Combining Heuristics and Formal Methods in a Tool for Supporting Simulation-Based Discovery Learning

Koen Veermans[1] and Wouter R. van Joolingen[2]

[1] Faculty of Behavioral Sciences University of Twente, PO Box 217
7500 AE Enschede, The Netherlands
veermans@edte.utwente.nl

[2] Graduate school of teaching and learning, University of Amsterdam, Wibautstraat 2-4
1091 GM Amsterdam, The Netherlands
wouter@ilo.uva.nl

Abstract. This paper describes the design of a tool to support learners in simulation-based discovery learning environments. The design redesigns and extents a previous tool to overcome issues that came up in a classroom learning setting. The tool focuses on supporting learners with experimentation to identify or test hypotheses. The aim is not only to support learning domain knowledge, but also learning discovery learning skills. For this purpose the tool uses heuristics and formal methods to assess the learners experimenting behavior, and translates this assessment into feedback directed at improving the quality of the learners discovery learning behavior. The tool is designed to be part of an authoring environment for designing simulation-based learning environments, which put some constraints on the design, but also ensures that the tool can be reused in different learning environments. After describing the design, a learning scenario is used to serve as an illustration of the tool, and finally some concluding remarks, evaluation results, and potential extensions for the tool are presented.

1 Introduction

Discovery learning or Inquiry Learning has a long history in education [1, 4] and has regained popularity over the last decade as a result of changes in the field of education that put more emphasis on the role of the learner in the learning process. Zachos, Hick, Doane, and Sargent [19] define discovery learning as “the self-attained grasp of a phenomenon through building and testing concepts as a result of inquiry of the phenomenon” (p. 942). The definition emphasizes that it is the learner who builds concepts, that the concepts need to be tested, and that building and testing of concepts are part of the inquiry of the phenomenon. Computer simulations have rich potential to provide learners with opportunities to build and test concepts, and learning with these computer simulations is also referred to as simulation-based discovery learning.

Like in discovery learning, the idea of simulation-based discovery learning is that the learner actively engages in a process. In an unguided simulation-based discovery environment learners have to set their own learning goals. At the same time they have to find and apply the methods that help to achieve these goals, which is not always easy. Two main goals can be associated with simulation-based discovery learning; development of knowledge about the domain of discovery, and development of skills

J.C. Lester et al. (Eds.): ITS 2004, LNCS 3220, pp. 217–226, 2004.

that facilitate development of knowledge about the domain (i.e., development of skills related to the process of discovery).

This paper describes a tool that combines support for learning the domain knowledge with specific attention for learning discovery learning skills. Two constraints had to be taken into account in the design of the tool. The first constraint is related to the exploratory nature of discovery learning. To maintain the exploratory nature of the environment, the tool may be *directive*, should try to be *stimulating* and must be *non-obligatory*, leaving room for exploration to the learner. The second constraint is related to the context in which the tool should be operating, SIMQUEST [5], an authoring environment for the design and development of simulation-based learning environments. Since SIMQUEST allows the designer to specify the model, the domain will not be known in advance, and therefore, the support cannot rely on domain knowledge.

2 Learning Environments

At the core of SIMQUEST learning environments are one or more simulation models; visualized to learners through representations of the model (numerical, graphical, animated, etc.) in simulation interfaces. SIMQUEST includes templates for assignments (f.i. exercises that provide a learner with a subgoal), explanations (f.i. background information or feedback on assignments) and several tools (f.i. experiment storage tool). These components can be used to design a learning environment that supports learners. The control mechanism determines when the components present themselves to the learner and allows the designer to specify the balance between system control and learner control in the interaction between learning environment and learner.

This framework allows authors to design and develop simulation-based learning environments, and to some extent support for learners working with these learning environments. However, it does not provide a way of assessing of and providing individualized support on the learners' experimentation with a simulation. This was the starting point for the design of a tool called the 'monitoring tool' [16]. It supported experimenting by learners based on a formal analysis of their experimentation in relation to hypotheses (these hypotheses had to be specified by the designer in assignments). A study [17] showed positive results, but also highlighted two important problems with the monitoring tool.

