratio of the groups' scores appeared comparable to what was found in the study reported in Chapter 4.

For the studies in this thesis a software agent was developed, which was able to assess the students' models during the task (for more information on the technical aspects of the agent, see Anjewierden et al., 2012). The software agent had two major functions. First, it enabled students to go through complete hypothesis cycles in the first model order progression phase. In this phase, where students have to identify relationships between elements without specifying them, it was technically impossible for the model editor to execute these models. Model runs in this phase therefore activated the software agent which made that students could still "perform" experiments with their models. The agent presented the results of these experiments in a bar chart tool in order for students to evaluate the data of their experiments so that they could reach conclusions. Second, in the third study (Chapter 4), the software agent functioned as a gate-keeper to control the restricted students' progression over phases. The restricted students in that study were only allowed to progress to a subsequent phase if their model was of sufficient quality. For future research the software agent holds promise as it enables a more sophisticated, and possible effective approach to give adaptive support on the students' actions. A software agent can detect patterns in the students' inquiry and modelling activities, and use this information to give tailor-made assistance and feedback at times appropriate. Such techniques have been successfully applied in small-scale modelling tasks (Bravo, van Joolingen, & de Jong, 2009), and are currently being implemented in more comprehensive model-based inquiry learning environments (de Jong et al., 2010). Research and development of techniques and environments like these could pave the way to active and effective methods of science education. Questions remain, however, as to what exactly this type of adaptive feedback should look like. The fourth study (Chapter 5) suggests that students need a more explicit account of what the activities in each model progression phase entail and how they should be performed. The worked examples were generally accessible to all students in the worked example condition. Providing this information in a just-in-time fashion by a software agent might improve students' performance even more, what future research should explore.

Previous research has provided an elaborate account of scaffolds that aim to compensate for students' skill deficiencies. Examples include proposition tables to help generate hypotheses (de Jong, 2006), adaptive advice for extrapolating knowledge from simulations (Leutner, 1993), and regulative scaffolds to assist students in planning, monitoring, and evaluating their inquiry (Manlove et al., 2006). Modelling support thus far has mainly focussed on making it easier for students to transfer their ideas into a formal model, for instance by creating graphical modelling representations (Löhner, 2005) or integrating drawing facilities in the modelling process (van Joolingen, Bollen, & Leenaars, 2010). However, the empirical foundations underlying the contents of these supports often remain hidden to the public eye. In this thesis, support was developed from the insight into students' scientific reasoning skill deficiencies that was acquired in the first empirical study.

But do students who perform better also learn more? As performance success appears a prerequisite for learning, the first studies in this thesis did not address this question. Theoretical and empirical evidence suggests that the performance measures (i.e., model quality scores) that assessed the instructional effects of model progression are indicative of the knowledge students acquired during the experiment. This assumption is based on constructionism, an instructional paradigm in which learning is considered synonymous to the knowledge construction that takes place when learners are engaged in building objects (Kafai & Resnick, 1996) Research has confirmed that the construction of models is associated with cognitive learning (e.g., van Borkulo, 2009) and that the quality of students' models is associated with their reasoning processes (Sins, Savelsbergh, & van Joolingen, 2005). It thus seemed plausible to infer the instructional effects of model progression the students' task performance. However, to paint a more complete picture, a posttest measuring learning outcomes was administered in the final study reported in Chapter 5. Contrary to expectations, the favourable effects found on the performance measure did not show on the learning outcomes measure. One possible explanation is that the posttest was not sensitive enough to the students' learning during the task. The posttest was construed to cover the contents of all model progression phases. As only few students reached the third phase, relatively many students were tested on subject matter they had not been able to investigate during the task. Future research should investigate how learning outcomes can best be assessed..

Overall conclusion and practical implications

The main goal of the current research was to find an answer to the question:

How can learning with computer simulations and models be improved by embedded support?

Via an iterative series of studies it was first confirmed that unsupported students do perform poorly on an inquiry learning task combined with modelling. The study in Chapter 2 showed that integrating learning from computer simulations and learning by modelling was problematic to students. The unsupported students hardly experimented with the simulation from which they could learn about the charging of the capacitor, and failed to build a correct model. These results suggest that students might benefit from support that helps them to better attune their inquiry behaviour to their level of domain knowledge, especially regarding relation construction in the models. This support could take the form of model order progression, which enables students to gradually increase the specificity of the relations between variables in their models. Students then first have to identify and sketch the model with its variables and relations prior to specifying the model content. This model content gradually increases in complexity too: students first have to provide a qualitative specification of each relationship (e.g., if resistance increases, then current decreases), and then translate these qualitative relations into quantitative specifications (i.e., $I = V/R$).

