Self-directed learning in simulation-based discovery environments

Ton de Jong, Wouter van Joolingen, Janine Swaak, Koen Veermans, Renate Limbach, Simon King, Daniel Gureghian

Combining human and machine expertise for self-directed learning in simulation-based discovery environments

Ton de Jong, Wouter R. van Joolingen,
Janine Swaak, Koen Veermans, Renate Limbach
University of Twente, The Netherlands
Simon King, Daniel Gureghian
EDS-Ingévision, France

Abstract

SIMQUEST[1] is an authoring system for designing and creating simulation-based learning environments. The special character of SIMQUEST learning environments is that they include cognitive support for learners which means that they provide learners with support in the discovery process. In SIMQUEST learning environments we try to find the balance between direct guidance of the learning process and sufficient freedom for learners to regulate the learning process themselves. In this article we describe the basic mechanisms of the SIMQUEST learning and authoring environments. We describe the functionality authors have in providing the learner with guidance, and report on some of our experiences on how authors use these opportunities and learners employ the cognitive support they are provided with .

Introduction

In our educational systems, the instruction and learning process still is generally characterised by the traditional lecture, in which the teacher explains to the students rules and principles of the domain. There is now a general conviction that this traditional way of *expository teaching* is not optimal for training employees that the market requires and who need *deep*, *flexible*, and *transferable* knowledge. This need has led to new pedagogical philosophies in which *constructivism* is the key item. In this philosophy learners construct knowledge themselves, and the learning process is characterised by placing a high responsibility into the hands of the learner instead of the teacher. Simulations are extremely suited for this type of learning since they encourage *discovery learning*, learners experiment and construct knowledge as 'scientists': they provide the simulation with input, observe the output, draw their conclusions, and go to the next experiment (Van Joolingen & De Jong, 1997). Experience and studies, however, show that learners are not always capable of handling their own learning process. In De Jong and Van Joolingen (1997) we have summarised a large number of studies that have shown a wide variety of problems learners may experience in discovery learning. In summary we can say that learners show problems with

---

[1] SIMQUEST is currently under development in the **SERVIVE** project. This project is sponsored by the EC in its Telematics programme as project ET1020. **SERVIVE** is the follow-up of the SMISLE project, also sponsored by the EC. In this article we report on our experiences as we have gathered them in the SMISLE project, and as we currently have them in the **SERVIVE** project.

all processes characteristic of discovery learning such as stating hypotheses, designing experiments, interpreting data and regulating the learning process (monitoring and planning).

In the EC sponsored SERVIVE project we develop an authoring system, named SIM-QUEST, that supports the creation of discovery learning environments. The SIMQUEST system is a follow-up of the SMISLE environment (see De Jong et al., 1994; De Jong & Van Joolingen, 1995), still following a similar learning and authoring philosophy (Van Joolingen et al., 1996). The learning philosophy holds that in SIMQUEST applications we aim to avoid learning problems by introducing cognitive support for the learner. This support consists of several 'learning tools': Learners may ask for small exercises (*so-called assignments*) that help them plan their actions and that can point them to specific phenomena; while experimenting learners can ask for background information in the form of definitions, relations to the real world etc. (this can be any kind of multi-media material); the simulation model can be presented to the learner in small steps that increase the model in complexity (so-called model progression); learners have tools that help them to monitor what they have been doing in a simulation session, that help them replay simulation sessions, compare outcome series, and make sound interpretations of the data; and, finally, also learners will have tools that help learners to compose and check hypotheses.

One of central questions we have in the project concerns the level of control that can be placed in the hands of the learner or that should be taken by the system. In this article we concentrate on two characteristics of SIMQUEST learning environments that are most relevant for this question: the nature of assignments, and the timing and obligation for learners to use these assignments.

