Providing Support in Learning from Computer Simulations: Interface Aspects [1]

Ton de Jong

Department of Social Science Informatics, University of Amsterdam, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands & Department of Philosophy and Social Sciences, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands[2]

Abstract: In contrast to computers assisted instruction programs (traditional and 'intelligent') *exploratory learning environments* offer the learner possibilities that go beyond the capabilities of a human teacher. One form of exploratory environments is *computer simulations* which provide the learner the opportunity to learn in an active way, as is advocated by most modern teaching theories. However, it has also become clear that learning by means of simulations puts a high cognitive demand on the learner. Additional support might be needed if learning from simulations is to be effective. This paper analyses the situation and indicates how this support can be given by a computer learning environment taking into account four characteristics of instructional use of computer simulations: simulation models, instructional goals, (exploratory, discovery) learning processes and learner activity. The paper mainly describes one of the aspects of such a learning environment: *the learner interface.*

Keywords: Computer simulation, intelligent tutoring system, interface design.

Introduction

The instructional use of computer simulations: Traditional forms of CAI such as tutorials and drills try to mimic the experienced teacher and are merely used because of the unavailability of those individuals. However, CAI will never match the competent teacher and ITS is also not yet ready to take that position. Recent developments in the use of computers in education show a different direction. 'Environments' are created in which learners may play around and exercise their abilities and knowledge. One type of such an environment is *computer simulations*. Computer simulations have widespread use in education. A recent overview of the use of CAI in higher education in the Netherlands (de Jong, 1990), indicates that about 50% of the programs are indicated as being a simulation or a combination of a simulation with some other form of CAI.

As described elsewhere (de Jong, 1991) instructional use of computer simulations can be characterized by the following four attributes:

1. Part of this paper is based at research conducted in the project SIMULATE. SIMULATE is part of SAFE, a R&D project partially funded by the CEC under contract D1014 within the Exploratory Action of the DELTA programme. A substantive part of this paper reflects work done in collaboration with Robert de Hoog and Frits de Vries (University of Amsterdam). I also like to thank Lynda Hardman for correcting the English.

2. Ton de Jong is now at Department of Education, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands

a. *Presence of formalized, manipulable underlying models*
By computer-based simulation is meant that a phenomenon, a process, a system or an apparatus (or whatever it is that is being simulated) is *formalized* into a model and implemented as a computer program. This model may have a qualitative character, or a quantitative one (or both). It is essential that the output of the program is *calculated* or *inferred* from the implemented model as a reaction to input from the learner.

b. *Presence of instructional goals*
Second, the simulation has to be used to reach a certain *instructional goal.* These goals can be of different types: *conceptual knowledge, operational knowledge* and knowledge of exploratory learning itself, *knowledge acquisition knowledge* (see van Berkum & de Jong, 1991). Sometimes there is a combination of instructional goals such as in Smithtown (Shute & Glaser, 1990) where both the domain (in this case microeconomics) and scientific discovery skills are objects of instruction.

c. *Elicitation of specific learning processes*
Third the simulation must be used to invoke specific *learning processes* characteristic of exploratory learning (such as 'hypotheses generation' and 'model exploration'). The path to the learning goal thus leads through these learning processes.

d. *Presence of learner activity*
Fourth, there must be some level of *learner activity.* This means that the learner must actually *manipulate* something at the simulation, e.g., setting input variables and parameters, choosing output variables, attaching measuring devices etc.

These four characteristics *together* describe the instruction/learning situation. A closely related type of instruction/learning is *modelling* which shares the above mentioned characteristics with learning with a simulation but has the additional characteristic that learners are allowed to interfere with the properties of the underlying model.

Support in learning with computer simulations: The basic situation of exploratory learning with computer simulations is a learner who freely interacts with the model as it is incorporated in the simulation. This situation, however, is not always optimal, and support might be needed. We distinguish two kinds of support: directive and non-directive support. *Directive support* steers the learner in a certain direction, e.g., when certain parts of the underlying model are not accessible to the learner, or when the learner gets direct feedback and/or hints for directions to follow. The second kind of support, *non-directive support,* does not steer the learner in a certain direction, but helps with accomplishing what s/he would have done in a completely free exploratory environment.

