Validating and Optimizing the Effects of Model Progression in Simulation-Based Inquiry Learning

Yvonne G. Mulder · Ard W. Lazonder ·
Ton de Jong · Anjo Anjewierden · Lars Bollen

Published online: 30 December 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract Model progression denotes the organization of the inquiry learning process in successive phases of increasing complexity. This study investigated the effectiveness of model progression in general, and explored the added value of either broadening or narrowing students' possibilities to change model progression phases. Results showed that high-school students in the 'standard' model progression condition ($n = 19$), who could enter subsequent phases at will, outperformed students from a control condition ($n = 30$) without model progression. The unrestricted condition ($n = 22$) had the additional option of returning to previous phases, whereas the restricted condition ($n = 20$) disallowed such downward progressions as well as upward progressions in case insufficient knowledge was acquired. Both variants were found to be more effective in terms of performance than the 'standard' form of model progression. However, as performance in all three model progression conditions was still rather weak, additional support is needed for students to reach full understanding of the learning content.

Keywords Inquiry learning · Model-based learning · Model progression

Introduction

Technology-enhanced inquiry learning environments enable students to develop a deep understanding of a

domain by engaging in scientific reasoning processes such as hypothesis generation, experimentation, and evidence evaluation. Computer simulations have long been incorporated in these environments, and 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, modeling is considered integral to the inquiry learning process in that students have to build models to express their understanding of the relation between variables (Van Joolingen et al. 2005; White et al. 1999). Students can check their understanding by running the model, and weighting its output against prior knowledge or the data from the simulation.

In educational practice, however, the educational advantages of inquiry learning are often challenged by the students' modest inquiry skills. De Jong and Van Joolingen's (1998) review proved that students are generally unable to infer hypotheses from data, design inconclusive experiments, show inefficient experimentation behavior, and ignore incompatible data. Similar problems arise during modeling. Hogan and Thomas (2001), for instance, noticed that students often fail to engage in dynamic iterations between examining output and revising models, and Stratford et al. (1998) observed a lack of persistence in debugging models to fine-tune their behavior.

These learning difficulties can be significantly reduced by embedding process and content explanations within the learning environment (e.g., Zhang et al. 2004; Fund 2007; Lazonder et al. 2010). Yet other studies have shown that these text-based supports can also be neglected during task performance (Aleven et al. 2003; Clarebout and Elen 2006), and become ineffective or even counterproductive when students gain experience (Kalyuga 2007). A potentially fruitful alternative might be to adapt task complexity

Y. G. Mulder (✉) · A. W. Lazonder · T. de Jong ·
A. Anjewierden · L. Bollen
Department of Instructional Technology, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: y.g.mulder@utwente.nl

to the students' increasing levels of domain understanding by structuring the task content according to a simple-to-complex sequence. This type of learning support was first introduced by White and Frederiksen (1990), who termed it 'model progression'.

Model progressions are often created by introducing the variables that can be investigated through the simulation one at a time. Research on this form of model progression has produced mixed findings. Some studies report that learning with increasingly more elaborate simulations is more effective than learning with a full simulation (Rieber and Parmley 1995; Swaak et al. 1998), whereas other studies found no such effects (De Jong et al. 1999; Quinn and Alessi 1994). These inconsistent findings suggest that assigning students to a gradually expanding set of simulation variables is probably not the best way to arrange inquiry learning tasks in a simple-to-complex sequence. A recent study confirmed that domain novices are quite capable of identifying relevant variables, but experience considerable difficulties in specifying the relations between these variables (Mulder et al. 2010). Instead of gradually working toward a full-fledged scientific equation to specify a relationship, novices tried to induce and model these equations from scratch. It thus seems that novice learners could benefit from model progressions that enable them to engage in increasingly specific reasoning about the way variables are interrelated.

This assumption was validated in a follow-up study that compared two types of model progression (Mulder et al. 2011). Both types divided the inquiry task in three successive phases, but differed with regard to the sequencing principle that determined how task complexity increased across these phases. Model *order* progression, the predicted optimal variant, gradually increased the specificity of the relations between variables, whereas model *elaboration* progression gradually expanded the number of variables in the task. Students who received either form of model progression performed better than students from an unsupported control group. A comparison among the two model progression conditions confirmed that students in the model order group outperformed those from the model elaboration group on the construction of relations in their models.