Assignments as a support for discovery learning

The educational function of assignments

One of the paramount problems of learners in the discovery learning process is the regulation of their learning behaviour. Obviously, in self-directed learning environments the demand on regulative capacities of learners is larger than it is in traditional lectures. Planning and monitoring are central to regulation. Unsystematic planning and monitoring in simulation based discovery environments is a wide spread phenomenon and is, for example, reported by Lavoie and Good, 1988, Simmons and Lunetta (1993), Shute and Glaser (1990), Schauble, Glaser, Raghavan, and Reiner (1991). Veenman, Elshout, and Meijer (1997) who followed learners in a number of simulation based discovery environments report an effect of metacognitive abilities on the discovery learning process and learning results. Charney, Reder, and Kusbit (1990) and Teodoro (1992) claim that subjects have considerable problems with setting goals for themselves. Supporting the learner in the planning and goal setting process was already taken up by Showalter (1970) who used *questions* as a way to guide the learner through the discovery process. His questions focused the learners attention to specific aspects of the simulation (see also Zietsman & Hewson, 1986). Tabak, Smith, Sandoval, and Reiser (1996) have added such questions with the aim of setting goals in a biological simulation. White (1984; 1993) helped learners to set goals in a simulation by introducing *games* that ask learners to reach a specific state of the

simulation. In the SMISLE and SimQuest learning environments we have introduced the mechanism of *assignments* to help learners in their goal setting behaviour.

Assignments in SimQuest learning environments

Authors using the SimQuest authoring environment are offered templates for different types of assignments. Characteristic for all assignments is that they present the learner with a specific task. In SimQuest we distinguish five different types of assignments.

The first one, *do-it assignments*, give the learner the general assignment to explore the simulation model. As such, they do not advise the learner into a specific direction. The only thing authors can do is to put the simulation in a specific state (or more states from which the learner may select). Other types of assignment that can be created with SimQuest are more directive for the learning behaviour.

Investigation assignments ask learners to investigate the relation between two or more given variables. After exploring the simulation learners may select an alternative from a list of predefined alternatives (a more complex investigation assignment can be specified by asking the learner to select all the correct alternatives from the list of propositions given) and feedback is given. This feedback can be direct feedback on the alternative chosen (in the form of any type of multimedia content), or following a selected alternative the learner can be directed to, for example, another assignment.

Explicitation assignments always have an initial state or sets of initial states for the simulation associated with them, the role of the learner being to run the simulation (with these different initial sets) and to observe the impact on the simulation. Learners are then presented a set of propositions that describe the phenomenon or phenomena observed, and they are asked to select the correct alternative(s). Figure 1 gives an example of an explicitation assignment from a SimQuest environment in the physics domain of collisions.

In *specification assignments* the learner has to predict the values of certain variables when the associated simulation stops. The values predicted by the learner are allowed a deviation from the absolute values as specified by the author in either absolute or relative terms. The author also specifies when the simulation stops by assigning values to variables. When these values are reached the simulation stops.

Finally, in *optimisation assignments* the learner has to vary the simulation's variables' values so that the constraints specified by the author are not broken and a target specified by the author is reached. Figure 2 shows such an assignment from a SimQuest environment in the physics domain of motion.
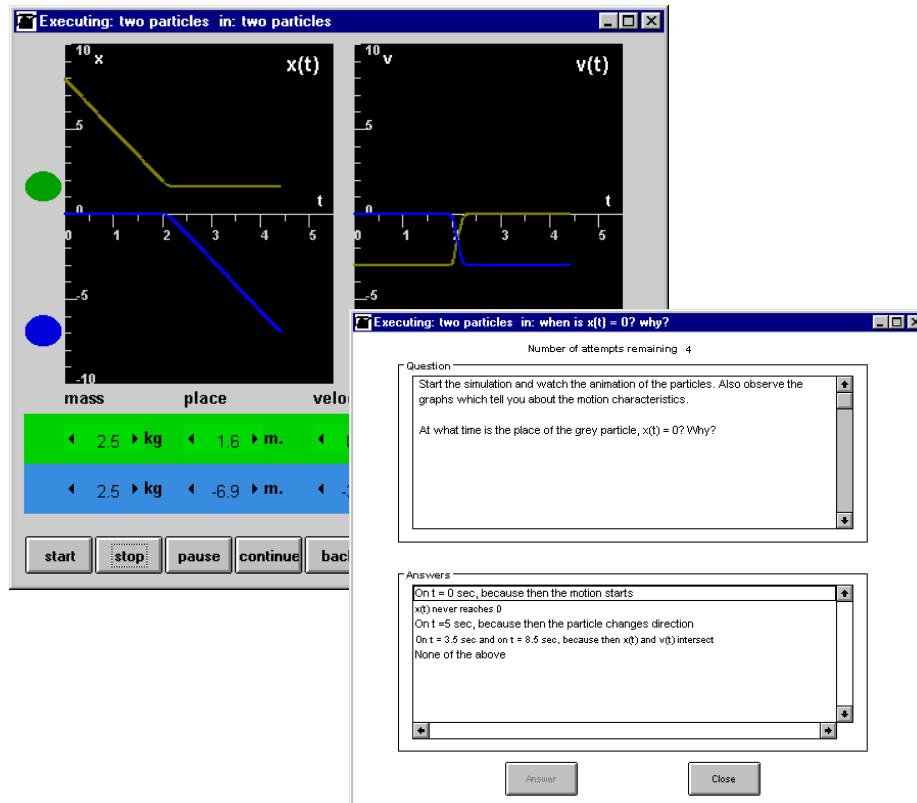
Figure 1. Explicitation assignment from a SIMQUEST learning environment on collisions