Defining and designing support for learning with computer simulations is the topic of a project called SIMULATE. The aim of the SIMULATE project is to do the groundwork for the development of an *authoring system* that helps authors to create simulations embedded in an intelligent learning environment (so called *Intelligent Simulation Learning Environments (ISLEs)*). The project started with an inventory of possible elements of ISLEs and here a selection of what has been accomplished in this inventory is presented particularly concentrating on one of the ISLE components: the *learner interface.*

Learning with ISLEs

The interface aspect: The main function of the interface of future ISLEs is to enable the learner to interact with the ISLE in a meaningful way. In essence this means that the interface should enable the four characteristics of instructional use of computer simulations as defined at the beginning of this paper.

The simulation model: The first characteristic for instructional use of computer simulations we distinguished was the presence of an *underlying computational model.* Here we take a layered view on computer simulation and distinguish the following layers (which are almost the same as introduced by Zeigler, 1976): the system that is modelled (the object system), the model of the simulated system outside the machine, the representation of that model inside the machine, and the presentation of the model in the machine to the learner.

For the learner interface the fourth layer is the most important. The function of the interface in this respect is showing this model to the learner. Principally there are two ways in showing the domain model: a *covert* way in which input variables and output states are shown, and an *overt* way in which properties of the model as they exist in the machine are shown to the learner. The *covert* way of showing the domain model is essential to learning with computer simulations.

Van Joolingen & de Jong (1991) review literature and identify two main dimensions for classifying internal characteristics of models: qualitative vs. quantitative and continuous vs. discrete simulations. They stress the fact that internal models can be very complicated and as Hartog (1989) says:

> *"Apparently the student is not expected to build the same mathematical model that is implemented on the computer when the model consists of, say, 20 or even 200 differential equations" (p. 196).*

It is therefore argued that the model as it is represented in the computer should resemble the model that the student is to acquire. This is accomplished in for example QUEST (White & Frederiksen, 1987) and according to Hartog *qualitative simulations* form a good basis for making 'teachable' representations.

A second important aspect of the interface in relation to the underlying model is establishing a link between the first and the fourth layer. In this respect a classification based on a distinction between the kinds of object systems the model in the simulation is of relevance:

- models representing some physical process, object or system,
- models representing an artificial system, i.e., a man-made object,
- models representing an abstract or hypothetical system.

An interface aspect closely related to this distinction is *fidelity level.* The correspondence between the real world model and the model in the simulation creates the basis for the fidelity level of the simulation. Although sometimes the feeling is that a high fidelity level is always desirable, it can also be assumed that full reality might be too overwhelming for (novice) learners (Reigeluth & Schwartz, 1989) or does not allow for insightful learning (Towne, Munro, Pizzini, Surmon, Coller & Wogulis, 1990).

Instructional goals: The second characteristic of learning with computer simulations is the presence of an instructional goal. The functions of the interface in relation to instructional goals are to communicate the instructional goal(s) to learners, to support learners in decomposing the instructional goal into subgoals and the final integration of these subgoals, and to help learners inspect instructional goals.

Communicating the instructional goal is a topic frequently discussed in books on designing traditional CAI and can also be accomplished in written material next to the simulation (Shute & Glaser, 1990). For learning with computer simulations, however, instructional goals are frequently quite comprehensive and have to be set by the learner him/herself. If the latter is the case, the interface should enable the learner to state (sub)goals, to relate these, to pursue and combine them. Support for setting and integrating subgoals by the learner is hardly ever seen in instructional software. An example of help the learner can be offered in breaking down an instructional goal into subgoals, can be found in IMTS (Towne et al., 1990).

The study process: Interaction with a computer simulation activates certain learning processes related to exploratory learning. De Jong & Njoo (1990) distinguish two main classes of learning processes: *regulative (planning and control)* and *transformative* processes that both can be supported through the learner interface.

Planning and control support

Planning in an exploratory environment is a crucial aspect for keeping the learning and interaction process under control. A high demand is put on the self regulatory capabilities of learners and supporting these through the interface is therefore advisable.

Of course making a goal decomposition is a (very domain related) form of planning but planning also takes place at a more global level where the study process is subdivided into a number of *phases*. De Jong & Njoo (1990) distinguish four separate main phases in the learning process: *analysis, hypothesis generation, testing*, and *evaluation*. Dividing the learning process in these phases can be considered as planning at a high level. A possibility for support therefore is to lead the learner through these phases, possibly offering different 'windows' for each phase (Njoo & de Jong, 1991b).