However, even the best-performing students in the model order condition 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 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. 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. This option, which was not available to students in the Mulder et al. (2011) study, seems more consistent with the iterative nature of the inquiry learning process.

The present study put both improvement options to the test. The basic premise underlying this research was that model order progression enhances performance, and that both broadening and narrowing students' possibilities to choose their own learning paths through the pre-defined task sequence will further improve its effectiveness. Both assumptions were investigated in a between-group design with four conditions. A comparison of the 'standard' model progression condition with the control condition assessed the effectiveness of model order progression per se. This analysis was a replication of the Mulder et al. (2011) study, and was deemed necessary because research on other forms of model progression failed to produce consistent cross-study findings, even when conducted by the same researchers or research groups (Swaak et al. 1998; De Jong et al. 1999; Quinn and Alessi 1994; Alessi 1995). Model order progression in the remaining two conditions was supplemented with one of the improvement options. Students in the unrestricted condition had the additional possibility of returning to previous phases whereas students in the restricted condition were neither allowed such downward progressions nor upward progressions in case of insufficient knowledge. Both variants were predicted to be more effective than the 'standard' form of model order progression in which students could enter subsequent phases at will, but not return to previous phases.

Methods

Participants

Ninety-one Dutch high-school students participated in the experiment as part of their regular physics lessons. The sample comprised 47 boys and 44 girls in the age of 15–17, who were assigned to experimental conditions on the basis of class-ranked pretest scores. This led to 20 participants in the restricted condition, 19 in the semi-restricted condition, 22 in the unrestricted condition, and 30 in the control condition.

Inquiry Task and Learning Environment

Participants worked on an inquiry task about the charging of a capacitor. They were assigned to examine an electrical circuit in which a capacitor was embedded, and create a

computer model that mirrors the capacitor's charging behavior. Participants performed this task within a stand-alone version of the Co-Lab learning environment (Van Joolingen et al. 2005) that stored all participants' actions in a log file.

The learning environment housed a simulation of an electrical circuit containing a voltage source, two light bulbs, and a capacitor. Participants could experiment with the simulation to find out how these components behaved, and then use the model editor tool to represent their knowledge in an executable computer model. As shown in Fig. 1, these 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 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 enabled participants to test their understanding by running the model and analyzing its output through a table and 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.

Configuration of the Learning Environment Within Each Condition

All conditions used the same instructional content, but differed with regard to whether and how model progression was implemented. Participants in the *control condition* worked with the standard configuration of the environment described above, and were not supported by model progression.

In the remaining three conditions, model progression was implemented by dividing the inquiry task in three 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., $I = V/R$).

The three model progression conditions differed with regard to the restrictions to enter these three phases (see Fig. 2). Participants in the *restricted condition* could not return to previous phases, and could progress to a subsequent phase if their model was of sufficient quality.

Fig. 1 Screenshot of the model editor tool

Fig. 2 Schematic overview of the four conditions

This check was performed by a software agent that assessed the students' models against a predefined ruleset. Participants could try to enter a subsequent phase at any time, but the software agent granted access only if their model satisfied the requirements of the ruleset.

The ruleset was based on the similarity between the student's model and the reference model shown in Fig. 1. Minimal requirements for the transition from Phase 1 to Phase 2 were the presence of either four constants (C , S , $R1$, and $R2$) or three constants and the stock variable (charge), one auxiliary variable (V_c , V_r , I , or R), and all relationship arrows connectors between these five elements. For the transition from Phase 2 to Phase 3, this ruleset was extended with the requirement to have a correct, qualitative specification for all but one of the relationship arrows.

The semi-restricted condition incorporated the 'standard' form of model progression that was used by Mulder et al. (2011). Participants in this condition could progress

to subsequent phases at will and without any restrictions imposed by the software agent. They could, however, not return to previous phases.

Participants in the unrestricted condition had no phase-change restrictions at all. They were free to go to subsequent and previous phases as they deemed fit.

