

## **Coping with complex environments: the effects of providing overviews and a transparent interface on learning with a computer simulation**

TON DE JONG,† ROBERT DE HOOG AND FRITS DE VRIES

*University of Amsterdam, Department of Social Science Informatics, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands*

*(Received 4 February 1992 and accepted in revised form 26 April 1993)*

Computers are used in increasingly complex environments for increasingly complex tasks. An example hereof is the use of computer simulations in instruction. Simulation offers an environment in which learners have to extract information from the system and must construct their knowledge themselves. This requires a high level of control for the learner over the (complex) environment. The present study investigates the influence of two representation aspects of simulation environments on the way of interacting with a simulation and on resulting test performance. The first aspect is giving learners additional *navigation* support by providing them with separate overviews of input and output. The second aspect concerns the *type of interface*: a conversational interface vs. a direct manipulation interface. Subjects had to learn about a theory of decision support with the use of one of four versions of basically the same simulation. In a control condition subjects were directly confronted with the simulation model in the form of a formula. Results showed that navigation support did not raise the subjects' scores. To the contrary, subjects receiving navigation support tended to have lower test performance. Subjects who received navigation support made fewer iterations during the simulation than the other subjects and the number of iterations was related to test performance. An explanation for their low scores might be that the navigation support distracted the subjects from their main task: learning about the model by manipulating the simulation. The direct manipulation interface was successful in increasing the number of changes to model variables. This, however, neither increased nor lowered the subjects' test performance. As expected, the direct manipulation interface resulted in far more efficient learning compared with the conversational interface.

### **1. Introduction**

Computers are used to assist people in performing tasks of an increasingly difficult, complex and comprehensive nature. In order to help users cope with this situation they are offered all kinds of support (context sensitive help, extra support functionality such as clipboards, etc.) and also, using the application is made easier by introducing "transparent" interfaces.

An example of such a complex situation is the use of computer simulation for computer assisted learning (CAL). Learning through simulation is a demanding task since simulation invites learners into an active engagement with the (model of the) domain and into frequent interaction with the program. Learners are encouraged to use *learning processes* (e.g. "hypothesis generation" and "data interpretation")

† Present address: The University of Twente, Department of Education, PO Box 217, 7500 AE Enschede, The Netherlands.

that involve an active structuring of knowledge, and to get involved in *learner activities* by manipulating aspects of the simulation (de Jong, 1991). This appeal to exploratory learning processes and student activity means that learners may get involved in ineffective and inefficient behaviour (Shute & Glaser, 1990) or, on the other hand, become rather inactive (Njoo & de Jong, 1991, 1993). Instructional simulation, therefore, provides a suitable environment for research on how to assist subjects working on a complex task in a computer environment.

In the present study we evaluated the effect of two distinct types of support for subjects involved in simulation-based learning. First, we supported learners in their navigation process by providing them with overviews of their input and the system's output, and second, a direct manipulation interface (replacing a traditional conversational interface) was offered to learners in order to simplify the interaction and to increase the impact of the feedback from the simulation results.

### 1.1. EXPLORATORY LEARNING PROCESSES AND SUPPORT

Learning with a computer simulation means that a learner is involved in a process of *exploratory* or *scientific discovery* learning. This is a complex process of which, despite some very influential studies (e.g. Langley, Simon, Bradshaw & Zytkow, 1987; Simon, 1989), still too little is known (Lesgold, 1990). Klahr and Dunbar (1988), who observed subjects interacting with a simulation of a simple device, distinguished two basic approaches in exploratory learning. In one approach subjects take a hypothesis as their starting point and try to find evidence that supports or refutes it. In the other approach subjects work the other way round and collect data from which they induce regularities. These types of subjects are denoted as *theorists* and *experimenters* respectively. Similar findings have been reported by Shute and Glaser (1990). Both studies found that subjects have considerable problems with the process of exploratory learning and exhibit many shortcomings. Klahr and Dunbar, for example, found that subjects tend to stick to hypotheses despite disconfirming evidence.

Njoo and de Jong (1991, 1993), who observed subjects learning with a simulation on control theory, make a distinction between *transformative* processes and *regulative* processes. Transformative processes are cognitive processes that more or less help directly in generating knowledge. Examples are "hypothesis generation" and "data interpretation". Regulative processes help to manage the study session with processes such as "monitoring", "planning", and "verifying". De Jong and Njoo (1991), who compared observations of learners in a simulation environment and subjects learning from text, found that regulation is far more important when working with the simulation. Monitoring ("where have I been?") and planning ("where will I go?") together can be called *navigation*. It is well known from studies on exploratory environments such as hypermedia (see e.g. Hammond, 1989) that navigation poses large problems to users, which makes navigation a good candidate for supplying *support*.

We may distinguish two main types of support: directive support (or guidance) and non-directive support. We define *directive support* as an explicit instructional measure aimed at steering the learner in a certain direction, whereas *non-directive support* covers assisting the learner in the complex process of exploratory learning without pushing him or her in a specific direction. Directive support comprises

actions such as giving hints, Socratic dialogue, and corrective feedback (van Berkum & de Jong, 1991), and non-directive support includes assistance such as hypothesis scratchpads, and goal decomposition trees (de Hoog, de Jong, & de Vries, 1991). In exploratory learning the emphasis is on the self-regulation of the learner. Certainly, non-directive support will interfere less with this self-regulatory character of exploratory environments than directive support.

## 1.2. LEARNER ACTIVITIES AND INTERFACE DESIGN

Learning with computer simulations not only means that learners follow active learning processes but also that they have to be active in their interaction with the simulation program. We will use the term learner activity for the "physical" interactions of a learner with a simulation. We can distinguish a number of different forms of learner activity. First, of course, the learner must be able to specify experimental settings such as setting the values of parameters and variables. The second type of learner activity is to make a choice for the next step in the simulation process; a critical activity when the simulation concerns a procedure. The third type concerns the choice of which data to select (for example by attaching instruments in the simulation). Fourth, the learner may have the possibility of making a choice for data presentation (e.g. setting a grid); and finally, the learner may have meta-control (such as control over pace and direction of the simulation). Some of these learner activities are closely related to some of the learning processes mentioned above. It is also evident that not all types of simulations comprise all opportunities for learner activities.