The first problem is that one of the strengths of the monitoring tool is also one of its weaknesses. The monitoring tool did not rely on domain knowledge for the analysis of the learners' experimentation. The strength of this approach is that it is domain independent, the weakness that it can not use knowledge about the domain to correct learners when this might be needed. This might lead to incorrect domain knowledge, and incorrect self-assessment of the exploration process, because the outcome of the exploration process serves as a benchmark for learners in assessing the exploration process [2]. In the absence of external feedback, learners have to rely on their own assessment of the outcome of the process. If this assessment is incorrect, the resulting assessment of the exploration might also be incorrect.

The second problem is that the design of the tool was based primarily on formal principles related to induction and deduction. This had the shortcoming that it could only give detailed feedback about experimentation in combination with certain categories of hypothesis, like for instance semi-quantitative hypotheses (f.i. *"If the velocity becomes twice as large then kinetic energy becomes four times as large").* In more common language this hypothesis might be expressed as: "*There is a quadratic relation between velocity and kinetic energy*", but this phrasing has no condition part that can be used to make a formal assessment of experiments.

As a solution for this second problem the tool is extended with less formal, i.e. heuristic assessment of the experimentation. The heuristics that were used for this purpose originate from an inventory [12] of literature [4, 7, 8, 9, 10, 11, 13, 14, 15] about problem solving, discovery learning, simulation-based learning, and machine discovery, in search for heuristics that could prove useful in simulation-based discovery learning. A set of heuristics (Table 1) related to experimentation and hypothesis testing was selected from this inventory for the present purpose.

Table 1. Heuristics that were incorporated in the tool

Keep track	Keep records of what you are doing. [7,9]
Simple values	Design experiments giving characteristic results. [8] Choose special cases, set any parameter to 1,2,3. [13]
Votat	If a variable is not relevant for the hypothesis under, test then hold that variable constant, or vary one thing at a time (VOTAT), or If not varying a variable, then pick the same value as used in the previous experiment [4, 7, 14, 15]
Identify Hypothesis	Generate a small amount of data and examine for a candidate rule or relation. [4]
Equal increments	If choosing a third value for a variable, then choose an equal increment as between first and second values. Or if manipulating a variable, then choose simple, canonical manipulations. [14]
Confirm Hypothesis	Generate several additional cases in an attempt to either confirm or disconfirm the hypothesized relation. [4]
Extreme values	Try some extreme values to see if there are limits on the proposed relationship.[14]
Inductive discovery heuristics	-If you have recorded a set of values for X and a set of values for Y, and the values of X and Y are have a constant ratio of increments, then infer that a linear relation exists between X and Y -If you have recorded a set of values for X and a set of values for Y, and the absolute value of X increases and the absolute value of Y increases, and these values are not linearly related, Then consider the ratio of X and Y -If you have recorded a set of values for X and a set of values for Y, and the absolute value of X increases and the absolute value of Y decreases, and these values are not linearly related, then consider the product of X and Y [10, 11]

Heuristic assessment of the experimentation will allow the tool to provide feedback on experimentation without needing specific hypotheses as input for the process of evaluating the learners' experiments. Consequently, the hypotheses in the assignments can now be stated in "normal" language, which makes it easier for the learners not only to investigate, but also to conceptualize them. If the hypothesis in the assignment is no longer used as input for the analysis of the learners' experimentation, it is also no longer needed to connect the experimentation feedback to assignments. This means that feedback on the correctness of the hypothesis can be given in the assignment, thus, solving the first problem. The feedback on experimentation can be moved to the tool in which the experiments are stored; a more logical place to provide feedback on experimentation. Moving the feedback to this tool requires it to be redesigned, and this was the starting point for a redesign of the tool.

3 Redesign of the Experiment Storage Tool

Originally the experiment storage tool was only a storage place for experiments. If the tool should provide feedback on experimenting it means there should be a point at which this feedback is communicated to the learner, preferable not disrupting the learner. It was decided to extend the experiment storage tool with a facility to draw graphs, and combine the feedback with the learner-initiated action of drawing a graph. Figure 1gives an overview of the position of the new tool within the system.

