# Cognitive tools for discovery learning

Wouter Van Joolingen

*International Journal of Artificial Intelligence in Education (1999), 10, 385-397*

# Cognitive tools for discovery learning[1]

**Wouter van Joolingen,** *Graduate School of Teaching and Learning, University of Amsterdam*[2]
*wouter@ilo.uva.nl*

*Based on an invited talk presented at ITS98 conference*

**Abstract.** Cognitive tools, defined here as instruments that support or perform cognitive processes for learners in order to support learning, can bridge the difference between open learning environments, like discovery learning environments and traditional supportive instructional environments. This article discusses a definition of the concept of cognitive tool and its use in learning. Two examples of cognitive tools for discovery environments are presented, and it is made clear how these tools can serve as hooks for anchoring intelligent instruction. Finally design issues for integrating cognitive tools in a discovery environment are discussed.

## INTRODUCTION

This article addresses the issue of supporting discovery learning processes. Discovery learning is seen as a promising way of learning for several reasons, the main being that the active involvement of the learner with the domain would result in a better structured base of knowledge in the learner as opposed to more traditional ways of learning, where knowledge is said to be merely transferred to the learner. Of course, a continuum exists from ‘pure discovery’ to ‘pure expository learning’, and in fact neither of the extremes will be found in educational practice. The same continuum can be found in computer environments for learning, from (expository) learning environments presenting information to the learner to open-ended environments like computer simulations or web based inquiry environments giving much freedom to the learner. This continuum gets a meaning at the moment that we want to interactively support the learner in gaining knowledge from a learning environment. Where in a system controlled, expository environment, the question is which information to present based on the interaction with the learner, in a discovery environment the question is how to assist the learner in selecting and interpreting information from the learning environment. The shift in locus of control from system to learner forces us to rethink the means of supporting the learner. Hence we also have to think differently of the role and location of system intelligence as an aspect of the learning environment.

In this article the role of learner support by a discovery learning environment and the role for intelligent system components in such environments will be investigated. The concept of cognitive tool as introduced by Lajoie and Derry (1993), will be used as a central idea in describing supportive means in discovery environments.

The approach discussed in this article is to support learners in performing discovery skills with cognitive tools. Cognitive tools are defined as instruments included in a learning environment allowing learners to make cognitive processes, like discovery skills, and their results explicit. Cognitive tools therefore can play a supportive role in discovering a domain. Central themes that will be discussed in this article are:

[1] The work described in this article is a result of co-operation with many others. The author would especially like to thank the members of the SimQuest team, especially Ton de Jong and Koen Veermans, who contributed significantly to the ideas and software presented here.

[2] The work presented here was carried out when the author was still at the Department of Educational Science and Technology of the University of Twente

- *How cognitive tools can be designed, starting from a theory on discovery learning.* A design theory for cognitive tools is necessary to be able to offer genuine support for discovery learning. It will be discussed how a dual search space model of discovery can lead to the choice and design of cognitive tools.
- *Characteristics of cognitive tools in terms of their impact on the learning process.* Cognitive tools have intended and unintended effect on discovery learning processes. They influence the way discovery processes are carried out and the learner's freedom.
- *The integration of cognitive tools in a simulation environment.* It will be discussed why integration is important and how this integration can be effectuated in a simulation-based learning environment.

In this article a number of examples of cognitive tools will be presented and discussed, as they appear in the SimQuest authoring environment for simulation-based discovery learning, as well as in other discovery environments.

The article starts with elaborating on the concepts of discovery learning and cognitive tools. In following parts of the article, the two concepts will be related and it will be shown that cognitive tools provide hooks for introducing system intelligence to support discovery learning without disrupting the nature of discovery. Finally, the SimQuest system is introduced as an authoring system explicitly introducing the concept of cognitive tools for discovery learning into a practical application.

## Discovery learning

Discovery learning is a type of learning where learners construct their own knowledge by experimenting with a domain, and inferring rules from the results of these experiments. The basic idea of this kind of learning is that because learners can design their own experiments in the domain and infer the rules of the domain themselves they are actually *constructing* their knowledge. Because of these constructive activities, it is assumed they will understand the domain at a higher level than when the necessary information is just presented by a teacher or an expository learning environment.