Learner activities are affected through the learner interface. At present two metaphors govern the interface world. In *conversational interfaces* the user normally uses a keyboard and types commands for telling the computer what s/he wants to happen. Typically these commands are somewhat similar to, but still far away from, natural language. Examples of this conversation style are: command languages, query languages, form fill-in systems, menu selection systems (Gaines & Shaw, 1986a,b; Shneiderman, 1987).

A way of interaction in which objects can be manipulated directly can be found in the so-called direct manipulation metaphor (Hudson, 1987). Shneiderman coined the term "direct manipulation" and its key ideas are: "visibility of objects and actions of interest, rapid reversible incremental actions, replacement of command language syntax by direct manipulation of objects" (Shneiderman, 1987: p. 180). Objects on the screen are representations of real world objects, and actions in the most simple form are touching and moving with the mouse. This means that a user can execute a task with a minimum of syntactic knowledge and only superficial semantic knowledge of the internal computer language. The user can concentrate fully on the semantics of the objects and actions of the task (Shneiderman, 1987). In general we might therefore expect *more efficient* task performance with direct manipulation interfaces as compared with conversational ones.

In the context of "learning and instruction" the direct manipulation interface also has the advantage of *immediate feedback* because learners may immediately observe the results of their actions. This is generally assumed to be beneficial for learning (Anderson, 1983). A recent study by Te'eni (1990) confirmed that a direct manipulation interface enhanced performance through direct, task-embedded,

feedback. He also found that direct manipulation interfaces enhanced the efficiency of task performance. The efficiency effect was found for both simple and complex tasks, whereas the better performance with direct manipulation was more pronounced for complicated tasks.

### 1.3. RESEARCH QUESTIONS

In the present study we evaluated two ways of assisting the user (learner) in coping with the complex task of learning from a simulation.

The first is introducing support for the learning process of *navigation*. The process of navigation is considered an important process in each more or less complex (computer) task in which a number of subtasks have to be performed. Results on support for navigation in a learning environment therefore may be generalized to other environments. We introduced a *non-directive support* measure in the form of providing *overviews*. Overviews show learners a diagram of their input history or (separately) of the output of the simulation over a number of sessions. As a result of providing overviews we expected learners to gain insight into their exploratory behaviour which they could use for planning their behaviour more consistently, consequently we expected them to display more structured exploration behaviours, and thus gain better learning results.

As a second assistance to the learner we introduced a direct manipulation interface that we labelled the "*dashboard interface*". Through the direct access to variables in the model, the dashboard interface was expected to stimulate the exploratory behaviour of learners and thus help to surmount the problem of passiveness as it is sometimes observed. This should be reflected in a higher activity level compared to the more traditional "*conversational*" interface. Based on the already mentioned study by Te'eni (1990), we expected better learning results with the dashboard interface based on direct, task-embedded, feedback and also a higher efficiency in learning with this interface. Additionally, a direct manipulation interface makes returning to a previous state in the simulation very easy, and can therefore, partly, act as a replacement of navigation support.

## 2. Method

### 2.1. THE DOMAIN: DECISION SUPPORT THEORY

The domain that was used was *decision support theory*. We used as a starting point an existing decision support system called MIDAS (see e.g. Breij, de Hoog & Zandvliet, 1989, for a description). MIDAS is written in Turbo Pascal and runs on IBM PCs. MIDAS stands for Multi-attribute Individual Decision Assistance System and is meant to assist people in choosing among alternatives by assessing and combining values for the attributes of these alternatives (e.g. which car to choose). The underlying model is not simple, yet very structured (see e.g. Bronner & de Hoog, 1984). Concepts used in MIDAS are:

- *Alternatives* the user may choose;
- *Attributes* that are used in assessing the alternatives;
- *Positions* for each alternative on each attribute;

- *Importance* the user attaches to each attribute;
- *Acceptability* of the difference between the actual position and the ideal point for each attribute.

These items are combined in a decision function that assigns an overall value (score) to each alternative, so that alternatives can be ranked. A number of decision functions exist. For MIDAS this decision function is given in the basic model in MIDAS and can be represented in equation (1):

$$S(a_j) = 100 - \left[ \sum_{i=1}^n w_i |a_{ji} - I_i|^{r_i} \right], \quad (1)$$

where:

- $A = \{a_1, a_2, \dots, a_m\}$ , set of alternatives;
- $K = \{i_1, i_2, \dots, i_n\}$ , set of attributes;
- $s(a_j)$  = score of alternative  $a_j \in A$ ;
- $i$  = attribute  $i \in K$ ;
- $a_{ji}$  = position of alternative  $a_j$  on attribute  $i$ ;
- $I_i$  = ideal position for a person on attribute  $i$ ;
- $w_i$  = weight/importance of attribute  $i$ ;
- $r_i$  = acceptability of the difference between  $a_j$  and  $I$  for attribute  $i$ .

The higher the score  $s(a_j)$ , the higher the preference for the alternative  $a_j$ .†

In words, we can list some important properties of the model in MIDAS. First, the model has a general structure: the additivity of the attribute terms, the transformed differences on each attribute, and the multiplicativity of weights. Second, the model has a number of variables: the ideal points, the positions of the alternatives, the weights, the acceptability, and as an output variable the score of an alternative. Key concepts are attribute independency, redundancy of attributes, double attributes, strongly peaked distribution of positions of alternatives on attributes, and dominance structure. The model used in MIDAS is a compensatory model, which means that a large difference between an ideal point and the position of an alternative on one attribute can be compensated by a small difference on another attribute. It is, however, important to note that when the *acceptability* for one attribute becomes very large, the model approaches a dominance model. What is important for the present study is that the domain involved has a number of (higher order) concepts that learners may discover through working with a simulation and that the domain is sufficiently complex to warrant a genuine learning task.