Learner and author experiences

We offer authors the possibility for creating assignments so that they may help the learner in the planning of the discovery process. Assignments may help the learner to decide on what to do next and may also help (by presenting the right variables and by putting the simulation in relevant states) to cover all relevant aspects of the model to be taught. We do now have experience with a number of environments being created by authors (teachers) from university and vocational levels. In a number of cases we have carefully evaluated the authoring process (see e.g., Kuyper, De Hoog, & De Jong, 1997). From these evaluations it became clear that authors also see assignments as an important instructional tool, and they use the whole variety of available types of assignments. There are, however, two deviations from using assignments in the way we, as the development team, intended to them to be used. The first one is that authors see assignments as the central mechanism in the learning process whereas we originally saw them as a support means that a learner might use if the self-directed discovery process got stuck. A second deviation is that sometimes authors tend to use assignments as intermediate tests, telling learners first to discover the rules and then use assignments to see if they understand them.
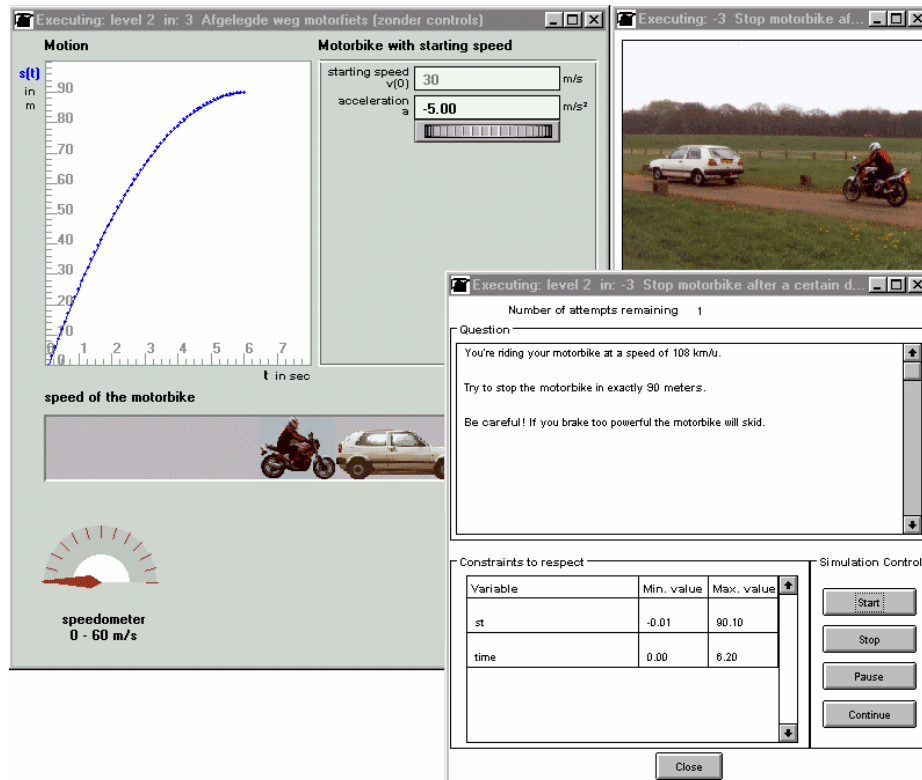
Figure 2. Optimisation assignment from a SIMQUEST learning environment on motion