In practice, it has been very hard to find solid evidence for this hypothesis. For instance, when learners are confronted with a simulation of a physics domain, and are asked to discover themselves which physical laws can explain the phenomena that can be observed, many learners will not reach a satisfactory result. The reason for this is not necessarily that the idea of discovery learning is not a good idea at all, but it indicates that learners need more than just the domain to learn about it. Apart from access to domain information, they need assistance in selecting and interpreting this information to build their knowledge base.

In research on scientific discovery learning, it has been found that in order for discovery of learning to be successful, learners need to posses a number of discovery skills (De Jong & Van Joolingen, in press), including *hypothesis generation*, *experiment design*, *prediction*, and *data analysis*. In addition, regulative skills like *planning* and *monitoring* are needed for successful discovery learning (Njoo & De Jong, 1993). Apart from being supportive for learning about the domain at hand, these skills are usually also seen as a learning goal in itself, as they are needed in a complex information society. Lack of these skills can result in ineffective discovery behavior, like designing inconclusive experiments, confirmation bias and drawing incorrect conclusions from data. In its turn, ineffective discovery behavior does not contribute to creating new knowledge in the mind of the learner.

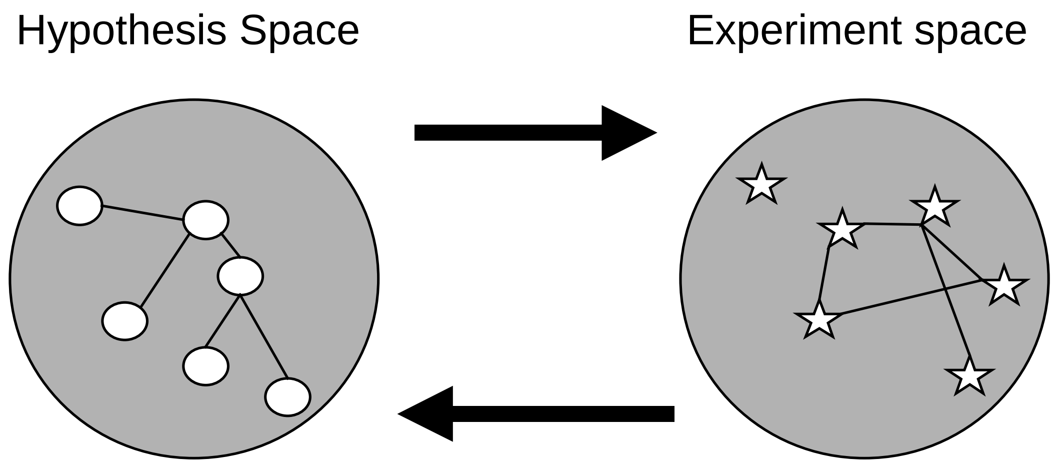


**Figure 1.** Discovery described as a search in two related search spaces.

In order to improve the effect of discovery learning, and to find beneficial effects of the basic principle of knowledge construction, one can try to support these discovery learning processes. However, as mentioned above, supporting discovery is not trivial as the very nature of discovery requires that the learner has sufficient freedom to select and interpret information meaning that any kind of support that limits this freedom in principle disrupts the nature of the discovery process.

In order to find a way out of this paradoxical situation, first discovery learning processes will be discussed in more detail in order to find the ways in which they may be supported. These models of discovery can then form the basis of supportive measures, like the cognitive tools introduced in the following sections.

*Models of discovery learning*

Models of discovery learning go back to the work of Newell and Simon (1972), Simon and Lea (1974), Qin and Simon (1988) and Klahr and Dunbar (1988). In this view the two problem spaces reflect a space of rules, or hypotheses, that are possible descriptions of the domain, and an instance or experiment space, which represents the data that can be collected in the domain. The two searches in these spaces are related: hypotheses can direct the search in experiment space and results of experiments can influence the search for new hypotheses.