Closely related to the aspect of *planning* is the aspect of *control* and *navigation*. One of the significant problems with exploratory learning environments is that learners can get lost somewhere in the process. Planning and tracking tools might help the learner in navigating through the simulation. If the learner is offered means (e.g., scratch or note pads) for laying down his/her intentions and plans, this might help the learner with keeping track of his intentions and actions. Alternatively, the system might provide the learner with a simple trace of his/her actions. These traces or scratchpads can also be utilized for helping the learner with *controlling* his/her study process. In one of our ongoing studies we are looking at ways to provide learners with *overviews* of their interaction with the simulation (de Jong, de Hoog, & de Vries, 1991).

Support for specific (transformative) learning processes

It is a general feeling that not much is known of the learning processes that take place during the exploratory study process (Lesgold, 1990). A list of specific learning processes as applicable to learning with computer simulations is given by de Jong & Njoo (1990). We will refer to the non-directive support the learner can be offered for performing these learning processes as '*learner instruments*'. This concept will be illustrated by discussing examples for the learning processes *hypothesis generation* and *prediction*.

One of the principal learning processes one likes to encourage in exploratory learning is *creating hypotheses* or *predictions*. It is known that creating hypotheses is a difficult process students often don't perform at all (Njoo & de Jong, 1991a) or not too well (Shute & Glaser, 1990). Also hypotheses may differ in a number of characteristics, such as *generality of scope*, and *precision* (Spada, Stumpf, & Opwis, 1989).

The learner interface might help the learner in expressing a hypothesis or prediction by offering a (dedicated) scratchpad or a spreadsheet. An example of this can be found in CIRCSIM-TUTOR (Kim, Evens, Michael & Rovick, 1989) an ITS in the domain of medicine, more specifically it treats problems associated with blood pressure. In CIRCSIM-TUTOR learners are posed with a perturbation of the cardiovascular system (e.g., "the atrial resistance is decreased to 50% of normal", see Kim et al., 1989). Subsequently they are asked to predict what will happen to seven components of the cardiovascular system. This prediction is made in a qualitative way (increase, decrease, steady) for three different moments in time. For making a note of this prediction learners are offered a 7 (components) x 3 (moment in time) spreadsheet. The spreadsheet names the components and the moments in time and is in this way rather restrictive. A slightly different approach is taken in PPT (Pathophysiology tutor, Michael, Haque, Rovick & Evens, 1989). Here learners can indicate a predefined hypothesis by a list of nested menus that each give a more specific list of hypotheses in the field of physiopathology. In Smithtown, a microworld for elementary microeconomics (Shute & Glaser, 1990), three menus are presented to the learner. The first one contains 'connectors' (if, then, as, when, and, the), the second one 'variables' (such as: price, population, income, interest rate), and the third one contains 'descriptors' (such as: increases, decreases, equals, is part of). By combining connectors with variables and descriptors, the learner can state a hypothesis (e.g., as interest rate increases, price increases). Four types of hypotheses instruments thus seem to exist:

- 'Empty' scratchpads that offer just a place on the screen to note down something (see for example Towne et al., 1990). These scratchpads merely offer the possibility of reducing working memory load;
- Structured scratchpads that offer the learner a certain structure;
- Predefined scratchpads that present elements of hypotheses and ask the learner to assemble an hypothesis (e.g., as in Smithtown);
- Predefined menus of ready made hypotheses (Plötzner, Spada, Stumpf & Opwis, 1990).

Dedicated scratchpads can be designed for other learning processes (model exploration, testing etc.) as well. These scratchpads may also offer an opportunity of gaining information from the learner that can be incorporated into the learner model.

Learner activity: A characteristic of learning with computer simulations is that learners can manipulate the simulation. They can set parameters, change input values etc. In discussing interface aspects a classification of learner activities made by van Joolingen & de Jong (1991) is followed:

a. *Defining experimental settings*
   The learner interface should provide 'physical' handles on the model, i.e., it must enable the learner to manipulate the model. It is important that the handles on the simulation do not frustrate the learner in executing this manipulation. The learner interface has to support the *choice of variables and parameters that will be changed* and also the *changing of the values themselves.*
b. *Collecting data*
   Quite a few simulations present only one type of data as a result of student input. No choice is left for the learner. However, a number of simulations offer the learner options for choosing what, how and when to measure. In a number of simulations the learner is free to attach 'measuring devices' (test equipment such as voltmeters etc.) at places that s/he likes. So s/he not only manipulates the input, but also determines him/herself where and how to obtain the output (see for example ELAB (Böcker, Herczeg & Herczeg, 1989)).
c. *Procedure choices*
   When the simulation is used to teach a procedure or a skill, not only the values of variables and parameters are important, but also *when* these values are entered. The sequence of entering values and asking for data becomes part of the instructional goal. Two possibilities exist here: First, the learner is offered a discrete list of possible actions to perform, and he chooses one of them. The list is a sublist of all possible actions available in the program and

selected by the author of the program (see e.g., Booth & Hames, 1987). Second, the learner has all possible actions in the program available during the interaction. A good example here is flight simulator. The learner has all input and output devices available all the time.