In a number of empirical studies we have been able to see the influence of assignments on the learning process. Here we report on three studies, each covering a different domain from physics. These are transmission lines (De Jong, Härtel, Swaak, & Van Joolingen, 1996), collisions (De Jong et al., 1997) and harmonic oscillations (Swaak, Van Joolingen, & De Jong, 1997). One of the critical observations we did in these studies was that learners saw assignments as 'the' guidance for their discovery process. In cases where they were not forced to do any assignments (see also in the next section on 'control') they made the maximum use of them and it was frequently the case that learners saw the completion of all available assignments as the goal of their work ("I am done, I have completed all assignments"). In fact, students were right in recognising the importance of assignments. In the studies mentioned above we could estimate the effect of assignments by comparing the same simulation environments with and without assignments. Overall, in conditions where assignments were present learning gains (if measured as 'intuitive knowledge', see Swaak & De Jong, 1996) were higher than in conditions where assignments were not available.

Control over the environment

Structuring the discovery process

Whereas single assignments help learners to plan their immediate actions, several studies have indicated that learners also have problems in the overall planning of their learning process. For example, Glaser, Schauble, Raghavan, and Zeitz (1992) found that successful discoverers make plans over experiments, whereas unsuccessful learners concentrate on local decisions. In the literature we find several attempts to lead the learner through an overall sequence of actions. These actions quite often concern steps in the overall discovery process. For example, Lewis, Stern, and Linn (1993), Njoo and De Jong (1993a; 1993b), White (1993), and Shute and Glaser (1990) are examples of studies where learners are led (in a more or less compulsory way) through a fixed sequence of actions (e.g. devise an hypothesis, design an experiment etc.). Some studies that have compared structured and unstructured environments gave beneficial effects of structuring. Other studies (e.g., Veenman & Elshout, 1995) could not find these effects, or could only find these effects for specific groups of subjects. In summary, existing studies are rather inconclusive on the effects of structuring the environment. The differences in outcome can possibly be attributed to the differences in how the environment was structured, but also to differences in characteristics of the domains involved and of the characteristics of the learners.

Control structure in SIMQUEST

In SIMQUEST learning environments we use a control mechanism that has a distributed character (see also Van Joolingen & De Jong, 1996). Each instructional measure has an internal state. All instructional measures can assume the states "disabled", "enabled", or "active". Specific instructional measures can add more values for the state variable. For instance, assignments also can have the state values "failed" or "succeeded", which describes the results of the learner's interaction with the assignment. The complete state of the learning environment is defined by the collection of all states of the instructional measures. Instructional measures also contain a set of rules. These rules determine the change of state. For instance, a rule may say that when the state of an assignment becomes "failed", another instructional measure must change state to "active", which means that this instructional measure will display itself and is ready for interaction with the learner. Rules can be attached to any change of state. Typically, there will be rules which arrange the learning environment (i.e. open up the appropriate simulation interface, display an introductory text etc.) and rules which respond to the behaviour of the learner (like the activation of an explanation on assignment failure or success). A special state value is that of "enabled". This means that the learner will see the instructional measure on a list and is able to inspect a description of it. The learner can select and activate instructional measures that are enabled. In this way, learners can take initiative in the learning interaction by choosing for themselves *if* they need support from the enabled instructional measures and *when* they need it. The rules inside the instructional measures will ensure that the learning environment itself remains consistent.

The control structure in SIMQUEST is dedicated for making learning environments in which the responsibility for choosing 'actions' is distributed over the learner and the environment, meaning that the initiative for activating instructional measures (e.g., assignments or explanations) can depend on both the preferences of the learner and on the internal state of the learning environment (see Van Joolingen & De Jong, 1996).