Pretest

A pretest consisting of eight open questions assessed participants' prior knowledge of electrical circuits. Four items addressed the meaning of key domain concepts (i.e., voltage source, resistance, capacitor, and capacitance); the other four items addressed the physics equations that govern the behavior of a charging capacitor. Participants' answers to the eight questions were scored as either correct or incorrect using the rubric of Mulder et al. (2011). The rubric's 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. 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 operation of the modeling tool.

The experimental session started with the announcement 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 consult the help files to learn about the specifics of the condition they were assigned to. 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.

Measures

All data were assessed from the log files. Variables under investigation were time on task, learning activities, and performance success. Time on task concerned the duration of the experimental session. Learning activities were the number of experiments with the simulation, the number of model runs, and the number of phase changes. The latter measure was defined as the number of attempts to enter an other phase. Where appropriate, a distinction was made between progressions to subsequent and previous phases.

Performance success was assessed from the participants' final models using the rubric of Manlove et al. (2006). The resulting score represents the number of correctly specified variables and relations in the models. 'Correct' was judged from the reference model displayed in Fig. 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. Up to three additional points could be earned if the direction, type (i.e., qualitative specification), and magnitude of effect (i.e., quantitative specification) of the relation was correct. The maximum performance success score was 54. The rubric's inter-rater reliability for variables (Cohen's $\kappa = .74$) and relations (Cohen's $\kappa = .92$) was sufficient.

Results

Preliminary analyses were performed to check whether the matching of participants had resulted in comparable levels

of prior knowledge across conditions. The mean pretest scores are presented in Table 1. Univariate analysis of variance (ANOVA) revealed no significant differences in prior knowledge among the four conditions, $F(3, 87) = .61$, $p = .612$. Data for time on task indicated that participants in all conditions needed approximately 90 min to complete the assignment. ANOVA showed that time on task differed among conditions, $F(3, 87) = 4.34$, $p = .007$, $\eta_p^2 = .130$.¹ Planned contrasts, using the simple method with the semi-restricted condition as reference category, showed that the control condition used more time than the semi-restricted condition. The other differences in time reported in Table 1 were not statistically significant.

Table 1 also gives an account of the activities participants performed within the learning environment. Multivariate analysis of variance (MANOVA) indicated a significant difference in the number of simulation experiments and model runs, $F(6, 174) = 9.29$, $p < .001$. Subsequent ANOVAs produced a significant effect for condition on both types of activities (simulation experiments: $F(3, 87) = 12.82$, $p < .001$, $\eta_p^2 = .307$; model runs: $F(3, 87) = 12.87$, $p < .001$, $\eta_p^2 = .307$). Planned contrasts showed that participants in the control condition performed more simulation experiments and fewer model runs than participants from the semi-restricted condition. The differences among the model progression conditions were not statistically significant.

The three model progression conditions had different phase change restrictions. There was a significant association between the type of restriction and whether or not participants reached Phase 2, $\chi^2(2, N = 61) = 19.36$, $p = .006$. The transition from Phase 2 to Phase 3 was independent of the type of phase change restriction, $\chi^2(2, N = 41) = 4.16$, $p = .125$. As the odds ratio indicates, participants in the semi-restricted condition were 8.75 times more likely to enter Phase 2 than participants from the restricted condition. Even though the latter participants tried to change phases 5.21 times on average ($SD = 7.31$), these attempts were often foiled by the software agent: only six participants were granted access to Phase 2 and none of them managed to enter Phase 3. Participants in the semi-restricted and unrestricted condition were free to progress to subsequent phases, and did so more often (see Table 2). The odds ratio suggests that the likelihood of entering Phase 2 was 2.67 times lower in the semi-restricted condition than in the unrestricted condition. In addition, all but four participants in the unrestricted condition used the possibility to visit a previous phase. The

¹ The partial eta squared (η_p^2) effect-size estimate indicates the proportion of the variance attributable to the independent variable when other factors are partialled out. A value of .01 is generally considered a small effect, .09 a medium effect, and .25 a large effect.