Klahr and Dunbar (1988) found two different search strategies in these dual search spaces. One starts with a certain hypotheses and uses experiment space search to find confirming or rejecting evidence for this hypothesis. When hypotheses are rejected, first a new hypothesis is formed before new experiments are generated to put this to the test. This strategy is labeled a *theorist* strategy. The opposite *experimenter* strategy starts with collecting data before a hypothesis is stated. Klahr and Dunbar (1988) found that the main selection criterion between these strategies lies in the prior knowledge of the discoverer. When a learner can state the correct hypothesis, perhaps as one of a list before the search starts he or she is likely to use a theorist strategy.

Van Joolingen and De Jong (1997) extended the theory put forward by Klahr and Dunbar. They explicitly represented the knowledge of the discoverer in terms of subspaces of hypothesis space. The main concepts they introduced are the *learner hypothesis space* and the *learner search space* (see Figure 2). The learner search space contains all hypotheses the learner possibly knows of, in terms of variables in the domain, and relations that may exist between those variables. The learner search space consists of those hypotheses that the learner actually may consider as candidate for the actual relation that holds in the domain. For instance, when the learner knows of the variables of position and velocity and knows of an exponential relationship, an exponential relation between position and velocity would be in the learner hypothesis space. Would the learner's domain knowledge (or something else) tell the learner that such a relation would not make sense, the same relation would not be part of the learner search space. This difference in status between various – possible – hypotheses in the hypothesis space can be used to describe or explain differences in learner behavior. The basic assumption is that it will be more difficult to apply search operations that cross a border of a subspace in hypothesis space than to remain within a subspace. For instance to apply a search operation which moves from within the learner search space to outside this space, requires the learner to consider a relationship which did not come to mind at first investigation, whereas a search operation within the learner search space only shift focus from one alternative considered to another. Of course, the border is never "crossed" because as soon as a search operation generates a hypothesis, this will be considered to be inside the learner search space.