## 2.2. EXPERIMENTAL SET-UP

In the experiment subjects had to discover the properties of the model behind the simulation program MIDAS by working with (different forms of) the simulation. The aim was to comprehend the central concepts in the model, more specifically the roles of *ideal points*, *weights*, and *acceptability*. In fact, these were the variables that the subjects could manipulate in the simulation.

† For presenting results to users in the right hand side of the expression the factor 100 is introduced, in order to keep to the “natural feeling” that a *higher* score gives a *higher* preference. 100 is an arbitrary number in this case of course.

The specific topic we chose was a subject that is attractive to all subjects: holidays. Four alternative holiday destinations were introduced: Egypt, London, the Canaries and Norway. We fixed four attributes for which the three experimental variables (ideal point, weight and acceptability) could be set: *temperature*, *distance*, *level of activity*, and *culture/nature*. The position of the alternatives on these attributes were given to the subjects in advance (and they could consult these figures throughout the learning phase of the experiment). They could not assign or change these positions themselves.

We have termed the instructional system that we used SUPER-MIDAS.† In the experiment we introduced five conditions each comprising a different version of SUPER-MIDAS. Since we assumed that subjects were unfamiliar with the domain of decision support, all conditions started with an introduction to the topic. In one condition this introduction was only followed by displaying the formula underlying the decision support simulation (i.e. equation 1). This condition was included in order to appraise the effectiveness of simulation against a direct exposure to the model. One condition contained a “bare” simulation without any additional support. In two conditions navigation support was introduced, in one of them subject generated and in the other one program generated. In the final condition the interface of SUPER-MIDAS was changed into a direct manipulation interface. In detail these experimental conditions were:

### *I. Introduction and formula*

In the first condition, SUPER-MIDAS did not include a simulation part, it merely consisted of an introduction in which the different elements of the MIDAS model were discussed and illustrated with the example of buying a car. It contained one (small) exercise in which learners could change the values of weights and observe the effects of these changes on preference scores. This introduction part (without the equation) was also used as the introduction for the other experimental conditions. At the end of this introduction, subjects were given the equation underlying MIDAS.

### *II. Introduction and simulation and conversational interface*

In the second condition, SUPER-MIDAS provided the subjects not only with the introduction but also with a simulation in which they could manipulate the ideal points, weights, and acceptabilities. The interface was a conversational one, values of variables were changed after a prompt by the system and the simulation outcomes were shown in tabular form after the subject had indicated s/he wanted a simulation run.

### *III. Introduction, simulation, notation forms and conversational interface*

In the third condition, SUPER-MIDAS was essentially the same as in experimental condition II. The only difference was that subjects were now supported by specific (simple) notation forms that were meant to help them navigate through the simulation. These notation forms contained for each iteration a predefined table in

† SUPER-MIDAS is short for SUPportive Extension of Robert's MIDAS.

which the subjects could note their input for the different variables for each attribute, and a table that listed the alternatives and contained empty cells to record the scores that resulted from running the simulation. It also contained some space for writing down whatever they thought necessary. Subjects were not forced to fill in the overview tables at each iteration.

#### *IV. Introduction, simulation, overviews and conversational interface*

In the fourth condition, subjects could ask for program-generated overviews of their exploratory behaviour. In an overview, learners were shown their interaction in a graphical way, separately for input and output. For the input the subjects were shown a graph displaying which type of variable (weight etc.) they had changed at each preceding iteration. For the output side they were presented with a graph of the scores for each of the alternatives for each of the iterations they had done so far. Overviews, thus, did *not* give scores of output variables as a function of values of input variables. Overviews just displayed the general pattern of shifts between types of input variables and (separately) of the changes in the scores of the output variables over iterations.

#### *V. Introduction, simulation, and dashboard interface*

In the fifth condition, SUPER-MIDAS was essentially the same as in condition II, the only difference being a different, dashboard, type of interface. In this interface subjects could change the values of input variables by pointing the mouse at meters and dragging the pointer to the value they wanted to give to a specific input variable. At the same time (as with a dashboard) other meters showed the values of the output variables (in this case preference scores for holiday alternatives).

Figure 1 gives an excerpt of the conversational interface, Figure 2 presents an example of an output overview (an example of input overviews can be found in Figure 8) and Figure 3 shows the dashboard interface.†

```

Below your ideal points on the first two attributes are displayed

                (1) Temperature                (2) Distance
                cold(=1)                       short trip(=1)
                hot (=7)                        long trip (=7)

Ideal           4                               1

Do you want to change these ideal points?
Answer y(=es) or n(o)!
>

```

FIGURE 1. An excerpt from the conversational interface.

† The simulation program for the direct manipulation condition was written in the object-oriented language Smalltalk/V version 2.0 from Digitalk Incorporated (Smalltalk/V is a registered trademark of Digitalk Inc.). The framework of Dashboard, a demonstration application included in the Smalltalk/V package, has been effectively re-used for this experiment.

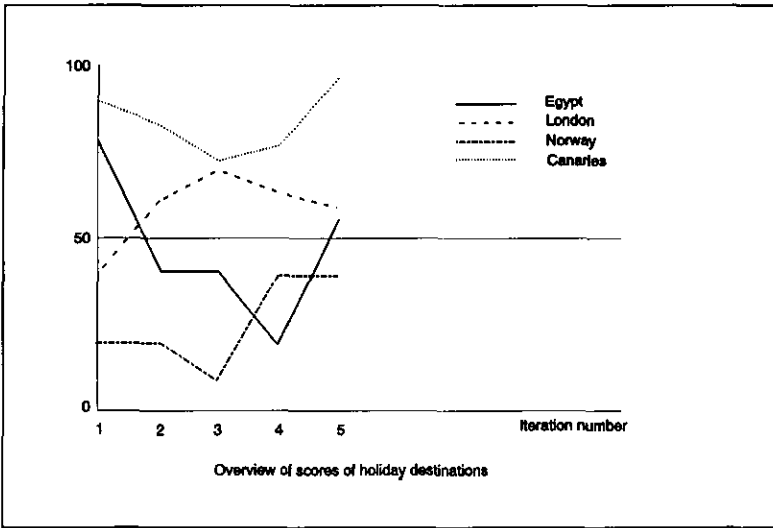


FIGURE 2. Example of output overview.