Model order progression was indeed found to increase students' performance on an inquiry learning task combined with modelling. However, not to a satisfactory degree. In an attempt to fine-tune model order progression no effect was found for restricting students to enter subsequent phases until sufficient understanding has been acquired, nor for allowing students who get stuck in a particular phase to return to previous phases to remediate knowledge gaps. It was concluded that students need additional support in order to better understand what the activities in each model progression phase entail, and how they should be performed. This support –in the form of worked out examples– was found to significantly improve students' performance.

These positive results were found despite the students lack of prerequisite knowledge of the domain. Even though electrical circuits had been covered as part of their curriculum, students failed to reproduce this basic knowledge on the prior knowledge test. Still, a combination of model progression and worked examples offered a notable, albeit modest improvement in students' learning from simulations combined with modelling. As for practical implications, students are likely to benefit more from this learning approach as their entry level of domain knowledge is sufficient. Therefore, in actual practice, teachers should keep an eye out to detect these knowledge gaps in time. It would be advisable that teachers respond to these knowledge gaps by first recapitulating the required prior knowledge.

Alternative suggestions that have consistently been offered throughout this thesis include extending time on task and providing additional support. However, teachers may consider extra class time unfeasible or undesirable, and one might indeed wonder how much extra time should be devoted to a relatively small topic such as the charging of a capacitor. Even though inquiry learning admittedly takes more time than direct instruction, the scope of an inquiry unit should match the amount of time available in the curriculum. A more practical solution might therefore be to offer additional support. In actual practice, teachers can provide students with the relevant domain knowledge or procedural assistance during the task. As a result, students can gain from all the benefits that inquiry learning and modelling have to offer (e.g., Campbell, Zhang, & Neilson, 2011; Deslauriers & Wieman, 2011; Rutten, van Joolingen, & van der Veen, 2012; van Borkulo, 2009; van Joolingen et al., 2005), without getting stuck by the difficult challenges that this integrated approach poses. However, for teachers who wish to implement learning from simulations combined with modelling, we advise to supplement the inquiry and modelling task with model order progression and worked examples.

References

Anjewierden, A., Mulder, Y. G., Bollen, L., Lazonder, A. W., de Jong, T., & de Hoog, R. (2012). *Agent-based support for the analysis of graph-like structures created by students*. Manuscript in preparation.

Branch, J. L. (2000). Investigating the information-seeking processes of adolescents: The value of using think alouds and think afters. *Library & Information Science Research, 22*, 371-392. doi: 10.1016/s0740-8188(00)00051-7

Bravo, C., van Joolingen, W. R., & de Jong, T. (2009). Using Co-Lab to build System Dynamics models: Students' actions and on-line tutorial advice. *Computers & Education, 53*, 243-251. doi: 10.1016/j.compedu.2009.02.005

Campbell, T., Zhang, D. H., & Neilson, D. (2011). Model based inquiry in the high school physics classroom: An exploratory study of implementation and outcomes. *Journal of Science Education and Technology, 20*, 258-269. doi: 10.1007/s10956-010-9251-6

de Jong, T. (2006). Scaffolds for computer simulation based scientific discovery learning. In J. Elen & R. E. Clark (Eds.), *Handling complexity in learning environments* (pp. 107-128). Amsterdam: Elsevier.

de Jong, T., van Joolingen, W. R., Giemza, A., Girault, I., Hoppe, U., Kindermann, J., . . . van der Zanden, M. (2010). Learning by creating and exchanging objects: The SCY experience. *British Journal of Educational Technology, 41*, 909-921. doi: 10.1111/j.1467-8535.2010.01121.x

Deslauriers, L., & Wieman, C. (2011). Learning and retention of quantum concepts with different teaching methods. *Physical Review Special Topics-Physics Education Research, 7*, 010101. doi: 10.1103/PhysRevSTPER.7.010101

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data*. Londen: The MIT Press.