Table 1 Descriptive statistics for participants' performance by condition

	Restricted		Semi-restricted		Unrestricted		Control	
	<i>M</i>	SD	<i>M</i>	SD	<i>M</i>	SD	<i>M</i>	SD
Pretest score	1.65	1.46	1.21	1.13	1.27	1.28	1.23	1.01
Time on task (min)	85.36	11.15	86.58	14.02	90.99	5.47	95.48	11.66
<i>Learning activities</i>								
Simulation experiments	15.95	12.77	12.47	6.50	13.45	8.94	35.40	23.22
Model runs	63.70	38.40	68.47	51.25	63.59	34.64	17.47	10.45
<i>Performance success</i>								
Variables	9.85	4.06	8.00	3.33	9.86	2.77	6.63	1.63
Relations	6.75	5.28	7.79	5.55	8.59	3.97	3.17	3.24

Table 2 Mean performance success scores for variables and relations at phase change by model progression condition

Performance success	Restricted (<i>N</i> = 20)			Semi-restricted (<i>N</i> = 19)			Unrestricted (<i>N</i> = 22)		
	<i>n</i>	<i>M</i>	SD	<i>n</i>	<i>M</i>	SD	<i>n</i>	<i>M</i>	SD
<i>From Phase 1 to Phase 2</i>									
Variables	6	13.17	2.99	15	8.20	3.76	20	8.85	3.30
Relations	6	11.17	3.55	15	6.47	4.26	20	6.05	4.49
<i>From Phase 2 to Phase 3</i>									
Variables	0	–	–	5	6.80	3.56	9	9.67	3.24
Relations	0	–	–	5	4.20	3.03	9	9.33	4.27

average number of phase regressions ($M = 3.63$, $SD = 2.16$) approached the number of phase progressions ($M = 4.31$, $SD = 2.36$).

Participants' performance success was analyzed by MANOVA with the constituent scores for variables and relations as dependent variables (see Table 1). This analysis produced a significant effect for condition, $F(6, 174) = 5.08$, $p < .001$. Subsequent ANOVAs showed that condition affected both the number of correct variables in the students' models, $F(3, 87) = 7.19$, $p < .001$, $\eta_p^2 = .199$, and the quality of the relations between these variables, $F(3, 87) = 7.68$, $p < .001$, $\eta_p^2 = .209$. Planned contrasts revealed that the variable scores in the semi-restricted condition were comparable to those in the control condition, whereas the relation scores were significantly higher in the semi-restricted condition. A reverse pattern in scores was obtained for participants in the restricted and unrestricted condition: their models contained more correct variables, but were comparable in terms of relations.

Performance success within the model progression conditions was also assessed at each phase change. Table 2 reports the descriptive statistics for these assessments, indicating how the quality of the participants' models developed through time. MANOVA produced a significant effect for condition on performance success at the first phase change, $F(4, 76) = 2.50$, $p = .049$, but not at the

second, $F(2, 11) = 2.58$, $p = .121$). Subsequent ANOVAs demonstrated that the difference at the first phase change involved the scores for both variables, $F(2, 38) = 4.72$, $p = .015$, $\eta_p^2 = .199$, and relations, $F(2, 38) = 3.43$, $p = .043$, $\eta_p^2 = .153$. Planned contrasts showed that this difference arose because participants in the restricted condition had significantly higher variables and relation scores than participants in the semi-restricted condition. Performance success in the unrestricted condition was comparable to that in the semi-restricted condition.

Discussion

The first aim of this study was to assess the effectiveness of model progression by comparing the performance of students who received the 'standard' form of model progression (i.e., the semi-restricted condition) with that of students from the control condition who received no such support. Results showed that even though control participants spent approximately 10 min more time on task, and carried out more than twice as many simulation experiments (but fewer model runs), performance success was higher in the semi-restricted condition, in particular because participants' models contained better relations. (Even more pronounced differences in performance

success are found when time on task is controlled for in the analyses.) It thus seems that model progression leads to more efficient and effective performance. This outcome corroborates the conclusion of Mulder et al. (2011) that students benefit from model order progressions that gradually increase the specificity of the students' reasoning about the relations between variables. The success of the present replication is particularly noteworthy because previous attempts to replicate the effectiveness of model progression have generally been unsuccessful (e.g., De Jong et al. 1999; Quinn and Alessi 1994). A possible explanation is that the latter studies progressed task complexity along different dimensions (i.e., the degree of realism in the simulation interface and the number of variables that could be manipulated).