d. *Setting data presentations*
Quite a few simulations offer the learner the possibility of seeing the data resulting from running the simulation in different modes, for example in a graphical or tabular form. The learner may choose from the two possibilities and can easily change between them. A second (and widely used) example of influencing the data presentation is allowing a grid to be set to graphical output.

e. *Metacontrol*
Metacontrol over the simulation concerns activities of the learner that control the pace of simulation or that set constraints on the simulation. Constraints to the simulation are mostly set by an external agent (the tutor for example) but in some cases the learner might choose to explore a simulation under a specific constraint. Also, the learner might control simulation time which means slowing down or speeding up a simulation process.

Designing intelligent simulation learning environments

The preceding section indicated the basic functionality of the interface when a learner interacts with a simulation for which non-directive support is offered. The complete interface of an ISLE as to be defined within the SIMULATE project, however, also has to incorporate the possibility of showing system messages that reflect directive support (such as hints on what to do next (see for example IMTS, Towne et al., 1990)). Moreover, the interface should offer general user/system interaction devices such as quitting the session, or resizing windows, and possibly some tools such as general scratchpads/notebooks. Finally, there is a possibility that for updating information in the learner model the learner will be questioned directly. All these functionalities have to be organized in a meaningful way on the computer screen.

There seem to be two basic ways to functionally group input and output: according to the type of action involved or according to the object(s) of the action(s). The first categorization implies that actions are organized according to a number of generic action classes. An example forms the generic action class that consists of all actions that are related to help. Another example is all actions related to the lay out of the screen, such as resizing of windows. The second categorization groups actions according the type of object they are related to. An example of this categorization in simulations is a class called 'model(s)'. All permissible actions concerning the model(s) are grouped under this header, for example zooming in/zooming out, partial inspection etc. In both cases the basic problem is to find what one could call 'generic' classes, that is, classes that, in a way meaningful for the learner, group related actions or objects together. An overview as presented in this paper offers a basis for conceptually clustering (grouping) aspects of the interface. (see also de Hoog, de Jong & de Vries, 1991).

The overview presented here is just a part of a larger inventory of all possible elements of ISLEs. In the SIMULATE project this inventory activity has been continued in what has been called a *formalization* activity in which information as gathered in the inventory has been *extended* and *described* in a more formal way. The extension renders more complete and detailed lists of elements of ISLEs with information (through a task analysis) on how learners will interact with these specific elements (e.g., working with hypotheses scratchpads), see Schuttenbeld, Bulthuis, de Vries & de Hoog (1991). The formal description is introduced in order to have a parsimonious, unambiguous description that allows for communication between different components of an ISLE. The formalization activity is also a first step in the direction of creating an authoring tool for ISLEs in that it defines a *library* of building blocks of ISLEs an author may choose, specialize and instantiate (see de Jong, Tait, & van Joolingen, 1990).

The final SIMULATE system is envisaged to consist of a permanent *shell* that basically contains a set of interpreters, that is filled with *default information*, functionality of an ISLE that is always present, and *author generated information*. For providing this author generated information the author is supported by the above mentioned library of building blocks of ISLEs, and further by a set of rules that constitute *recommendations* of good instructional design and a *methodology*.

References

Berkum, J.J.A. van, & Jong, T. de (1991). Instructional environments for simulations. *Education & Computing, 6*, 305-359.

Böcker, H.D., Herczeg, J., & Herczeg, M. (1989). ELAB - An electronics laboratory. In: D. Bierman, J. Breuker & J. Sandberg (Eds). *Proceedings of the 4th International Conference on AI & Education*, 15-25. Amsterdam: IOS.

Booth, A.G., & Hames, B.D. (1987). *Protein Purification*. Oxford: IRL Press.

Hartog, R. (1989). Qualitative simulations and knowledge representation for intelligent tutoring. In: H. Maurer (Ed.). *Computer Assisted Learning. Proceedings of the 2nd International Conference ICCAL*, 193-213. Berlin: Springer-Verlag.

Hoog, R. de, Jong, T. de, & Vries, F. de (1991). Interfaces for instructional use of simulations. *Education & Computing, 6*, 359-385.