Hilbert, T. S., & Renkl, A. (2009). Learning how to use a computer-based concept-mapping tool: Self-explaining examples helps. *Computers in Human Behavior, 25*, 267-274. doi: 10.1016/j.chb.2008.12.006

Hilbert, T. S., Renkl, A., Kessler, S., & Reiss, K. (2008). Learning to prove in geometry: Learning from heuristic examples and how it can be supported. *Learning and Instruction, 18*, 54-65. doi: 10.1016/j.learninstruc.2006.10.008

Hmelo, C. E., Nagarajan, A., & Day, R. S. (2000). Effects of high and low prior knowledge on construction of a joint problem space. *Journal of Experimental Education, 69*, 36-56. doi: 10.1080/00220970009600648

Kafai, Y. B., & Resnick, M. (Eds.). (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world.* Mawhaw, NJ: Lawrence Erlbaum Associates.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12*, 1-48. doi: 10.1207/s15516709cog1201_1

Lazonder, A. W., Wilhelm, P., & Hagemans, M. G. (2008). The influence of domain knowledge on strategy use during simulation-based inquiry learning. *Learning and Instruction, 18*, 580-592. doi: 10.1016/j.learninstruc.2007.12.001

Lazonder, A. W., Wilhelm, P., & van Lieburg, E. (2009). Unraveling the influence of domain knowledge during simulation-based inquiry learning. *Instructional Science, 37*, 437-451. doi: 10.1007/s11251-008-9055-8

Leutner, D. (1993). Guided discovery learning with computer-based simulation games: Effects of adaptive and non-adaptive instructional support. *Learning and Instruction, 3*, 113-132. doi: 10.1016/0959-4752(93)90011-N

Löhner, S. (2005). *Computer based modeling tasks, the role of external representation.* Unpublished doctoral dissertation , University of Amsterdam, Amsterdam.

Manlove, S., Lazonder, A. W., & de Jong, T. (2006). Regulative support for collaborative scientific inquiry learning. *Journal of Computer Assisted Learning, 22*, 87-98. doi: 10.1111/j.1365-2729.2006.00162.x

Mulder, Y., Lazonder, A., de Jong, T., Anjewierden, A., & Bollen, L. (2012). Validating and optimizing the effects of model progression in simulation-based inquiry learning. *Journal of Science Education and Technology*, Advance online publication. doi: 10.1007/s10956-011-9360-x

Rutten, N., van Joolingen, W. R., & van der Veen, J. T. (2012). The learning effects of computer simulations in science education. *Computers & Education, 58*, 136-153. doi: 10.1016/j.compedu.2011.07.017

Schauble, L., Klopfer, L., & Raghavan, K. (1991). Students' transitions from an engineering to a science model of experimentation. *Journal of Research in Science Teaching, 28*, 859-882. doi: 10.1002/tea.3660280910

Sins, P. H. M., Savelsbergh, E. R., & van Joolingen, W. R. (2005). The difficult process of scientific modeling: An analysis of novices' reasoning during computer-based modeling. *International Journal of Science Education, 14*, 1695-1721. doi: 10.1080/09500690500206408

van Borkulo, S. P. (2009). *The assessment of learning outcomes of computer modelling in secondary science education*. Unpublished doctoral dissertation. University of Twente. Enschede.

van Joolingen, W. R., Bollen, L., & Leenaars, F. A. J. (2010). Using drawings in knowledge modeling and simulation for science teaching. In R. Nkambou, J. Bourdeau & R. Mizoguchi (Eds.), *Advances in intelligent tutoring systems* (Vol. 308, pp. 249-264). New York: Springer.

van Joolingen, W. R., de Jong, T., Lazonder, A. W., Savelsbergh, E. R., & Manlove, S. (2005). Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning. *Computers in Human Behavior, 21*(4), 671-688. doi: 10.1016/j.chb.2004.10.039

White, B. Y., & Frederiksen, J. R. (1990). Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence, 42*, 99-157. doi: 10.1016/0004-3702(90)90095-h

Chapter 7

Nederlandse samenvatting

Inleiding

Elektronische leeromgevingen voor onderzoekend leren bieden leerlingen de mogelijkheid om natuurkundige verschijnselen te bestuderen én zich een beeld te vormen van wat 'onderzoek doen' inhoudt. Computersimulaties vormen vaak de basis voor dit soort leeromgevingen, waarin leerlingen door systematisch te experimenteren (d.w.z. hypotheses opstellen, experimenten uitvoeren en het resultaat van deze experimenten evalueren) de eigenschappen van natuurkundige materialen en verschijnselen leren begrijpen. Steeds vaker worden er ook software tools voor modelleren toegevoegd aan deze leeromgevingen. Deze combinatie van onderzoekend leren met simulaties en modelleren ondersteunt het iteratieve onderzoekend leerproces doordat leerlingen de opgedane kennis in een model kunnen weergeven en dit model vervolgens kunnen toetsen en waar nodig aanpassen of verfijnen.