The present research also examined whether the effects model progression are enhanced by broadening or narrowing phase change restrictions. Students in all three model progression conditions spent as much time on task and conducted as many simulation and model runs. Despite this equivalence of learning activities, students in the restricted and unrestricted condition had better final models with more correctly specified variables than students in the semi-restricted condition. The advantages of the restricted variant can be explained by the software agent that obliged students to create high-quality models within each phase. While this quality threshold caused few students to progress to subsequent phases, it also ensured that the students who entered Phase 2 had high-quality models, and the ones who remained in Phase 1 had to devote most of their attention to specifying variables.

Higher performance success in the unrestricted condition seems due to the possibility to revisit previous phases. This opportunity provides a safety net that may have persuaded comparatively many students to visit subsequent phases. As these upward progressions occurred only slightly more often than downward regressions, it seems that working in a more advanced phase made students aware of certain imperfections in their models which they then tried to improve in a previous phase. This was substantiated by the fact that their model scores upon first entering Phase 2 and 3 resembled those in the semi-restricted condition.

From these findings it can be concluded that both variants offer a notable albeit modest improvement to the implementation of model progression. A qualified optimism is in order because the students' final models were as mediocre as in the Mulder et al. (2011) study. It thus seems that even with more appropriate phase change restrictions, students need more time or additional support to fully understand the task content. This is perhaps most apparent in the restricted condition, where only 6 of the 20 students reached Phase 2. Successful phase changes occurred after

approximately 80 min, which obviously leaves too little time to make it to the final phase. This in turn might explain why the difference in relation scores in Phase 2 'vanished' in the final model score: too few students in the restricted condition reached a point in their inquiry where they could concentrate solely on the relations in their models.

Insufficient time or support could have impaired performance in the unrestricted condition as well. Students in this condition cycled through the phases, and these iterations enabled them to specify more variables correctly. A difference in relation scores was not found, however. This seems due to the fact that most phase changes (85%) occurred between Phase 1 and 2 which implies that phase regressions were aimed at improving variables. Similar iterations between Phase 2 and 3 could have enabled students to enhance the relations in their models, but time constraints caused few students to reach Phase 3, and the ones that did apparently had too little time to take full advantage of the opportunity to return to Phase 2. On the positive side, this result indicates that students generally managed to attune phase changes to their level of understanding. The relatively high model scores upon entering Phase 3 substantiate this claim.

Future research might investigate how both model progression variants can be further improved. Extending time on task appears to be the most straightforward solution, but one may wonder whether extra time alone is sufficient to successfully complete the task. Model progression merely adapts task complexity to the learners' evolving domain understanding, and does not offer any directions or guidance on how the task itself should be performed. The absence of such explicit support may have caused students in this study to progress slowly through the phases, and/or create suboptimal models. Extra class time does not alleviate these problems, and science teachers may consider this option unfeasible or undesirable. A more practical solution might therefore be to supplement the current lessons with additional support to help students perform the task more efficiently and effectively.

Prior work in this direction examined the use of assignments to structure the students' inquiry activities within model progression phases. These attempts proved unsuccessful (Swaak et al. 1998), even when students receive adaptive feedback on their solutions (Veermans et al. 2000). A more sophisticated, and possibly more effective approach would be to offer adaptive support on the students' actions through the software agent. By using data mining techniques, the agent can detect patterns in the students' inquiry and modeling activities, and use this information to give tailor-made assistance and feedback at times appropriate. Such techniques have been successfully applied in small-scale modeling tasks (Bravo et al. 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.