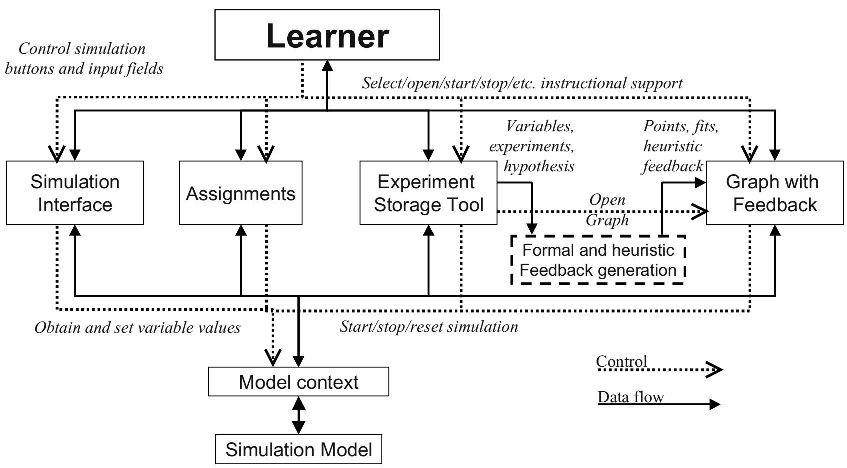

Fig. 1. The structure of control and information exchange between a learner and a SimQuest learning environment with the new experiment storage tool with graphing and heuristic support

Drawing a graph is not a trivial task and has been the object of instruction in itself [6]. It was therefore decided to let the tool take care of drawing the graph, but to provide feedback related to drawing and interpreting graphs to the learner, as well as, feedback related to experimenting. The learner has to do is to select a variable for the x-axis, and a variable for the y-axis, which provides the tool with important information that can be used for generating feedback. Through the choice of variables the learner expresses interest in a certain relation.

Learners can ask the tool to fit a function on the experiments along with drawing a graph. Basic qualitative functions (monotonic increase and monotonic decrease), and quantitative functions (constant, linear, quadratic, and reciprocal) are provided to the learners. More functions could of course be provided, but it was decided to restrict the set of functions to the functions first, because too many possibilities might overwhelm learners. Fitting a function is optional, but when a learner selects this option it provides the tool with valuable extra information for the analysis of the experimentation.

Learners can also construct new variables based on existing variables. New variables can be constructed using basic simple arithmetic functions add, subtract, divide, and multiply. Whenever the learner creates a new variable, a new column will be added to the experiment storage tool, and this column will also be updated for new experiments. The learner can compare these values to other values in the table to see how the newly constructed variable relates to the variables that were already listed in the monitoring tool. The redesigned version of the monitoring tool with its new functionality is shown in Figure 2.

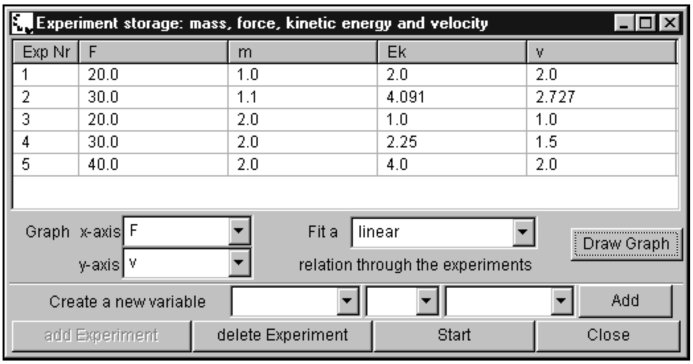

Fig. 2. The experiment storage tool.

4 Providing Heuristic Support

The previous section described the basic functions of the experiment storage tool. This section describes how the tool will provide support for the learner. Three different parts can be distinguished in the support: drawing the data points, calculating and drawing a fit, and providing feedback based on the heuristics from Table 1 The first two parts are rather straightforward, and will therefore not be described in detail.

The heuristics from Table 1 were divided into general heuristics and specific heuristics. General heuristics include heuristics that are valuable for experimenting regardless of the context of application. A heuristic like “keep track of your experiments” is, for instance, always important. Specific heuristics include heuristics that are dependent on the context of application. “Choosing equal increments” between experiments, for instance, depends on the kind of hypothesis that the learner is looking for. It is a valuable heuristic when you are looking for a quantitative relation between variables, but when you are looking for a qualitative relation between variables it is not really necessary to use this heuristic. In this case it might be more useful to look at a range of values, also including some extreme values, than to concentrate on using “equal increments”.

The division between general and specific heuristics is reflected in the feedback that is given to the learners when they draw a graph. General heuristics are always used to assess the learner's experiments, and can always generate feedback. Specific heuristics are only be used to assess the learner's experiments if the learner fits a function on the experiments. Which of the specific heuristics are be used, depends on the kind of function. For instance, the 'equal increments' heuristic will not be used if the learner fits a qualitative function on the experiments.