Ondanks de mogelijkheden die dergelijke elektronische leeromgevingen bieden, blijken leerlingen behoefte te hebben aan ondersteuning van het leerproces. Vandaar dat de overkoepelende onderzoeksvraag voor de in dit proefschrift beschreven studies, was:

Hoe kan onderzoekend leren met computersimulaties en modellen worden verbeterd door geïntegreerde ondersteuning?

Om deze vraag te beantwoorden zijn vier studies uitgevoerd. Tijdens deze studies werkten middelbare scholieren in de Co-Lab leeromgeving aan een onderzoekend leren-taak over het natuurkunde onderwerp 'opladen van een condensator'. In Studie 1 is geprobeerd de ondersteuningsbehoefte van leerlingen tijdens het leren met simulaties en modellen in kaart te brengen. In Studie 2 is onderzocht of modelprogressie (het in kleine stapjes complexer maken van de leertaak) in de vastgestelde ondersteuningsbehoefte voorziet. In Studie 3 zijn twee alternatieven om modelprogressie te verbeteren onderzocht. In Studie 4 is tenslotte onderzocht of de toevoeging van uitgewerkte voorbeelden het onderzoekend leren positief beïnvloedt.

Studie 1: De ondersteuningsbehoefte van leerlingen

De eerste studie, beschreven in Hoofdstuk 2, was bedoeld om inzicht te krijgen in de ondersteuningsbehoefte van leerlingen tijdens het leren met computersimulaties en modellen. In deze studie is gekeken welke onderzoeksvaardigheden de leerlingen wel en niet uit zichzelf kunnen toepassen. Dit werd gedaan door leerlingen zonder inhoudelijke kennis te vergelijken met een referentiegroep met een veel hoger kennisniveau en een tussenliggende referentie groep.

Drie onderzoeksvaardigheden stonden centraal: hypothesen opstellen, experimenteren en resultaat van deze experimenten interpreteren en beoordelen. Er werd verwacht dat leerlingen deze vaardigheden effectiever en efficiënter kunnen uitvoeren al naar gelang hun inhoudelijke kennis toeneemt. Betere onderzoeksvaardigheden zorgen op hun beurt voor een grotere kennistoename. In het licht van deze wederkerige invloed werd voorspeld dat leerlingen met een hoger kennisniveau meer (en meer specifieke) hypotheses opstellen, en minder experimenten hoeven uit te voeren om een volledig correct model te bouwen dan leerlingen van het tussenliggende kennisniveau. Vergelijkbare verschillen werden verwacht bij een vergelijking van de leerlingen met het tussenliggende kennisniveau en de leerlingen zonder voorkennis. Er waren geen vooraf opgestelde verwachtingen ten aanzien van de derde onderzoekend leren vaardigheid (d.w.z. resultaat van de experimenten interpreteren en beoordelen).

Eenendertig leerlingen deden mee aan deze studie. Deelnemers waren geselecteerd op basis van hun kennisniveau en worden aangeduid als *low-level novices* (10 leerlingen uit 3 VWO), *high-level novices* (10 leerlingen uit 6 VWO) of *experts* (11 universitaire studenten elektrotechniek). Alle deelnemers werkten aan een onderzoekend leren taak over het opladen van een condensator, met de elektronische leeromgeving Co-Lab waarin zij experimenten konden doen met een computer simulatie en vervolgens hun opgedane kennis in een model moesten weergeven.

Resultaten gaven aan dat experts minder tijd nodig hadden om de taak af te ronden dan de high-level novices, die evenveel tijd nodig hadden als de low-level novices. De kwaliteit van de modellen verschilden ook tussen de groepen: de experts maakte de beste modellen gevolgd door de high-level novices en tot slot de low-level novices. Kwalitatieve analyses van deze modellen lieten zien dat zelfs de low-level novices redelijk goed wisten te bepalen welke variabelen in het model van belang waren, maar niet hoe deze variabelen met elkaar samen hangen (m.a.w. de relaties tussen de variabelen).