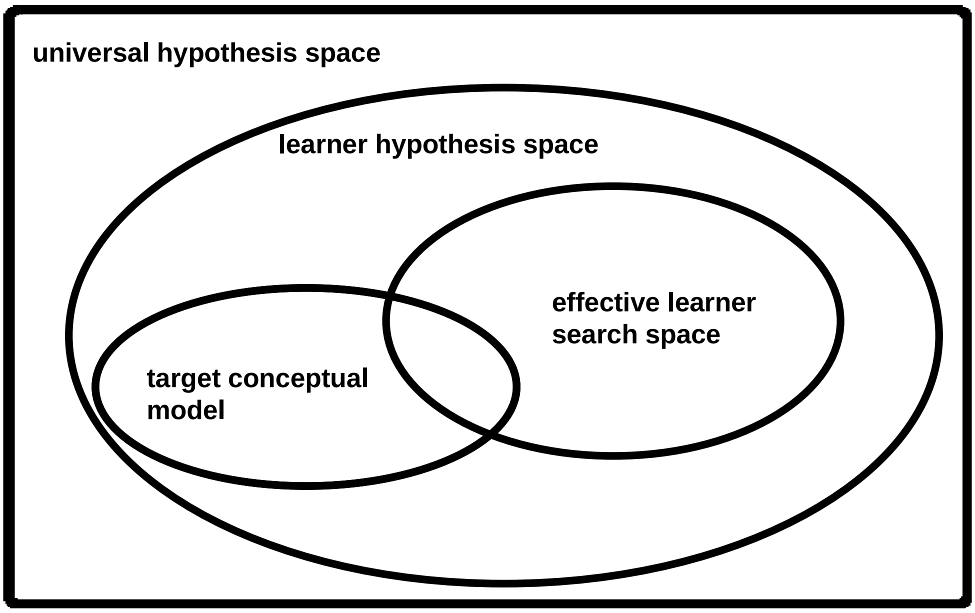


**Figure 2.** The decomposition of hypothesis space as described by Van Joolingen and De Jong (1997).

De Jong and Van Joolingen (1998) provide an overview of research performed in this area which shows that learners can have all kinds of problems with discovery learning processes. The theoretical approach to discovery learning as outlined above provides some insight into why discovery learning may be troublesome for learners, and hence be not as successful as its advocates may claim. First of all the learning processes as identified by Njoo and De Jong (1993) may themselves be problematic. For instance, their research indicates that both transformative and regulative processes may cause problems for learners. Transformative learning processes relate directly to the search processes in the dual search space theories discussed above. Learners may lack knowledge of the search processes themselves, for instance not know the idea of generalization of hypotheses, or they may have insufficient knowledge of what kind of hypotheses to state or what kinds of experiments to perform. These problems have all been observed in practice. For instance a known problem is that of *conformation bias* where learners perform only experiments that are able to confirm a hypothesis. In terms of the dual space search theory this can be described as an (unwanted) constraint on experiment space. Many problems with discovery learning can be described in terms of the search spaces. Sometimes learners have constraints on the search in either of the two search spaces that prevent successful discovery. In other cases, learners cannot find or construct constraints that would lead them in a successful direction. For instance, when doing experiments, a useful constraint is to vary not too much at a time. Learners missing this constraint will have trouble drawing sensible conclusions from their experiments. In hypothesis space, constraining oneself to a subset of the space will help focussed and more efficient discovery.

**Cognitive tools**

The term ‘cognitive tools’ was coined by the book edited by Lajoie and Derry (1993). In this book the view was expressed that computers could support learning by explicitly supporting or representing cognitive processes. In such a sense computers could serve as being a ‘mind extension’, augmenting the limited capacity of the brain. More general we can define cognitive tools as being instruments that are designed for supporting cognitive processes and thereby extending the limits of the human cognitive capacities. In principle anything can be a cognitive tool, for instance a sheet of paper and a pencil can be a cognitive tool to support the cognitive process of remembering items, extending the limited capacity of working memory.

When applied to learning, cognitive tools can be seen as supporting *learning processes*. Learning processes are the basic entities of describing the activities a learner needs to do in order to increase his or her understanding of a certain domain. This can be to remember something, to practice a procedure, to solve a problem, to set a hypothesis or some other process. Learning processes, when carried out properly, contribute to the construction of knowledge by the learner. However, learning processes may be quite difficult for a learner to perform, for instance because the process is very complex, not understood or because several learning processes need to be carried out at the same time. For instance, solving a complex physics process requires that the learner understands the physical properties of the problem, converts it to a mathematical representation, solves the resulting mathematical problem, understands the results in terms of the physics of the situation and monitors this whole process of creating representations and switching between them. This leaves the learning process as rather difficult, because multiple processes are carried out in parallel and not all of them may be very well understood by the learner. In teaching situations teachers usually assist the learner in learning how to perform processes by letting them concentrate on the essentials of the problem, for instance by offering them a divide and conquer strategy (first work out the physics characteristics, then solve the mathematical side of the problem).

In this example of physics problem solving, especially in physics domains where the mathematics is rather complex, such as electromagnetism, a mathematical symbolic equation solver (like Mathematica) can serve as a cognitive tool for these learners. The tool can take over part of the cognitive processes (solving the basic mathematics problems), in such a way freeing up working memory for concentrating on the real problems, those concerned with the physics. Note that the equation solver here *acts* as a cognitive tool. In other contexts, it may also serve different purposes.

Other examples of cognitive tools are tools for visualization of processes and domains in graphs, animations etc., as well as tools that can impose a structure on a reasoning process. For instance, visual instruments like Belvedere (Suthers, Weiner, Connely, & Paolucci, 1995) allow learners to see the structure of an argumentation, in such a way supporting the reasoning process by externalizing it.

It would be fairly easy to see almost all things that support learning as a cognitive tool. In order to limit the scope of the term, we define it within the context of this article as an *instrument* that is part of the learning environment that supports or performs an identifiable *cognitive process* that is part of the complete learning experience by the learner. In general, cognitive tools will not be the main instruments to present domain information, but they will be the instruments helping the learner to direct the process, to perform a part of it, or to externalize part of processes and memory.