2.3. SUBJECTS

Subjects were 53 first year Psychology students. They participated in the experiment as part of their duty to act as a subject. Participation in this experiment was voluntary. The number of subjects differed between conditions due to the fact that some students didn't show up at the experiment. Subjects had not followed any specific courses on decision support theory as part of their curriculum and we assumed that they had no prior knowledge on this topic. The experimental set-up

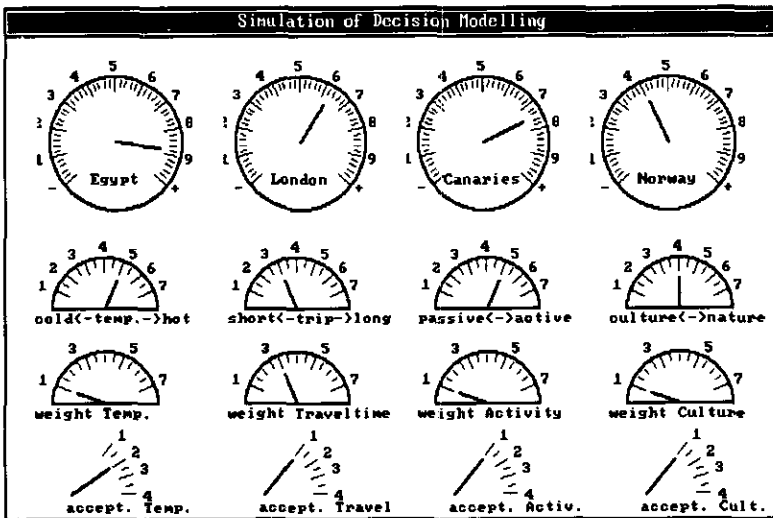


FIGURE 3. Screenshot of the "dashboard interface". The upper four meters represent output, the lower 12 meters can be manipulated to provide input to the simulation.



TABLE 1  
*Experimental design*

Group	Model presentation	Support	Interface	Number of subjects
I	Introduction and Formula			12
II	Introduction and Simulation		Conversational	10
III	Introduction and Simulation	Notation forms	Conversational	14
IV	Introduction and Simulation	Overviews	Conversational	8
V	Introduction and Simulation		Dashboard	9

together with the number of subjects in each experimental group is summarized in Table 1.

2.4. PROCESS AND PERFORMANCE MEASURES

Each session was logged and the log-files were used for assessing the exploratory behaviour of the subjects.

For the performance measure we created two types of questions. One type asked subjects quite directly for “definitional” knowledge. In the main type of question subjects were asked to predict what would happen in situations that were described, and for answering these questions correctly they needed insight into the key concepts of the domain (see Section 2.1). The test consisted of 17 multiple choice (3 or 4 alternatives) questions. Table 2 gives two examples of questions in the test. In order to promote their motivation for mastering the topic, it was announced at the start of the experiment that subjects would receive a reward of DFL 20 (approx. \$10 US) if they succeeded in answering 80% of the questions (or more) correctly.

TABLE 2  
*Examples of test items*

Given certain positions of alternatives on the attributes certain rank orders of alternatives are impossible.

This statement is:

- a. Correct
- b. Only correct if the positions of the alternatives on the attributes are not too different from each other
- c. Incorrect
- d. Only incorrect if the positions of the alternatives on the attributes are not too different from each other

The minimum score of an alternative can occur independently from the values of the weights and/or acceptabilities.

This statement is:

- a. Correct
- b. Only correct if all positions of an alternative on the attributes are also at their minimum
- c. Incorrect
- d. Only incorrect if all ideal points are different

## 2.5. PROCEDURE

Students who participated in the experiment first received a verbal and paper explanation of the experiment. They were instructed that the goal of the sessions was to understand the decision support model, and they were given five examples of test questions. The experimental session consisted of three parts: the introduction, the simulation (Group I received the underlying formula instead of the simulation) and the test. All parts were administered by the computer. Subjects were forced (by the program) to spend a minimum of 45 min in the tutorial plus simulation. After leaving the tutorial and entering the simulation, they could not return to the tutorial. Also, when starting the test they had no access to the tutorial and simulation. Subjects were not allowed to make notes (except, of course, for Group III who used notation forms). Subjects from Group III could not use their notes when taking the test.

The positions of the alternatives on the four attributes were fixed and given to the subjects in a hand out. Subjects had access to these figures while working with the simulation. Subjects could change ideal points, weights, and acceptabilities in the simulation. Since there were four attributes this means that subjects could change 12 individual variables in all. In the conversational interface (Groups II, III, and IV), subjects could change at each iteration the values for either the ideal points, weights, or the acceptabilities, thus leaving a maximum of four changes (related to the four attributes) at each iteration.† In the dashboard interface subjects could change each of the 12 variables at any time.

## 3. Results

For answering the research questions as expressed in Section 1.3 we looked at the test scores and the learning behaviour of subjects from the different experimental conditions and, if relevant, related behaviour with test results.

### 3.1. TEST SCORES

The average test scores over subjects within each of the five experimental conditions are shown in Figure 4.

The first conclusion from these data is that it is indeed a difficult subject for our subjects. Average scores of 60% correct are not really high taking into account that the items were (3 to 4 item) multiple choice questions.

A second conclusion concerns the difference between the experimental conditions. An  $F$ -test showed no significant overall effect of experimental condition on test scores ( $F_{4,48} = 1.7$ ,  $p = 0.16$ ), but there is a tendency that the groups that have received support (III and IV) have a *lower* score than the groups that received the introduction and the formula (I) or the introduction and a simulation (II and V). What is absolutely evident is that what we have called support, in this case, didn't help the subjects to raise their test scores. Also, in the dashboard interface group no higher scores than in the "conversational interface" group (II) were found. Both findings contradict our expectations.

† Except for the first iteration where students had to set the ideal points and where they were also able to assign values to the weights. If they preferred not to give values to the weights, students were informed that these were set to default values (of 1), as were the acceptabilities.