Er werden slechts enkele verschillen gevonden in het leergedrag van de deelnemers. De drie groepen genereerden even specifieke hypotheses en voerden evenveel experimenten uit; bovendien was een vergelijkbaar percentage van deze experimenten exploratief. Het was verrassend dat slechts weinig low-level novices begonnen met experimenteren met de simulatie om domeinkennis op te doen. Tijdens het modelleren runden de experts hun model vaker dan de beide novice groepen. Een analyse van de derde onderzoekend leren vaardigheid (resultaat van de experimenten interpreteren en beoordelen) gaf aan dat low-level novices vaker hun hypothese verwierpen dan high-level novices en experts. Dit is waarschijnlijk terecht, omdat het aannemelijk is dat hun hypotheses vaker incorrect waren. Het accepteren en wijzigen van hypotheses kwam in de drie groepen even vaak voor. De hypothesen van de novices waren echter zo specifiek dat ze waarschijnlijk willekeurig dingen aan het proberen waren. Uit de analyse van de redenaties achter aanpassingen aan het model kwam dit ook naar voren: de experts en high-level novices beredeneerden deze aanpassingen, terwijl de low-level novices hun aanpassingen niet inhoudelijk konden onderbouwen.

Deze bevindingen wijzen op een duidelijke ondersteuningsbehoefte. De kwaliteit van de modellen die door de low-level novices waren gemaakt doet vermoeden dat deze leerlingen geen kennis hebben verworven tijdens de taak. Dit lijkt te komen doordat de low-level novices een vergelijkbare onderzoeksstrategie hanteerden als de deelnemers met meer voorkennis. Echter, zonder voorkennis (en ondersteuning) was deze strategie waarschijnlijk minder geschikt en zeker minder effectief. Ondersteuning bij onderzoekend leren zou er daarom voor moeten zorgen dat leerlingen hun onderzoeksvaardigheden beter afstemmen op hun kennisniveau. Leerlingen hebben er waarschijnlijk baat bij wanneer zij niet direct met kwantitatief (d.w.z. zeer specifiek) modelleren kunnen beginnen, maar in plaats daarvan modellen moeten maken waarin de relaties geleidelijk aan steeds specifieker moeten worden gedefinieerd.

Studie 2: Modelprogressie

Naar aanleiding van deze resultaten is in de tweede studie (zie Hoofdstuk 3) onderzocht of modelprogressie in de ondersteuningsbehoefte van leerlingen kan voorzien. Het basisidee achter modelprogressie is dat een leertaak wordt opgedeeld in kleine stapjes van toenemende complexiteit. Er werden twee typen modelprogressie onderscheiden. Bij model order progressie (MOP) neemt de complexiteit toe doordat de relatiebeschrijvingen steeds specifieker worden, terwijl bij model elaboratie progressie (MEP) de complexiteit toeneemt doordat het aantal elementen in het model wordt uitgebreid. Omdat uit de vorige studie naar voren kwam dat leerlingen, zonder ondersteuning, wel in staat zijn om de belangrijke variabelen te identificeren, maar niet kunnen bepalen hoe deze variabelen met elkaar samenhangen, werd verwacht dat MOP –die ingrijpt op de relaties tussen variabelen– de optimale vorm van ondersteuning is.

Om deze veronderstelling te onderzoeken zijn 4-VWO leerlingen op grond van hun voortoetsscores gematched over drie condities (MOP, $n = 28$; MEP, $n = 26$; controle, $n = 30$). Alle leerlingen werkten aan dezelfde taak over het opladen van een condensator in een elektrisch circuit in een leeromgeving die was afgestemd op de kenmerken van hun conditie. In de MOP conditie was het modelleren in drie fases opgedeeld. In de eerste MOP fase was het modelleren beperkt tot het schetsen van de structuur van het model (een weergave van de variabelen en de relaties). In de tweede MOP fase moesten de leerlingen de relaties kwalitatief specificeren (bijv. als de weerstand toeneemt, dan neemt de stroomsterkte af). In de derde MOP fase moesten de leerlingen de relaties kwantitatief specificeren (bijv. $I=V/R$). In elke fase moest het gemaakte model de volledig complexe simulatie beschrijven. Ook in de MEP conditie was de simulatie in drie fases opgedeeld. De simulatie in de eerste MEP fase bevatte een elektrisch circuit met een batterij en een lampje. In de tweede MEP fase was deze simulatie uitgebreid met een tweede, parallel geschakeld, lampje; in de derde MEP fase werd een condensator aan de simulatie toegevoegd. In elke fase moesten de leerlingen de inzichten die ze over de simulatie opdeden direct kwantitatief modelleren. De leertaak in de controle conditie was niet opgedeeld in fases, de leerlingen werkten met de volledig complexe simulatie die ze direct kwantitatief moesten modelleren.