In the next sections, the use and design of cognitive tools into simulation-based discovery learning environments will be discussed.

## SUPPORTING DISCOVERY LEARNING WITH COGNITIVE TOOLS

In supporting discovery learning, cognitive tools may play an important role. As we have seen above, learning processes in discovery may be very complex and all kinds of problems for learners with discovery processes have been observed. In this section, we will investigate how cognitive tools can be used to support and direct the search processes in hypothesis and experiment space, by removing unwanted constraints and directing the search by learners in a fruitful direction. The challenge here is to make the tools unobtrusive, i.e. keeping the discovery character of the learning environment while at the same time trying to support and direct the learning processes going on. Two special examples will be discussed, one for supporting the generation of hypotheses, thus supporting search in hypothesis space, the second on monitoring experimentation, in the support for searching experiment space. The final part of this section will address hooking intelligent learner support to cognitive tools.

**Hypothesis generation**

Hypothesis generation is a very crucial process in discovery learning. It is the process where the learner actually generates new knowledge and ideas. Hypothesis generation has been found to be one of the most difficult learning processes in discovery learning. In the current section we deal with the support of two major problems, the structure of the hypothesis itself and unwanted constraints on hypothesis space search.

Learners very often do not know what the basic elements of a hypothesis are (Van Joolingen & de Jong, 1991). Lacking this basic knowledge, the process of hypothesis generation will not even come to a start. This means that successful discovery is hindered by a relatively small flaw in the learner's knowledge. Hypotheses are statements that a certain relation holds between two or more variables. In formulating ideas, learners often divert from this form, expressing them in other ways. This still may be a valid representation of the idea for the learner, but a bad syntax will prevent proper testing of the hypothesis. Another problem can be that learners do not have enough prior knowledge about the extent of the hypothesis space. They may not know the variables to express ideas about or lack knowledge about the relations that exist and can be used to construct expressions.

Cognitive tools can help the process of hypothesis generation in different ways. This goes from *performing* the process for the learner, by providing a menu of ready-made hypotheses (Michael, Haque, Rovick, & Evens, 1989), to tools that constrain or extend the search process in hypothesis space. One cognitive tool that we used in our own research is the *hypothesis scratchpad* (Van Joolingen & de Jong, 1997). The basic idea was to support the cognitive process of hypothesis formation by constraining the expressions that a learner could state as a hypothesis and by showing the contents of hypothesis space to the learner. The cognitive tool is a notebook-like screen that structures the statements that the learner can make to be valid hypothesis. This is made operational by offering a template for a hypothesis that the learner can fill in with variables and relations.

Figure 3 shows the hypothesis scratchpad in the version that was introduced in the SMISLE system (De Jong & Van Joolingen, 1995, Van Joolingen & De Jong, 1996). The top shows the ingredients for a hypothesis, the middle part shows a hypothesis under construction and the bottom part shows the hypotheses that the learner has formulated before.

Relating this scratchpad to cognitive processes one sees that the template supports the basic hypothesis formation process. The ingredients for the hypothesis open up the dimensions of hypothesis space that the learner needs to form hypotheses. In such a way, the search process itself is stimulated, the learner can be given access to all or a part of hypothesis space. The bottom part of the hypothesis scratchpad serves as a memory of the trace through hypothesis space, in effect showing the effective learner search space. The list of hypotheses saves the hypotheses in the order they were created. It is impossible to delete them, but learner can annotate them and mark them as tested and assign them a truth value.