Learner experience and author experience

Our experience with authors using the control structure in SIMQUEST (and SMISLE) is that they tend to structure the environment quite strictly and force the learner to go from assignment to assignment. In this way, authors, use only a limited part of the functionality that the SIMQUEST environment offers them for structuring the learning environment. One of the reasons for authors to work in this way is that they believe that this very constrained structure is necessary for an optimal learning process. For this reason we conducted a study in which we created two environments (called CIR-CUIT), one in which the learner could always choose freely from all assignments and was never forced to perform one (all assignment were always 'enabled'), and one in which learners were led from assignment to assignment ('exiting' one assignment 'activated' another assignment). The domain was from a physics topic, electrical circuits, and the learners came from a middle vocational training college. In the 'free' environment we had 21 subjects, of which 20 worked with the 'constrained' environment. A full description of the study can be found in Swaak, Blokhuis, Gutierrez, and López (1997). The main conclusion from the study is that constraining the students in their freedom in using support by forcing them to complete all assignments at one level of model progression before proceeding to another level did not make much difference. Neither the post-test scores nor the interaction data and navigation data as measured with the log-files, identified major differences between experimental groups. The explanation put forward for the absence of major differences between the conditions is that just the presence of assignments is sufficiently directive. Although the students in the free condition were completely free in doing or not doing assignments, they completed on the average 14 of the 19 assignments, and despite this number is significantly lower than the 17 completed assignments of the structured condition it is not substantially lower. The number of simulation runs and the amount of explanations consulted did not differ significantly for the two conditions. In addition, if we consider the navigation measures, we see that the numbers of model progression switches differ for the two experimental groups, but that the time spent at each of the model progression levels did not show any statistically significant differences. In a second study (see also Swaak et al., 1997), also in the field of electricity and in middle vocational training we introduced the same two types of environment, one free and one constrained (this environment was called ElectricA). This time we gathered in-depth data of 10 students (five in each condition). From the questionnaire data we can conclude that, though the contents of the two versions of ElectricA were exactly alike, they were perceived differently by the students. The 'free' and the 'constrained' versions of ElectricA had precisely the same outlooks, contained the same assignments, explanations, feedback etc., and only differed in the amount of freedom given to the learners. Nevertheless, most ratings from the students of the 'free' condi-

tion were more favourable than the ratings from the 'constrained' condition. The students from the 'free' condition gave a higher quality rating to the several features and parts of ElectricA, appreciated the instructional measures more, reported to have less problems on keeping track of what they had learned, and on knowing how to proceed. Furthermore, students of the 'constrained' condition indicated to be more frustrated by the possibilities of the environment, reported to need more information on how to operate ElectricA, and more background information. When students of both conditions were directly asked to report on the extent to which they felt constrained, the ratings of the groups were nearly similar. Yet, students' perception of freedom, was higher in the 'free' condition as compared to the 'constrained' condition.

Conclusion

In this article we have discussed how in the SIMQUEST authoring environment we have provided authors with functionality for creating learning environments with a mixed initiative for learner and system. We have also seen that both authors and learners use these facilities in a way that does not follow its intentions and reduces simulations learning environments to restricted environments where the predominant objective seems to be to perform a fixed sequence of assignments. We think that this phenomenon is partly due to the natural tendency of authors and learners to return to something familiar: a book with exercises at specific points. We expect that part of this will disappear over time when authors and learners have more experience with simulation based discovery environments. In the meanwhile, however, we have taken two initiatives to improve the situation. One initiative concerns the learner and addresses a kind of support for regulating discovery behaviour, associated with assignments, that intends to stimulate more self directed discovery learning. The other initiative concerns the author and holds a 'pedagogical advisor' in the SIMQUEST environment, that authors may consult and that may help them in constructing more open environments. We will now elaborate a little on these two developments.

One of the reasons for learners to use assignments the way they do, is that the feedback currently given to assignments is mostly in the form of an explanation to the several alternatives presented in the assignment. Currently we are developing a tool that can be used to generate feedback that is more discovery process oriented and that analyses the experiments a learner has been doing in relation to the alternative chosen in the investigation assignment. Figure 3 gives an example of the type of feedback that can be generated. This development is related to the development of a so-called "monitoring tool" which allows the learner to store, inspect, replay, and compare experiments. In order to enable this new feedback mechanism we had to extend the control structure in SIMQUEST that we described before. In the description given, the state of the learning environment depends solely on the state of the instructional measures. In the case of explicit support on the discovery process itself this is insufficient, since most parts of the discovery process are reflected in the interaction of the learner with the simulation. For registering this we introduced the "watchdog" or "daemon". A daemon monitors the state of the simulation and changes its instructional state once a certain, pre-specified simulation state has been reached. In this way the daemon transfers events from the "simulation domain" into the "instructional domain". At the moment we have three kinds of daemon: one "timer" which measures