Zoals verwacht bleek dat de condities van elkaar verschilden in het aantal correcte relaties in de modellen, maar niet in het aantal correcte variabelen. De leerlingen die ondersteund werden door modelprogressie maakten betere modellen dan de leerlingen in de controle conditie. Bovendien waren de modellen van de MOP leerlingen beter dan die van de MEP leerlingen.

Deze resultaten laten zien dat modelprogressie als ondersteuning bij onderzoekend leren gecombineerd met modelleren effectief kan zijn. Bovendien blijkt dat de dimensie waarop de progressie plaatsvindt van invloed is op de mate waarin leerlingen baat hebben bij modelprogressie. Uit eerder onderzoek is gebleken dat leerlingen behoefte hebben aan ondersteuning op het gebied van de relaties tussen variabelen. Model *order* progressie, waarbij leerlingen modellen maken waarbij deze relaties in specificiteit toenemen, blijkt een positief effect te hebben op het aantal relaties dat leerlingen correct in hun model weergeven. Dit veronderstelt dat model order progressie aansluit bij de ondersteuningsbehoefte van leerlingen.

Echter, hoewel model order progressie tot significant betere modellen leidde, waren de leerprestaties enigszins teleurstellend: zelfs de MOP-leerlingen behaalden gemiddeld slechts een derde van de maximale score. Op basis van deze studie zijn twee alternatieven geopperd om model order progressie te verbeteren. Het eerste alternatief betreft een uitbreiding met een software-agent die functioneert als ‘poortwachter’ om te voorkomen dat leerlingen met onvoldoende kennis naar een volgende fase gaan. Het tweede alternatief is juist het volledig vrijgeven van faseovergangen –zowel naar volgende als voorgaande fases– om beter aan te sluiten bij het iteratieve aspect van onderzoekend leren.

Studie 3: Modelprogressie verder verfijnd

Naar aanleiding van de resultaten uit de vorige studie, zijn in de derde studie (zie Hoofdstuk 4) de twee alternatieven om modelprogressie te verbeteren onderzocht.

Dit onderzoek is uitgevoerd bij leerlingen uit 4 VWO die op basis van voortoetsscores werden gematched over drie condities (begrensd, n = 19; semi-begrensd, n = 19; onbegrensd, n = 22). Alle leerlingen werkten aan dezelfde taak over het opladen van een condensator in een elektrisch circuit, waarin model order progressie (MOP) was toegepast zoals beschreven in de vorige paragraaf. De condities verschilden met betrekking tot de restricties bij de overgang van modelprogressie fases. Leerlingen uit de *begrensde* conditie konden alleen doorgaan naar een volgende fase op het moment dat hun modellen voldoende kennis weergaven. Leerlingen uit de *semi-begrensde* conditie hadden de ‘standaard’ vorm van modelprogressie (vergelijkbaar met MOP uit de vorige studie) en waren dus vrij om naar de volgende fase te gaan als ze dat wilden. Zij konden echter niet terugkeren naar een eerder bezochte fase. Leerlingen uit de *onbegrensde* conditie waren vrij om naar zowel de volgende als de vorige fases te gaan.

Restricties op de faseovergangen bleken het leerproces te beïnvloeden. In vergelijking met de semi-begrensde conditie gingen er minder leerlingen uit de begrensde conditie naar de tweede fase, maar de leerlingen uit de begrensde conditie die wel naar de tweede fase gingen, deden dat met betere modellen. Leerlingen uit de onbegrensde conditie gingen daarentegen vaker naar de tweede fase dan de semi-begrensde leerlingen. Zij deden dit echter met een vergelijkbare kwaliteit van modellen.

Hoewel beide alternatieven het leerproces beïnvloedden, bleken zij de leerprestaties niet te verbeteren. Het lijkt er dus op dat zowel het vrijlaten als begrenzen van de faseovergangen de effectiviteit van modelprogressie niet verbetert. Waarschijnlijk hebben leerlingen dus behoefte aan extra ondersteuning om beter te begrijpen wat er in elke modelprogressiefase van ze verwacht wordt en hoe ze daarbij te werk moeten gaan.