The specific heuristics "identify hypothesis" can be said to represent the formal analysis that of the experiments that was used in the first version of the tool [16]. The first version of the tool checked whether the hypothesis could be identified based on the experimental evidence that was generated by the learner. It also checked whether this identification was proper. It did not check if the experimental evidence could also confirm the hypothesis. For instance, if the hypothesis is that two variables are linearly related, and only two experiments were done, at least one other experiment is needed for confirming this hypothesis. This extra experiment could show that the hypothesis that was identified is able to account for this additional experiment, but it could also show that the additional experiment is not on the line with the hypothesis that was identified based on the first two experiments. The "confirm hypothesis" heuristic takes care of this in the new tool.

5 A Learning Scenario

A learner working with the simulation can do experiments, decide whether to store the experiment in the tool or not. The tool keeps track of all these experiments and keeps a tag that indicates whether the learner stored an experiment or not. At a certain moment, the learner decides to draw a graph. The learner has to select a variable for the x-axis and for the y-axis, and press the button to draw a graph for these variables. At this point, the tool checks what 'type' of variables the learner is plotting, and based on this check the tool can stop without drawing a graph and present feedback to the learner, or proceed with drawing the graph. The first will happen if a learner tries to draw a graph with two input variables, since this does not make sense. Input variables are independent, and any relation that might show in a graph will therefore be the result of the changes that were made by the learner, and not of a relation between the variables. The tool will not draw a graph either when a student tries to draw a graph with an input variable on the y-axis, and an output variable on the x-axis. Unlike with the two input variables this could make sense, but it is common practice to plot the variables the other way around. In both cases the learner will receive feedback that explains why no graph was drawn, and what they could change in order to draw a graph that will provide more insight on relations in the domain.

If the learner selects an input variable on the x-axis, and an output variable on the y-axis, or two output variables the tool will proceed with drawing a graph, and will generate feedback based on the heuristics.

First, the general experimenting heuristics evaluate the experiments that the learner has performed. Each of the heuristics will compare the learner's experiments with the pattern (for an example see Table 2) that was defined for the heuristic. If necessary the heuristic can ask the tool to filter the experiments (f.i. only stored experiments). The feedback text is generated based on the result of this comparison, and returned to the tool. The tool temporarily stores the feedback until it will be presented to the learner.

Table 2. Example patterns for the 'equal increments' heuristic

Equal increments	If in a set of experiments in which the value for input variable on the x-axis changes, and the other input variables are kept the same There is no set of experiments in which the increment between the first and the second experiment is equal to the increment between the second and the third experiment Then remind the learner of the equal increment heuristic

The next step will be that the tool analyses the experiments using the same principles that were described in Veermans & van Joolingen [16]. Based on these principles the tool identifies sets of experiments that are informative for the relation between the input variable on the x-axis and the variable on the y-axis. For this purpose the experiments are grouped into sets in which all input variables other than the variable o the x-axis are kept constant. This will result in one or more sets of experiments that will be sent to the specific experiment heuristics, which will compare them with their heuristic pattern, and, if necessary, generate feedback text.

At this point the tool will draw the graph (see for example Figure 3). Together with the plots the tool will now present the feedback that was generated by the general experimenting heuristics. The feedback consists of the name of the heuristic, the outcome of the comparison with the heuristic pattern, an informal text that says that it could be useful to set up experiments according to this heuristic. The tool will provide information on each of the experiment sets that consists of the values of the input variables in this set and the feedback on the specific experiment heuristics.

If the learner decides to plot two output variables, it is not possible to divide the experiments formally into separate sets of informative experiments. Both output variables are dependent on one or more input variables, and it is not possible to say what kind of values for the input variables make up a set that can be used to see how the output variables are co-varying given the different values for the input variables. Some input variables might influence both output variables, and some only one of them. This makes it impossible to assess the experiments and the relation between the outputs formally. This uncertainty is communicated to the learners, warning them that they should be careful with drawing conclusions based on such a graph. It is accompanied by the suggestion to remove some experiments to get a set of experiments in which only one input variable is varied that than is the one that causes variation in the output variables. This feedback is combined with the feedback that was generated by the general experiment heuristics.