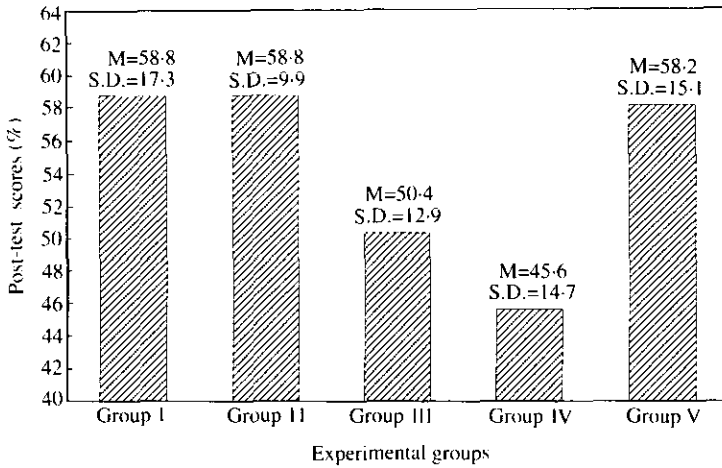


FIGURE 4. Average percentages of correctly answered questions in the test. M, mean; S.D., standard deviation.

### 3.2. INTERACTIONS WITH THE SIMULATION

#### 3.2.1. Overall activity level

One aspect of the interaction with the simulation is the *level of activity* of the subjects. In this study we measured activity level in a number of different ways.

Figure 5 depicts, for the Groups II, III and IV, the average number of iterations over subjects within each experimental group. This, of course, could not be measured for the “dashboard interface” group (Group V) since with the dashboard no real iterations exist. Subjects didn’t always make changes in the variables of the model when they went through an iteration. Figure 5, therefore, also depicts the (average) number of iterations in which changes to variables have been made.

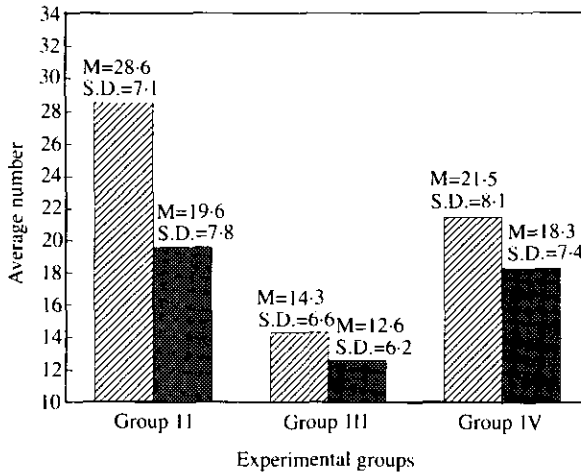


FIGURE 5. Average number of total iterations (▨) and average number of iterations in which variables have been changed (■) for the experimental groups II, III and IV.

The figure shows that the groups that received extra support (III and IV) show a *smaller number* of iterations. This holds especially for the subjects who received the note-taking forms (Group III). The overall effect of condition (group) was significant for the total number of iterations ( $F_{2,29} = 11.81$ ,  $p = 0.00$ ) and also for the number of iterations in which changes to variables have been given ( $F_{2,29} = 3.37$ ,  $p = 0.048$ ). A comparison of groups shows significant differences between the conditions II and III ( $t = 5.06$ ;  $p = 0.00$ ) and III and IV ( $t = -2.27$ ;  $p < 0.05$ ) for the total number of iterations. Similar significant differences appear when comparing the number of iterations at which changes have been made.

Another way to look at the activity level is counting the number of times a change to an individual variable in the model was made. In the conversational interface there was a maximum of four changes at each iteration. In this comparison we can also include the group who received the dashboard interface. For this group we registered the values of the input variables at each mouse click and thus we could determine the number of changes made.

Figure 6 gives the average number of changes made in each experimental condition over subjects.

As was expected, the number of changes made in the dashboard interface greatly exceeded the number of changes made in the conversational interface. The figure shows that this is reflected in an increase of the number of changes made by up to a factor of 10. The influence of experimental condition on number of changes made is, therefore, highly significant ( $F_{3,37} = 25$ ,  $p = 0.00$ ).

### 3.2.2. Interaction behaviour

The preceding section presented results on the *overall interaction activity*. This section will present some data on the *content* of this interaction. This pertains to the *magnitude of changes* made, the choice of *which variable to change* and the *patterns of interaction* in the simulation.

*Magnitude of changes.* One aspect of the exploratory behaviour of learners is the

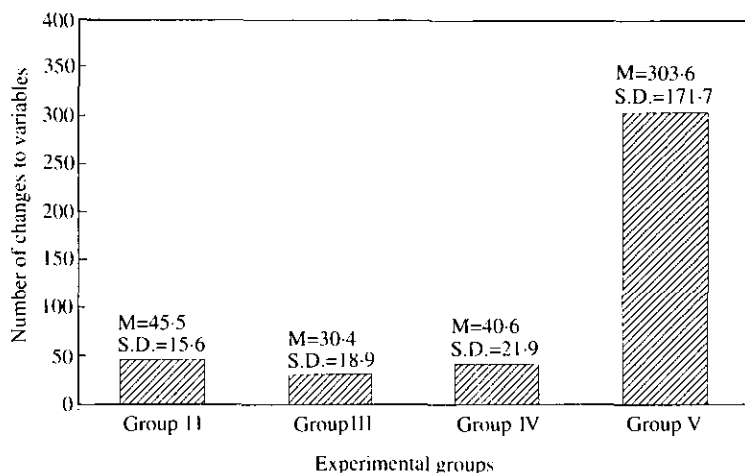


FIGURE 6. Average number of changes made to model variables.