Studie 4: Modelprogressie met uitgewerkte voorbeelden

De vierde studie (zie Hoofdstuk 5) onderzocht de effectiviteit van het toevoegen van heuristische uitgewerkte voorbeelden aan modelprogressie. Heuristische uitgewerkte voorbeelden onderscheiden zich van 'gewone' uitgewerkte voorbeelden doordat zij een serie strategieën laten zien om een complexe leertaak aan te pakken, in plaats van één enkele algoritmische oplossing voor een eenvoudige leertaak. Heuristische uitgewerkte voorbeelden kunnen dus de onderzoeksvaardigheden voor elke modelprogressiefase demonstreren. Hierdoor werd verwacht dat heuristische uitgewerkte voorbeelden de activiteiten van leerlingen zouden verbeteren en daardoor ook hun leerprestatie en -uitkomsten.

Op basis van voortoetsscores werden 4 VWO leerlingen gematched over twee condities (uitgewerkte voorbeelden, n = 46; controle, n = 36). Alle leerlingen werkten aan dezelfde taak over het opladen van een condensator in een elektrisch circuit, waarin model order progressie was toegepast zoals beschreven in de vorige twee studies. De leerlingen uit de uitgewerkte voorbeelden conditie konden per fase twee heuristische uitgewerkte voorbeelden raadplegen. Deze uitgewerkte voorbeelden werden aangeboden als een video die liet zien hoe een anoniem persoon de onderzoeksvaardigheden toepast in een vergelijkbare taak in een ander domein. De leerlingen uit de controle conditie hadden geen toegang tot deze uitgewerkte voorbeelden.

De resultaten gaven aan dat de leerlingen een vergelijkbaar niveau van voorkennis hadden, evenveel tijd nodig hadden voor de taak, maar verschilden in hoe ze deze tijd besteedden. Zoals de uitgewerkte voorbeelden aangaven, deden leerlingen uit de gelijknamige conditie meer experimenten met de simulatie en gebruikten ze meer tijd om de uitkomsten van de experimenten te evalueren. De leerlingen uit de controle conditie deden daarentegen weinig met de simulatie; zij besteedden het overgrote deel van hun tijd aan het maken en testen van hun model. Deze werkwijze bleek weinig effectief: de leerlingen uit de uitgewerkte voorbeelden conditie maakten betere modellen dan de leerlingen in de controle conditie. Ondanks deze verschillen in werkwijze en leerprestatie, waren er geen verschillen op de natoets. Dit geeft aan dat de leerlingen uit de uitgewerkte voorbeelden conditie tijdens de taak beter presteerden, maar hier niet meer van leerden.

Conclusie

Samengevat laten deze vier studies zien dat leerlingen ondersteuning nodig hebben bij het leren met computersimulaties en modellen. Deze ondersteuning moet leerlingen helpen de onderzoeksvaardigheden toe te passen op een niveau dat past bij hun kennisniveau, met name bij het definiëren van de relaties tussen de variabelen in het model. Dergelijke ondersteuning kan worden aangeboden in de vorm van model order progressie, waarbij de complexiteit van een taak gradueel oploopt door een toenemende specificiteit van de relatiebeschrijvingen. Leerlingen hoeven dan eerst alleen een structuur van het model te geven (een weergave van de variabelen en relaties). In de tweede modelprogressiefase moeten de leerlingen dan de relaties kwalitatief specificeren en pas in de derde modelprogressiefase hoeven zij de relaties kwantitatief te specificeren.

Hoewel model order progressie de prestatie van leerlingen op de taak verbeterde, was deze verbetering nog niet toereikend. Uit een poging om modelprogressie te verfijnen bleek dat het begrenzen noch het vrijlaten van de mogelijkheden om van fase te wisselen de leerprestaties kon verbeteren. Er werd geconcludeerd dat leerlingen waarschijnlijk meer baat zouden hebben bij extra ondersteuning die aangeeft wat er in elke modelprogressie fase van ze verwacht wordt en hoe ze te werk moeten gaan. Deze ondersteuning –in de vorm van heuristische uitgewerkte voorbeelden– leidde tot een significante verbetering van de leerprestaties.