the time that has passed since its activation and exits when a certain amount of time has elapsed, one that monitors the immediate state of the simulation and exits when a pre-specified state has been reached, and one that monitors (and analyses) the experiments performed by the learner. The latter is an example how learner modelling can be integrated within the system. Daemons can be created to monitor the experiments learners perform, the hypotheses they state, and the way they respond to questions. As daemons set an instructional state, the environment can react to these small learner models. This introduces an agent-like means of learner modelling where small daemons watch aspects of the learner behaviour. As daemons can be turned on and off during a session, the sophistication of the learner model can be adapted to the actual needs in a given situation.
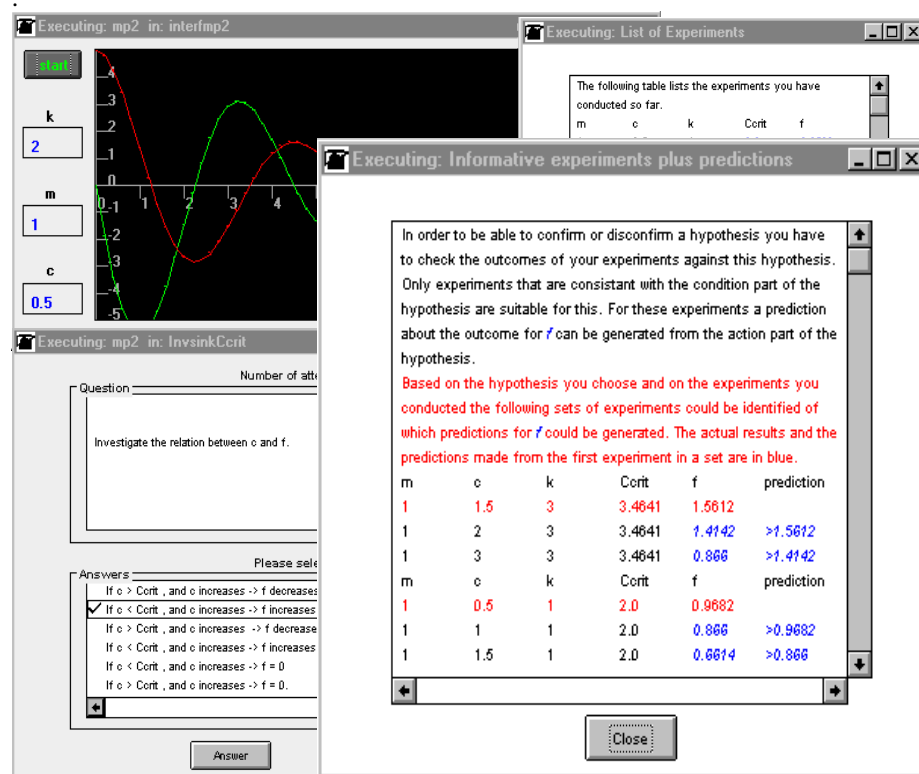

Figure 3. Discovery process feedback to an investigation assignment

For the authors we now came to realise that quite often they lack specific knowledge on how to design discovery learning environments. For this reason we will provide authors in SIMQUEST with a so-called advice tool. The advice tool is an elaborate hypertext system with textual and graphical information on how to design simulation based learning environments. The advice tool is accessible through a main window or in a context sensitive way through the element of an application that is under development. For example, when editing an assignment the author has direct access to advice on assignments. In the advice tool there are two dimensions. One is the topic on

which advice can be given about (e.g., model progression, or assignments) and the other dimension concerns a classification based on questions an author may have. Here we distinguish:

- *What is?* Tab sheets in the 'what is?' category give a definition of for example specification assignments, or explanations;
- *Example*. Tab sheets in this category show an example of for example instructional support, e.g., an example of a video explanation can be given;
- In the *considerations* category we include all kinds of instructional reasons for making use of specific instructional measures, and also e.g., considerations about learner characteristics, curriculum, and context characteristics..
- The *background* category gives information about studies we find in the literature.