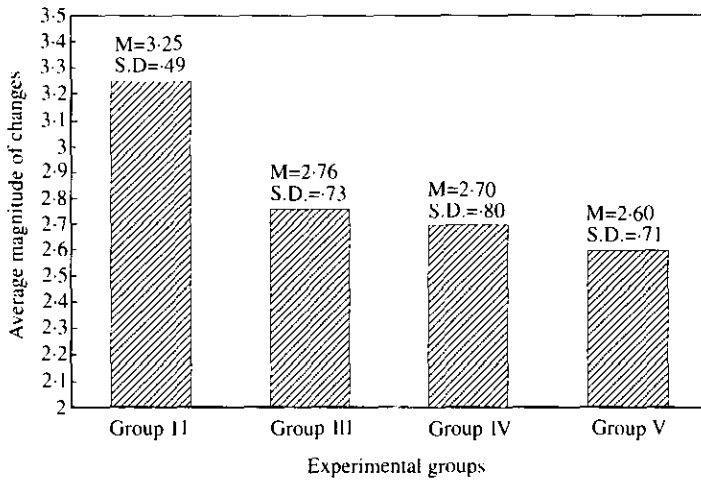


FIGURE 7. Average magnitude of change over all variables for the experimental groups II-V.

*magnitude* of the changes made. Figure 7 shows the average magnitude of changes to individual variables for the different experimental conditions over subjects. There is a trend that the average magnitude is smaller for the dashboard condition. This may indicate that the dashboard interface invited “tuning” behaviour, but there is no significant overall effect of treatment on average magnitude of change.

*Choice of variable to change.* The three variables that could be changed by the learners in this study do not have the same “intuitive understanding”. Clearly, the *acceptability* is the most difficult part of the model. On the other hand the *ideal point* is the most understandable variable and the *weight* is intuitively somewhere in between. What we find in the data for the Groups II, III and IV is that overall there are significantly more iterations in which the weight (mean = 6) and the acceptability (mean = 6.2) are changed than the ideal point (mean = 4) ( $t = -2.72$ ;  $p < 0.01$  and  $t = -3.14$ ;  $p < 0.01$  respectively). For the dashboard interface no significant differences between the types of variables were found. From these data we might, carefully, conclude that in manipulating the simulation subjects tend to concentrate on the more difficult aspects of the model which, however, doesn’t seem to be true for the direct manipulation interface.

*Interaction pattern.* Looking at the *interaction patterns* of subjects we may distinguish two main patterns of interaction which we have termed the *saw-blade* pattern and the *step* pattern. In the saw-blade pattern subjects make quick changes between the different variables, almost switching variable type at each iteration. In the step pattern subjects stay for a number of iterations with the same variable type. Figure 8 illustrates these two approaches by showing two interaction patterns taken from the data.

For the experimental Groups II, III and IV we have classified the interaction pattern. Here it is found that the saw-blade pattern is the most dominant one since 19 out of 32 subjects use this pattern. Seven subjects displayed a step pattern and the interaction patterns of the remaining six showed a mixture of these two prototypical patterns. For each subject a *pattern score* was determined as the quotient of the

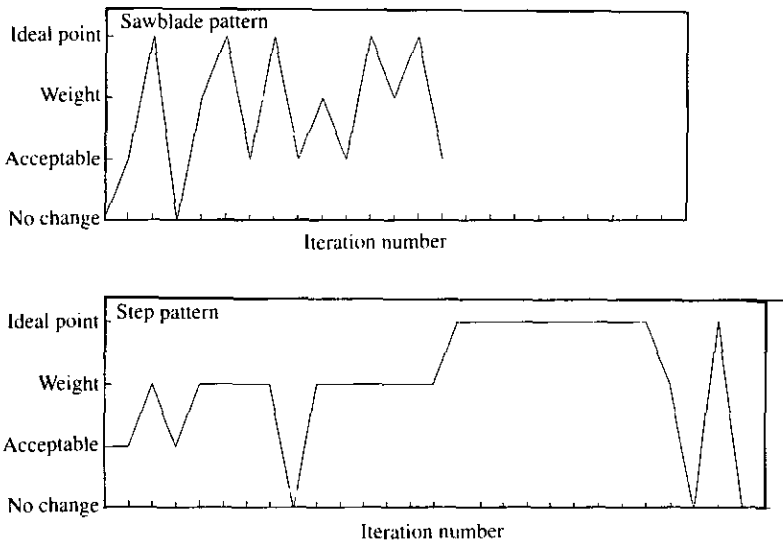


FIGURE 8. Examples of two prototypical interaction patterns.

number of times a step was made from changing one type of variable to another and the number of iterations in which a change to one of the model variables was made. This score ranges between 0 and 1 with a low score representing a step pattern, and a high score a saw blade pattern. An analysis of variance with experimental condition as independent variable and pattern score as the dependent variable shows that there is no difference between groups in interaction pattern ( $F_{2,29} = -2.31$ , n. s.). The average pattern scores in Groups II, III, and IV are respectively 0.68, 0.72 and 0.59. Our expectation that providing learners with overviews would influence their pattern of interaction is therefore not confirmed by the data.

3.2.3. *The relation between behaviour and test scores*

Table 3 presents correlations between the number of iterations made (for the total number of iterations, the number of iterations in which a change has been made,

TABLE 3  
Correlations between number of iterations and test score (*p*-values between brackets)

	Total iterations	Iterations with changes	Iterations for ideal points	Iterations for weights	Iterations for acceptability
Iterations with changes	0.75 (0.00)				
Iterations for ideal points	0.70 (0.00)	0.85 (0.00)			
Iterations for weights	0.63 (0.00)	0.88 (0.00)	0.68 (0.00)		
Iterations for accept.	0.55 (0.00)	0.78 (0.00)	0.49 (0.00)	0.49 (0.00)	
Test score	0.40 (0.02)	0.29 (0.10)	0.32 (0.07)	0.26 (0.15)	0.16 (0.37)

TABLE 4  
*Correlations between average magnitude of change in variables and test score  
 (p-values between brackets)*

	Overall average	Average ideal point changes	Average weight changes	Average acceptability changes
Average ideal point changes	0.84 (0.00)			
Average weight changes	0.89 (0.00)	0.71 (0.00)		
Average acceptability changes	0.84 (0.00)	0.66 (0.00)	0.68 (0.00)	
Test score	0.28 (0.09)	0.23 (0.15)	0.36 (0.02)	0.20 (0.21)

and the number of iterations in which a change has been made to a specific variable in the model) and the test score. Since iterations are involved, data only refer to the experimental conditions with a simulation and conversational interface. From the correlations we learn that, generally speaking, activity level expressed as number of iterations is correlated to performance. We also see that subjects who are active in changing one type of variable in the simulation are also active in changing the others.