Learners can also decide to fit a function through their experiments, and if possible, a fit will be calculated for each of the experiment sets. These fits will be added to the graph, and additional feedback will be generated and presented to the learner. This additional feedback consists of a calculated estimation of the fit and more elaborate feedback from the specific experiment heuristics. The estimation of the fit

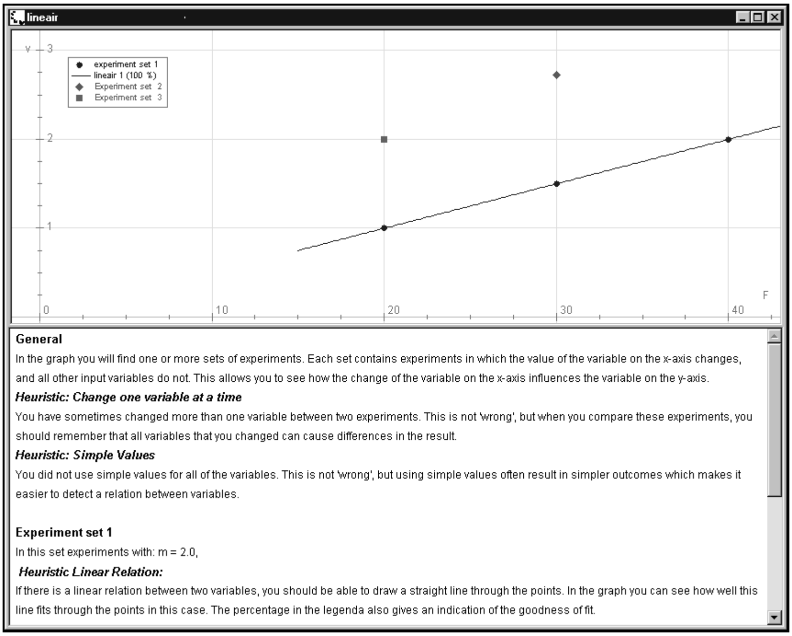

Fig. 3. Example of a graph with heuristic feedback based on the experiments in Figure 2

is expressed with a value on a scale ranging from 0% to 100%, with 0% denoting no fit at all, and 100% a perfect fit. The feedback that is generated by the specific experiment heuristics can be more elaborate when the learner fits a function, because the function can be seen as a hypothesis. This hypothesis allows a more detailed specification of the specific experimentation heuristics. The minimum number of experiments that is needed to be able to identify a function through the experiments can be compared with the actual number of experiments in each of the experiment sets. If the actual number is smaller than the required number this is used to generate feedback. The minimum number to confirm a hypothesis is the minimum number that can identify the hypothesis, plus one extra experiment that can be used for confirmation. Learners are also suggested to look at both the graph and the estimation of the fit to guide their decision on the correctness of the fit. At the same time one of the inductive discovery heuristic is used to suggest the learner to create a new variable that could help to establish a firm conclusion on the nature of the relationship.

6 Concluding Remarks About the Design of the Tool

The previous sections described the design of the tool for supporting hypothesis testing. The tool uses both formal and heuristic methods to analyze the experiments that learners perform in the process of testing a hypothesis, and, based on the result of the analysis, draws conclusions about the quality of the learners' hypothesis testing process. A learning scenario illustrated how the tool can support learners. It is not a learner-modeling tool, in the sense that keeps and updates a persistent model of the learner's knowledge, but is in the sense that it interprets the behavior of the learner, and uses this interpretation to provide individualized and contextualized feedback to the learner. The fact that tool uses both formal and heuristic methods, makes it broader in its scope than a purely formal tool.

In relation to the goal for the tool and the constraints it can be concluded that:

1. *The tool can support testing hypotheses and drawing conclusions.* Sorting the experiments into sets that are informative for the relation in the graph, drawing these sets as separate plots, generating feedback on experimentation, and generating feedback that can help the learner in the design and analysis of the experiments, supports hypothesis testing. Drawing separate plots, and presenting an estimated fit for a fitted function supports drawing conclusions.
2. *It leaves room for the learners to explore.* The tool leaves learners free to set up their own experiments, to draw graphs, and to fit relations through these graphs, thus leaving room for the learners to explore the relation between variables in the simulation.
3. *It is able to operate within the context of the authoring environment.* The tool is designed as a self-standing tool, and can be used as such. It does not have dependencies other than a dependency on the central manager of the simulation model.