Figure 4 gives an overview of the advice tool. In the content of the tab sheets we try to use a user oriented writing style and stay as closely as possible to the daily practice of teachers.
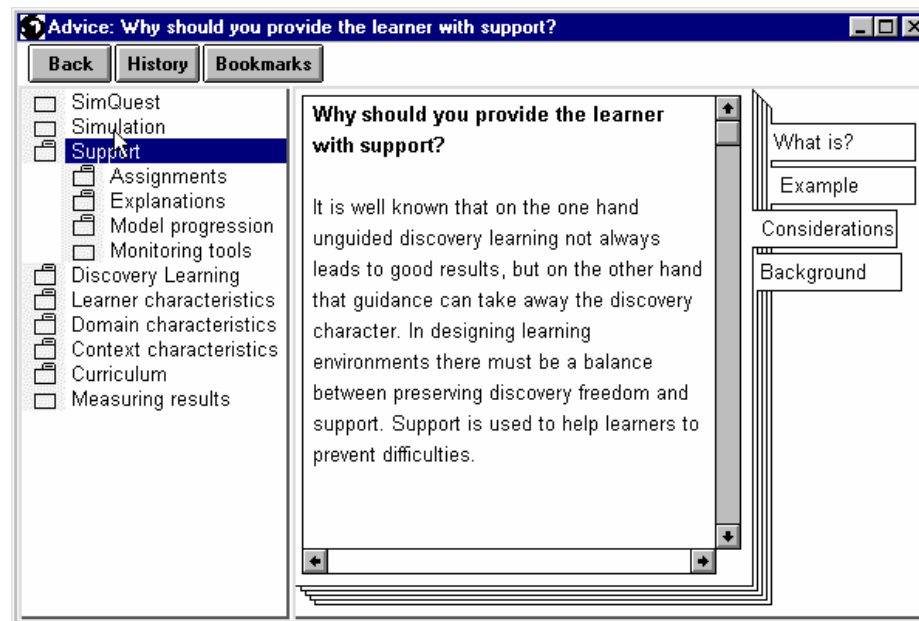

Figure 4. Example screen from the advice tool

By providing the learners with tools that directly aim at stimulating central discovery processes (in addition to the learning tools we already had available) and by providing authors with good information and examples on how to create simulation based discovery environments we hope to work towards a situation in which learning is not directed by a single agent, but is the result of combined expertises of system and learner.

References

Charney, D., Reder, L., & Kusbit, G.W. (1990). Goal setting and procedure selection in acquiring computer skills: A comparison of tutorials, problem solving, and learner exploration. *Cognition and Instruction*, *7*, 323-342.

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, Germany: Springer-Verlag.

de Jong, T., van Joolingen, W., Scott, D., de Hoog, R. , Lapied, L., Valent, R. (1994). SMISLE: System for Multimedia Integrated Simulation Learning Environments. In T. de Jong & L. Sarti (Eds.) *Design and production of multimedia and simulation based learning material* (pp. 133-167). Dordrecht: Kluwer Academic Publishers.

de Jong, T., & van Joolingen, W.R. (1995). The SMISLE environment: Learning with and design of integrated simulation learning environments. In P. Held & W.F. Kugemann (Eds.) *Telematics for education and training* (pp. 173-187). Amsterdam: IOS Press.

de Jong, T., Härtel, H., Swaak. J., & van Joolingen, W. (1996). Support for simulation-based learning; the effects of assignments in learning about transmission lines. In A. Díaz de Ilarazza Sánchez & I. Fernández de Castro (Eds.), *Computer aided learning and instruction in science and engineering* (pp. 9-27). Berlin: Springer Verlag.

de Jong, T., Martin, E., Zamarro J-M., Esquembre, F., Swaak, J., & van Joolingen, W.R. (1997). Support for simulation-based learning; the effects of assignments and model progression in learning about collisions. Manuscript submitted for publication.

de Jong, T., & van Joolingen, W.R. (1997). Discovery learning with computer simulations. Manuscript submitted for publication

van Joolingen, W.R., & de Jong, T. (1996). Design and implementation of simulation-based discovery environments: the SMISLE solution. *Journal of Artificial Intelligence and Education*, *7*, 253-277.

van Joolingen, W.R., & de Jong, T. (1997). An extended dual search space model of learning with computer simulations. *Instructional Science*, (in press).

van Joolingen, W., King, S., Alusse, A., Blankenburg, J., Celdran, M., Gamradt, T., Gureghian, D., Kuyper, M., Martinez, H., Paulsen, S-O, & Scott, D. (1996). Developing the SIMQUEST authoring environment. In: T. de Jong (Ed.), *Designing integrated computer simulation environments for discovery learning* (pp 71-97). Enschede: University of Twente.