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

References

- 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
- Aleven V, Stahl E, Schworm S, Fischer F, Wallace R (2003) Help seeking and help design in interactive learning environments. *Rev Educ Res* 73:277–320
- Bravo C, Van Joolingen WR, De Jong T (2009) Using Co-Lab to build system dynamics models: students' actions and on-line tutorial advice. *Comput Educ* 53:243–251. doi:10.1016/j.compedu.2009.02.005
- Clarebout G, Elen J (2006) Tool use in computer-based learning environments: towards a research framework. *Comput Hum Behav* 22:389–411. doi:10.1016/j.chb.2004.09.007
- De Jong T, Van Joolingen WR (1998) Scientific discovery learning with computer simulations of conceptual domains. *Rev Educ Res* 68:179–202. doi:10.3102/00346543068002179
- De Jong T, Martin E, Zamarro JM, Esquembre F, Swaak J, Van Joolingen WR (1999) The integration of computer simulation and learning support: an example from the physics domain of collisions. *J Res Sci Teach* 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 WR, Giezma A, Girault I, Hoppe U, Kindermann J et al (2010) Learning by creating and exchanging objects: the SCY experience. *Br J Educ Technol* 41:909–921. doi:10.1111/j.1467-8535.2010.01121.x
- Fund Z (2007) The effects of scaffolded computerized science problem-solving on achievement outcomes: a comparative study of support programs. *J Comput Assist Learn* 23:410–424. doi:10.1111/j.1365-2729.2007.00226.x
- Hogan K, Thomas D (2001) Cognitive comparisons of students' systems modeling in ecology. *J Sci Educ Technol* 10:319–345. doi:10.1023/A:1012243102249
- Kalyuga S (2007) Expertise reversal effect and its implications for learner-tailored instruction. *Educ Psychol Rev* 19:509–539. doi:10.1007/s10648-007-9054-3
- Lazonder AW, Hagemans MG, De Jong T (2010) Offering and discovering domain information in simulation-based inquiry learning. *Learn Instr* 20:511–520. doi:10.1016/j.learninstruc.2009.08.001
- Manlove S, Lazonder AW, De Jong T (2006) Regulatory support for collaborative scientific inquiry learning. *J Comput Assist Learn* 22:87–98. doi:10.1111/j.1365-2729.2006.00162.x
- Mulder YG, Lazonder AW, De Jong T (2010) Finding out how they find it out: an empirical analysis of inquiry learners' need for support. *Int Sci Educ* 32:2033–2053. doi:10.1080/09500690903289993
- Mulder YG, Lazonder AW, De Jong T (2011) Comparing two types of model progression in an inquiry learning environment with modelling facilities. *Learn Instr* 21:614–624. doi:10.1016/j.learninstruc.2011.01.003
- Quinn J, Alessi S (1994) The effects of simulation complexity and hypothesis-generation strategy on learning. *J Res Comput Educ* 27:75–91
- Rieber LP, Parmley MW (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. *J Educ Comput Res* 13:359–374. doi:10.2190/M8VX-68BC-1TU2-B6DV
- Stratford SJ, Krajcik J, Soloway E (1998) Secondary students' dynamic modeling processes: analyzing, reasoning about, synthesizing, and testing models of stream ecosystems. *J Sci Educ Technol* 7:215–234. doi:10.1023/A:1021840407112
- Swaak J, Van Joolingen WR, De Jong T (1998) Supporting simulation-based learning: the effects of model progression and assignments on definitional and intuitive knowledge. *Learn Instr* 8:235–252. doi:10.1037/a0021017
- Van Joolingen WR, De Jong T, Lazonder AW, Savelsbergh E, Manlove S (2005) Co-Lab: research and development of an on-line learning environment for collaborative scientific discovery learning. *Comput Hum Behav* 21:671–688. doi:10.1016/j.chb.2004.10.039
- Veermans K, Van Joolingen WR, De Jong T (2000) Promoting self-directed learning in simulation-based discovery learning environments through intelligent support. *Interact Learn Environ* 8:229–255. doi:10.1076/1049-4820(200012)8:3;1-D;FT229
- White BY, Frederiksen JR (1990) Causal model progressions as a foundation for intelligent learning environments. *Artif Intell* 42:99–157. doi:10.1016/0004-3702(90)90095-h
- White BY, Shimoda TA, Frederiksen JR (1999) Enabling students to construct theories of collaborative inquiry and reflective learning: computer support for metacognitive development. *Int J Artif Intell Educ* 10:151–182
- Zhang J, Chen Q, Sun Y, Reid DJ (2004) Triple scheme of learning support design for scientific discovery learning based on computer simulation: experimental research. *J Comput Assist Learn* 20:269–282. doi:10.1111/j.1365-2729.2004.00062.x