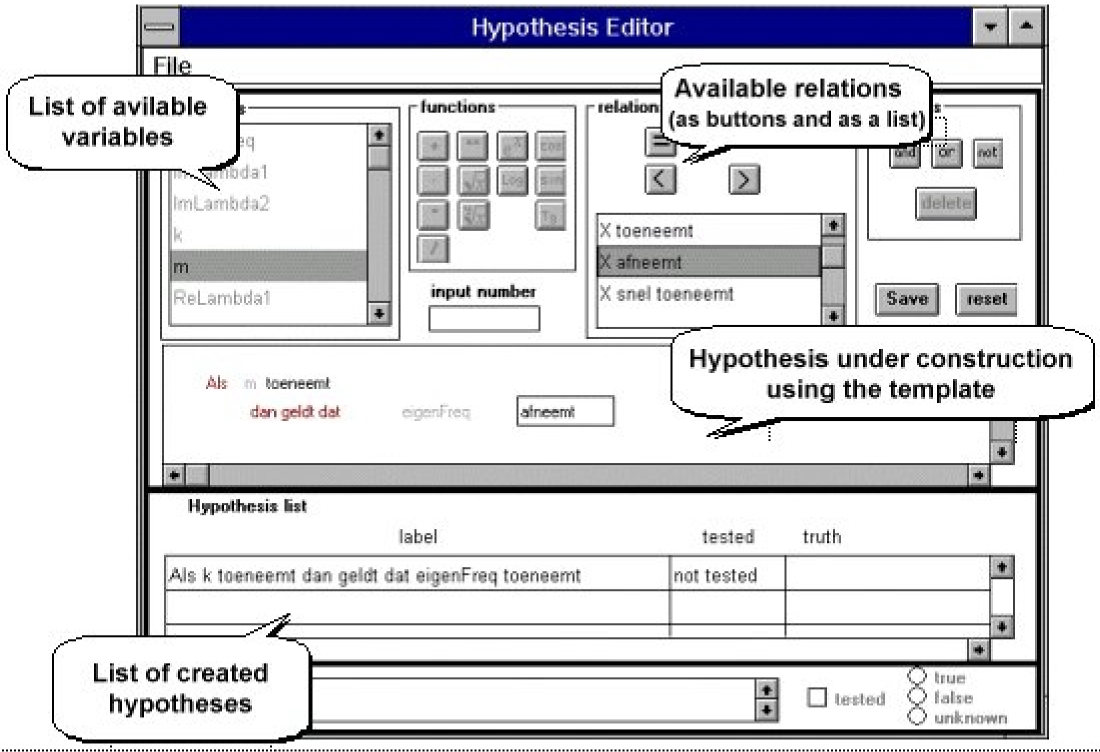


**Figure 3.** The hypothesis scratchpad, in the version of the SMISLE system (De Jong & Van Joolingen, 1995, Van Joolingen & De Jong, 1996).

The hypothesis scratchpad was tested in experimental settings with different instructions to learners on its use. In Van Joolingen and De Jong (1993) it was found that the hypothesis scratchpad was especially suitable to help learners with their initial exploration of hypothesis space and that such an initial exploration resulted in more, better and more explicitly tested hypotheses. In effect the scratchpad made learners more aware of the hypothesis generation process. Unfortunately the study failed to show any effects on posttests, but that may be attributed to the relatively short time learners worked with the system (a simulation of chemical error analysis problems).

**Monitoring experiments**

Another cognitive process associated with discovery learning is that of experimentation. In order to be successful in a discovery environment, learners must design experiments that serve as inspiration for hypotheses or that test hypotheses. In both cases, the experiments must be well designed. For instance, it is wise not to vary more than one variable at a time, to choose sensible variable variations (for instance multiply a variable by an integer number to test linear relationships), and to be aware of discontinuous changes in output variables (thresholds). In order to test a hypothesis usually a multitude of experiments has to be carried out. This puts a burden on working memory if the effects of all manipulations of variables have to be remembered by the learner. Therefore, a simple memory aid can already be quite helpful in storing experiments performed, and remembering them. Figure 4 shows an example of such a tool as described in Veermans & Van Joolingen (1998). The tool stands in close connection to the simulation (in this case of wastewater generation) and is able to read experiments from the simulation and store them on its internal list. The learner is in full control here, he or she can decide to add or delete an experiment. The tool offers means to manage the experiment like sorting experiments, replaying experiments and ordering variables. These relatively simple possibilities help learners to make sense out of the data they generate.