Kuyper, M., de Hoog, R., & de Jong, T. (1997). Using WWW technologies for acquiring user requirements in distributed software development. Manuscript submitted for publication

Lavoie, D.R., & Good, R. (1988). The nature and use of predictions skills in a biological computer simulation. *Journal of Research in Science Teaching*, *25*, 335-360.

Lewis, E.L., Stern, J.L., & Linn, M.C. (1993). The effect of computer simulations on introductory thermodynamics understanding. *Educational Technology*, *33*, 45-58.

Njoo, M., & de Jong, T. (1993a). Exploratory learning with a computer simulation for control theory: Learning processes and instructional support. *Journal of Research in Science Teaching*, *30*, 821-844.

Njoo, M., & de Jong, T. (1993b). Supporting exploratory learning by offering structured overviews of hypotheses. In D. Towne, T. de Jong & H. Spada (Eds.), *Simulation-based experiential learning* (pp. 207-225). Berlin, Germany: Springer-Verlag.

Schauble, L., Glaser, R., Raghavan, K., & Reiner, M. (1991). Causal models and experimentation strategies in scientific reasoning. *The Journal of the Learning Sciences*, *1*, 201-239.

Showalter, V.M. (1970). Conducting science investigations using computer simulated experiments. *The Science Teacher*, *37*, 46-50.

Shute, V.J., & Glaser, R. (1990). A large-scale evaluation of an intelligent discovery world: Smithtown. *Interactive Learning Environments*, *1*, 51-77.

Simmons, P.E., & Lunetta, V.N. (1993). Problem-solving behaviors during a genetics computer simulation: beyond the expert/novice dichotomy. *Journal of Research in Science Teaching*, *30*, 153-173.

Swaak, J., Blokhuis, V., Gutierrez, M., & López, M. (1997). *Evaluating simulation discovery environments with learners.* **SERVIVE** deliverable D3.2. Enschede: University of Twente.

Swaak, J., & de Jong, T. de (1996). Measuring intuitive knowledge in science: the development of the what-if test. *Studies in Educational Evaluation*, 22, 341-362.

Swaak, J., van Joolingen, W.R., & de Jong, T. (1997). Support for simulation-based learning; the effects of model progression and assignments on learning about oscillatory motion. Manuscript submitted for publication

Tabak, I., Smith, B.K., Sandoval, W.A., & Reiser, B.J. (1996). Combining general and domain-specific strategic support for biological inquiry. In C. Frasson, G. Gauthier & A. Lesgold (Eds.), *Intelligent Tutoring Systems* (pp. 288-297). Berlin, Germany: Springer-Verlag.

Teodoro, V. D. (1992). Direct manipulation of physical concepts in a computerized exploratory laboratory. In E. de Corte, M. Linn, H. Mandl & L. Verschaffel (Eds.), *Computer-based learning environments and problem solving* (NATO ASI series F: Computer and Systems Series) (pp. 445-465). Berlin, Germany: Springer-Verlag.

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

Veenman, M.J.V., Elshout, J.J., & Meijer, J. (1997). The generality vs domain specifity of metacognitive skills in novice learners across domains. *Learning and Instruction*, *7*, 187-209.

White, B.Y. (1984). Designing computer games to help physics students understand Newton's laws of motion. *Cognition and Instruction*, *1*, 69-108.

White, B.Y. (1993). ThinkerTools: causal models, conceptual change, and science education. *Cognition and Instruction*, *10*, 1-100.

Zietsman, A.I., & Hewson, P.W. (1986). Effect of instruction using microcomputers simulations and conceptual change strategies on science learning. *Journal of Research in Science Teaching*, *23*, 27-39.