We also looked at the number of changes to individual variables and have done this separately for the groups with the conversational and the group with the dashboard interface. Here we find again that correlations between the number of changes to each variable are high which implies that individuals who make frequent changes to one variable also often change the other ones. The correlations between number of changes and the test score, however, never reach a significant level.

Table 4 gives the correlations between the average magnitude of changes made by each subject over all model variables, average magnitude of changes to each individual model variable and the test score. The table indicates that there is a high correlation between the average magnitude of changes made to the three different variables. So, subjects who tend to make large changes, do so for each of the variables. Also, we see significant correlations between the average magnitude of the changes and test scores indicating that large changes are related to higher test scores.

Not only changes to individual variables can be related to test performance but also the interaction pattern of the subjects. It shows that there is a non-significant negative correlation between pattern score and the score at the test ( $r = -0.23$ , n.s.).

### 3.3. USE OF SUPPORTIVE MEASURES

In two of the experimental conditions subjects could create or ask for overviews that were aimed at supporting the navigation behaviour of the learners. We have already seen that this additional support didn't result in higher test scores. It is therefore interesting to see how this support was actually used by the subjects.

In experimental Group III, subjects were offered dedicated notation forms for writing down their personal overviews. It appeared that these notation forms were frequently used by the subjects. Half of the 14 subjects noted down everything from the screen on their notation forms and the other half made a selection of the data from the simulation. Six of the subjects used the available blank space on the notation forms to record relations as they observed them between input and output data, eight subjects didn't use the blank space at all. Of course numbers are too small to warrant a statistical comparison but we found that subjects who made a selection *and* noted down relations had an average (percentage) score at the test of 56, whereas the subjects that didn't make a selection *and* didn't use the blank space had an average test score of 38.

In experimental Group IV, subjects had the possibility of asking for system-generated overviews of input and output. Over all subjects, 45% of all iterations were followed by a request for an overview. We regard this as a very frequent use of the overview feature. There was a non-significant negative correlation ( $r = -0.28$ ) between number of times an overview was asked for and scores at the test.

#### 3.4. TIME ON TASK

Subjects in this study were forced to spend at least 45 min in the introductory part plus the simulation.† For the Groups II, III, IV and V we have looked at the time that the subjects spent in the simulation part. Overall the subjects used an average time in the simulation of 45 min with, however, considerable differences between conditions, as is shown in Figure 9.

Figure 9 shows that in the experimental conditions where additional support was given time spent in the simulation part was considerably longer than it was for the conditions without additional support. The differences between conditions are

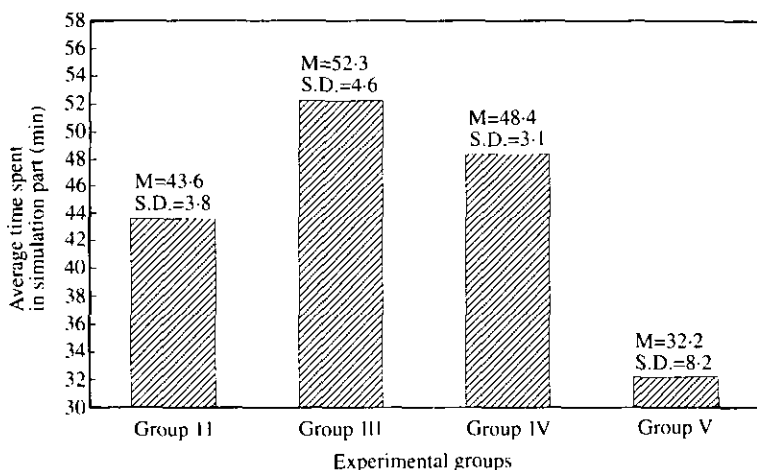


FIGURE 9. Average time spent in simulation part in minutes.

† Group I, that received no simulation, had to spend 45 min with the introduction and formula.



significant for all comparisons as measured by *t*-tests. In Figure 4 we have seen that this extra support has not resulted in higher test scores. This is also reflected in a negative correlation between time spent in the simulation and test score ( $r = -0.26$ ,  $p = 0.09$ ).

#### 4. General conclusion and discussion

In this study we have evaluated the effect of providing extra support to subjects learning from a computer simulation and also the influence of differences in interface design on both the learning outcome and the interaction process.

Our data show that providing navigation support in the form of overviews did not raise the subjects' scores at a post test as was expected. To the contrary, subjects in the experimental conditions who made use of either subject or system-generated overviews showed a tendency to have a *lower* score at the test. Also, subjects in these two conditions made fewer iterations in the simulation and we found that over all subjects the number of iterations was related to test performance. From this we may, speculatively, infer that the extra support distracted the learners from their main task: learning with the simulation itself. Our data indicate that subjects used the overview functionality fairly well. Also contrary to what we expected, overviews did not influence the interaction pattern of learners. If the interaction pattern of subjects needs to be changed, a more directive instructional measure than providing overviews is probably needed.

The dashboard interface group displayed, following the prediction, a far higher rate of manipulation. They manipulated variables at a level up to about ten times as high as in the conversational interface groups. However, the subjects from this condition did not score higher at the test than the subjects in the comparable conversational interface group. Apparently, the subjects from the dashboard interface condition did not profit from the direct feedback as was expected on the basis of Te'eni's (1990) results.

Next to the effectiveness of learning we examined the efficiency of learning that our subjects showed. Here, our data show that subjects from the dashboard interface condition learned very efficiently using considerably less time in the simulation. The two groups of subjects that received navigation support did not only have a (non-significant) lower test score than the other simulation groups, they also used more time for learning.