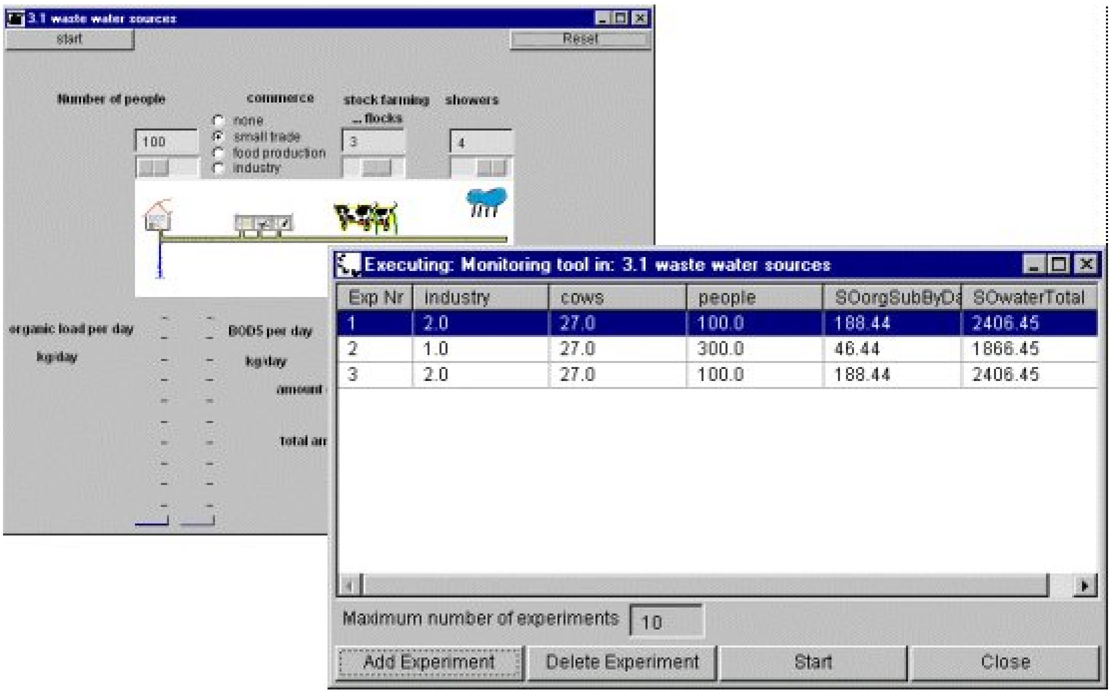


**Figure 4.** Example of a monitoring tool for keeping track in experiment space.

As a cognitive tool the monitoring tool, as it is called, clearly supports the search process in experiment space, but in a different way than the hypothesis scratchpad supports hypothesis space search. Whereas the hypothesis scratchpad is mainly proactive, stimulating the learner to explore areas of hypothesis space by showing their presence on the scratchpad, the monitoring tool is reactive, in the sense that it comes into action only after experiments have been done by the learner. Relieving memory is clearly its main task, but offering means to get an overview of the part of the experiment space that has been searched is an important function of the monitoring tool as well.

## Hooking intelligent support

The hypothesis scratchpad and the monitoring tool have in common that they both are relatively simple (it is admitted that the interface of the hypothesis scratchpad is a bit complex). They have a few functions, mainly for recording and editing. They interact with the discovery environment, a simulation, by retrieving and setting variable names and values. However, both tools have a great potential for much more intelligent support for the learner.

Intelligent learner support is a problem in open learning environments like discovery environments. Because of the open nature of the learning environment, it is hard to track what a learner does and thinks and therefore it is difficult to provide adaptive support to the learner. In the first instance the cognitive tools discussed here provide the learner with support on performing learning processes in a non-adaptive way, and their structure is constant. However, a strong point is that they let learners make their learning process explicit. Based on these explicit learning processes, a system (especially the cognitive tool itself) can provide the learner with feedback on the learning process or adapt the environment to optimise the learning processes.