7 Evaluation and Possible Extensions of the Tool

The tool described in this paper has been implemented and used in a simulation environment on the physics domain of collisions. This environment has first been evaluated in a small usability study with four high school students, and later with 46 high school students from two schools. Only a few results related to the tool will be highlighted here, a more elaborate description can be found in [18]. The results show among others that use of heuristics in the learning environment lead to higher learning outcomes compared to the environment with the previous version of the tool. It also showed that learner's were able to set up proper experimentation with the support in the environment; that using graphs with the feedback correlated positively with learning outcomes, but also that some learners felt quickly ready to be without the feedback. A possible extension could therefore be to allow more freedom to the learner related to presentation of feedback about heuristics and to selection of experiments that should be included in analyses. Especially for proficient learners it might work better if they can decide that they don't need certain feedback anymore. What could also prove to be of additional value is to make the tool less domain independent. One could think of the possibility to allow for instance assignments to communicate to the experiment storage tool which heuristics should be used in analyses, and set parameters for the patterns of these heuristics. This would allow the designer to tailor the heuristics and the patterns more to the domain, and/or to learners that are going to work with the learning environment.

References

1. Bruner, J. S. (1961).The act of discovery. Harvard Educational Review, 31, 21-32.
2. Butler, D. L., & Winne, P. H. (1995). Feedback and self-regulated learning: A theoretical synthesis. Review of Educational Research, 65, 245-281.
3. Dewey, J. (1938). Logic: the theory of inquiry. New York: Holt and Co.
4. Glaser, R., Schauble, L., Raghavan, K., & Zeitz, C. (1992). Scientific reasoning across different domains. In E. De Corte, M. Linn, H. Mandl, & L. Verschaffel (Eds.), Computer-based learning environments and problem solving (pp. 345-373). Berlin: Springer-Verlag.
5. Joolingen, W. R. van, & Jong, T. de (2003). SimQuest: Authoring educational simulations. In T. Murray, S. Blessing & S. Ainsworth (Eds.), Authoring Tools for Advanced Technology Educational Software: Toward cost-effective production of adaptive, interactive, and intelligent educational software. Lawrence Erlbaum
6. Karasavvidis, I. (1999). Learning to solve correlational problems. A study of the social and material distribution of cognition. PhD Thesis. Enschede, The Netherlands: University of Twente.
7. Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. Cognitive Science, 12, 1-48.
8. Klahr, D., Fay, A. L., & Dunbar, K. (1993), Heuristics for scientific experimentation: A developmental study. Cognitive Psychology, 25, 111-146.
9. Kulkarni, D., & Simon, H. A. (1988). The processes of scientific discovery: The strategy of experimentation. Cognitive Science, 12, 139-175.
10. Langley, P. (1981). Data-Driven discovery of physical laws. Cognitive Science, 5, 31-54.
11. Qin, Y., & Simon, H. A. (1990). Laboratory replication of scientific discovery processes. Cognitive Science, 14, 281-312.
12. Sanders, I., Bouwmeester, M., & Blanken, M. van (2000). Heuristieken voor experimenteren in ontdekkend leeromgevingen. Unpublished report.
13. Schoenfeld, A. (1979). Can heuristics be taught? In J. Lochhead & J. Clement (Eds.), Cognitive process instruction (pp. 315-338). Philadelphia: Franklin Institute Press.
14. Schunn, C. D., & Anderson, J. R. (1999). The generality/specificity of expertise in scientific reasoning. Cognitive Science, 23, 337-370.
15. Tsirgi, J. E. (1980). Sensible reasoning: A hypothesis about hypotheses. Child Development, 51, 1-10.
16. Veermans, K., & Joolingen, W. R. van (1998). Using induction to generate feedback in simulation-based discovery learning environments. In B. P. Goetl, H. M., Halff, C. L. Redfield, & V. J. Shute (Eds.), Intelligent Tutoring Systems, 4th International Conference, San Antonio, TX USA (pp. 196-205). Berlin: Springer-Verlag.
17. Veermans, K., Joolingen, W. R. van, & Jong, T. de (2000). Promoting self directed learning in simulation based discovery learning environments through intelligent support. Interactive Learning Environments 8, 229-255.
18. Veermans, K., Joolingen, W. R. van, & Jong, T. de (submitted). Using Heuristics to Facilitate Discovery Learning in a Simulation Learning Environment in a Physics Domain.
19. Zachos, P., Hick, L. T., Doane, W. E. J., & Sargent, S. (2000). Setting theoretical and empirical foundations for assessing scientific inquiry and discovery in educational programs. Journal of Research in Science Teaching. 37, 938-962.