Another result of our study is that subjects who were directly exposed to the simulation model formula performed equally well at the test as subjects learning with a (plain) simulation. The condition in which learners were directly exposed to the model formula was included as a control condition. We expected subjects in simulation-based conditions to perform better on the test since this test largely aimed at measuring "deep", "insightful" knowledge. One explanation for the fact that these "simulation" subjects did not score better than the "formula" subjects could be that learning with a simulation was a novel task for most subjects involved in the experiment, which may have hampered their learning performance. An alternative explanation for these results could possibly be found in the work by Broadbent and colleagues on explicit and implicit learning (see e.g. Berry & Broadbent, 1984; Hayes & Broadbent, 1988). Following these ideas we could argue

that subjects in the simulation conditions (and especially the “dashboard” interface condition) would gain implicit knowledge that is not well captured by the written test. Berry and Broadbent (1984) have shown that implicit knowledge (ability to perform a certain task) can be uncorrelated to explicit knowledge (ability to answer questions about that task). A critical difference, however, between our study and the Broadbent studies is that in our case subjects’ main task was to learn about the model and not to control it and they therefore could anticipate upon the test, whereas in the Broadbent studies the test comes *unexpected* for the subjects. Even stronger, in our case, subjects were aware of the type of questions they would receive through the possibility of inspecting five example questions at the beginning of the experimental session. Finally, Berry and Broadbent (1988) found that a high “salience” of model variables creates a context in which implicit and explicit knowledge is not dissociated. In our experiment, variables were salient for the subjects through the introduction that they received and through the fact that all variables from the model were “visible” as input or output of the simulation.

The main lessons learned from this experiment are that exploration behaviour of learners can be influenced by supportive measures and type of interface and that aspects of exploration behaviour are correlated with level of test performance. There seems, however, to be an optimum level of activity in the simulation since the extreme higher rate of simulation use by the dashboard group did not result in higher test scores. We inferred that supportive measures may distract learners from working with the main task (the simulation), thus resulting in ineffective and inefficient behaviour. Clearly, support means an extra (new) task for learners that they also need to master before using it in a productive way. This is a conclusion that has impact beyond the situation of learning and instruction and also pertains, in our estimation, to all complex computer applications for which support is being designed. Finally, this conclusion may pose an argument against “one shot” experimental studies for evaluating interfaces.

This study was conducted within the SAFE/SIMULATE project partially funded by the DELTA programme of the European Communities.

## References

- ANDERSON, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- BERRY, D. C. & BROADBENT, D. E. (1984). On the relationship between task performance and associated verbalizable knowledge. *The Quarterly Journal of Experimental Psychology*, **36A**, 209–231.
- BERRY, D. C. & BROADBENT, D. E. (1988). Interactive tasks and the implicit–explicit distinction. *British Journal of Psychology*, **79**, 251–272.
- BREIJ, I., DE HOOG, R. & ZANDVLIET, L. (1989). Computer ondersteund onderzoek naar woonvoorkeuren. [Computer assisted research into living preferences]. In S. MUSTERD, Ed. *Methoden voor woning- en woonmilieubehoefte onderzoek*, pp. 30–46. SISWO.
- BRONNER, A. & DE HOOG, R. (1984). The intertwining of information search and decision aiding. *Acta Psychologica*, **56**, 125–139.
- VAN BERKUM, J. J. A. & DE JONG, T. (1991). Instructional environments for simulations. *Education & Computing*, **6**, 305–358.
- GAINES, B. R. & SHAW, M. L. G. (1986a). From timesharing to sixth generation: the

- development of human-computer interaction: 1. *International Journal of Man-Machine Studies*, **24**, 1-27.
- GAINES, B. R. & SHAW, M. L. G. (1986b). From timesharing to sixth generation: the development of human-computer interaction: 2. *International Journal of Man-Machine Studies*, **24**, 101-123.
- HAMMOND, N. (1989). Hypermedia and learning: who guides whom? In H. MAURER, Ed. *Computer Assisted Learning. Proceedings of the 2nd International Conference ICCAL*, pp. 167-182. Berlin: Springer Verlag.
- HAYES, N. A. & BROADBENT, D. E. (1988). Two modes of learning for interactive tasks. *Cognition*, **28**, 249-276.
- DE HOOG, R., DE JONG, T. & DE VRIES, F. (1991). Interfaces for instructional use of simulations. *Education & Computing*, **6**, 359-385.
- HUDSON, S. E. (1987). UIMS support for direct manipulation. *Computer Graphics*, **21**, 120-124.
- DE JONG, T. (1991). Learning and instruction with computer simulations. *Education & Computing*, **6**, 217-229.
- DE JONG, T. & NJOO, M. (1991). Learning and instruction with computer simulations: learning processes involved. 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. 411-429. Berlin: Springer.
- KLAHR, D. & DUNBAR, K. (1988). Dual space search during scientific reasoning. *Cognitive Science*, **12**, 1-48.
- LANGLEY, P., SIMON, H. A., BRADSHAW, G. L. & ZYTKOW, J. M. (1987). *Scientific discovery, computational explorations of the creative process*. Cambridge, MA: MIT Press.
- LESGOLD, A. (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, pp. 321-337. Oxford: Pergamon Press.
- NJOO, M. & DE JONG, T. (1991). Learning processes of students working with a computer simulation in mechanical engineering. In M. CARRETERO, M. POPE, R. SIMONS & J. I. POZO, Eds. *Learning and Instruction: European research in an international context: Volume III*, pp. 483-499. Oxford: Pergamon Press.
- NJOO, M. & DE JONG, T. (1993). Exploratory learning with a computer simulation for control theory: learning processes and instructional support. *Journal of Research in Science Teaching*, **30**(8).
- SIMON, H. A. (1989). The scientist as a problem solver. In D. KLAHR & K. KOTOVSKY, Eds. *Complex information processing*, pp. 375-399. Hillsdale, NJ: Lawrence Erlbaum.
- SHNEIDERMAN, B. (1987). *Designing the user interface, strategies for effective human-computer interaction*. Reading, MA: Addison-Wesley.
- SHUTE, V. J. & GLASER, R. (1990). A large-scale evaluation of an intelligent discovery world: Smithtown. *Interactive Learning Environments*, **1**, 51-77.
- TE'ENI, D. (1990). Direct manipulation as a source of cognitive feedback: a human-computer experiment with a judgement task. *International Journal of Man-Machine Studies*, **33**, 453-466.