Jong, T. de (1990). *Ontwikkelingen in Computer Ondersteund Onderwijs in het Nederlands Hoger Onderwijs in de jaren 1988/1989*. [Developments in computer assisted instruction in the Netherlands for the years 1988/1989]. Eindhoven University of Technology: OCTO rapport 90/01.

Jong, T. de (1991). Learning and instruction with computer simulations. *Education & Computing, 6*, 217-229.

Jong, T. de, & Njoo, M. (1990). Learning and instruction with computer simulations: learning processes involved. In: E. de Corte, M.C. Linn, H. Mandl & L. Verschaffel (Eds). *Computer-Based Learning Environments and Problem Solving*. NATO ASI Series F, Vol. 84. Berlin: Springer-Verlag.

Jong, T. de, Hoog, R. de, & Vries, F. de (1991). *SUPER-MIDAS; A computer simulation for learning about decision support theory*. University of Amsterdam: SAFE/SIM/WPIII/UVA-rep/SUPER-MIDAS.

Jong, T. de, Tait, K., & Joolingen, W.R. van (1990). *Authoring for intelligent simulation based instruction: a model based approach*. Paper presented at the DELTA and Beyond conference, The Hague (The Netherlands), October 18-19.

Joolingen, W.R. van, & Jong, T. de (1991). Characteristics of simulations for instructional settings. *Education & Computing, 6*, 241-263.

Kim, N., Evens, M., Michael, J.A., & Rovick, A.A. (1989). CIRCSIM-TUTOR: An intelligent tutoring system for circulatory physiology. In: H. Maurer (Ed.). *Computer Assisted Learning. Proceedings of the 2nd International Conference ICCAL*, 254-267. LNCS 360. Berlin: Springer-Verlag.

Lesgold (1990). Tying development of intelligent tutors to research on theories of learning. In: H. Mandl, E. de Corte, S.N. Bennett & H.F. Friedrich (Eds). *Learning & Instruction; European Research in an International Context, Vol. 2.1*, 321-339. Oxford: Pergamon Press.

Michael, J.A., Haque, M.M., Rovick, A.A., & Evens, M. (1989). The pathophysiology tutor: a first step towards a smart tutor. In: H. Maurer (Ed.). *Computer Assisted Learning. Proceedings of the 2nd International Conference ICCAL*, 390-400. LNCS 360. Berlin: Springer-Verlag.

Njoo, M., & Jong, T. de (1991a). *Learning processes of students working with a computer simulation in mechanical engineering*. To appear in: the proceedings of the 1989 EARLI conference, Madrid (accepted).

Njoo, M., & Jong, T. de (1991b). *Support for learning with computer simulations: giving hints, supporting learning processes, and providing hypotheses*. Paper presented at the 1991 AERA conference. Chicago, April 3-7.

Plötzner, R., Spada, H., Stumpf, M., & Opwis, K. (1990). *Learning qualitative reasoning in a microworld for elastic impacts*. Research group on cognitive systems, Forschungsbericht nr. 59. University of Freiburg: Psychological Institute.

Reigeluth, C.M., & Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. *Journal of Computer-Based Instruction, 16(1)*, 1-10.

Schuttenbeld, H., Bulthuis, W., Vries, F. de, & Hoog, R. de (1990). Formalisation of user interface aspects for instructional simulations. In: T. de Jong & K. Tait (Eds). *Towards Formalising the Components of an Intelligent*

Simulation Learning Environment, 173-245. Eindhoven University of Technology: SAFE/SIMULATE report SAFE/SIM/wpI- II/EUT-rep/formal.final.

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

Spada, H., Stumpf, M., & Opwis, K. (1989). The constructive process of knowledge acquisition: student modelling. In: Maurer, H. (Ed.). *Computer Assisted Learning. Proceedings of the 2nd International Conference ICCAL*, 486-500. LNCS 360. Berlin: Springer-Verlag.

Towne, D.M., Munro, A., Pizzini, Q.A., Surmon, D.S., Coller, L.D., & Wogulis, J.L. (1990). Model-building tools for simulation-based training. *Interactive Learning Environments, 1*, 33-50.

White, B.Y., & Frederiksen, J.R. (1987). *Causal Model Progressions as a Foundation for Intelligent Learning Environments* (Report No. 6686). Cambridge, MA: BBN Laboratories.

Zeigler, B.P. (1976). *Theory of Modelling and Simulation*. New York: John Wiley and Sons.