This idea was elaborated in Veermans and Van Joolingen (1998). The monitoring tool that was designed to record experiments was used as a means to assess the experiment space search by the learner and provide feedback in order to stimulate the learner to increase the quality of experiments. The assessment method used assumes that the learner is testing a hypothesis and that that hypothesis is known to the system. This was controlled by providing learners with assignments that explicitly asked the learner to test a given hypothesis. In other contexts, of course, a link with a hypothesis scratchpad is feasible. Given this context and a set of experiments the monitoring tool can provide feedback on the experiments the learner did and on the conclusions the learner deduced from the experimental results.

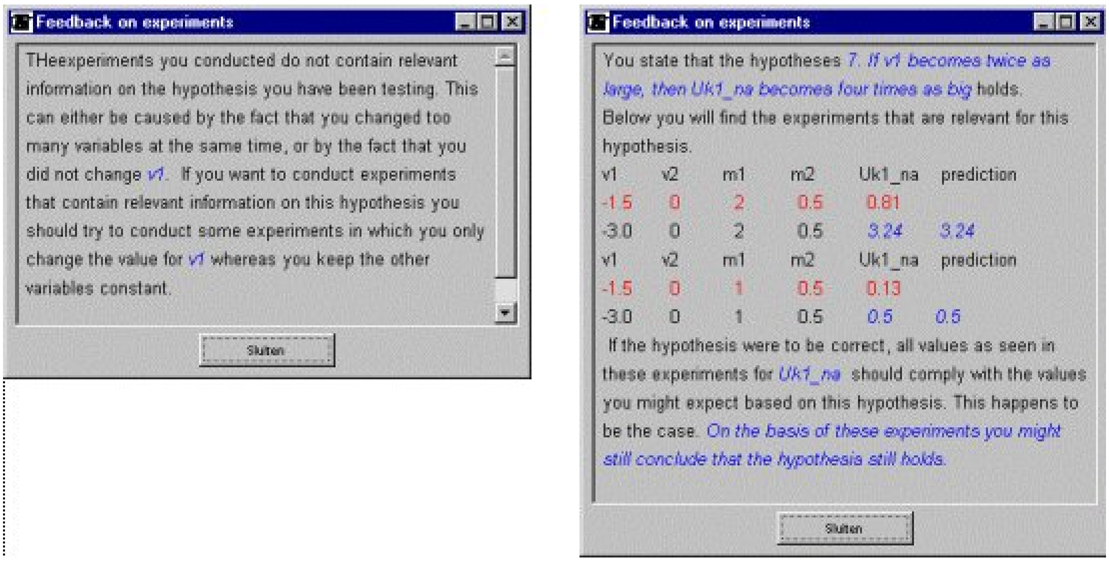


**Figure 5.** Two possible kinds of feedback that the learner can receive from the monitoring tool. The dialog on the left indicates that the quality of the experiments is such that no conclusion can be drawn. The dialog on the right indicates that the experiments done so far support the hypothesis.

Figure 5 shows two examples of feedback generated by the monitoring tool. One refers directly at the quality of the experiments performed, the other concerns the relation of the experimental results with the current hypothesis. The method used for generating this feedback is based on a set of rules determining what proper experiments are and then applying deduction processes from the hypothesis to experiment space and seeing how the experiments fit with the predictions generated by the hypothesis.

There are more ways to intelligently support the learner. For instance, the monitoring tool could advise the learner to perform a specific experiment, based on predictions generated from a hypothesis. Also, the simulation could be asked to constrain the experiment space, for instance by disabling manipulation of some variables. Likewise, the hypothesis scratchpad can do an analysis of hypothesis space search and provide feedback or constrain the learner, for instance by limiting the amount of variables and relations that are visible on the scratchpad.

## DESIGN ISSUES

The two examples of cognitive tools given above both interact with the discovery environment in an intimate way. For instance, the monitoring tool must know which variables there are, which values they can assume and it must be able to read and set these values in the discovery environment, for instance the simulation. Similarly, the hypothesis scratchpad must have information about the domain in order to build its lists of variables and relations and for generating intelligent feedback it probably needs to know intimate details about the domain models.

This situation impedes flexible authoring of discovery learning environments with cognitive tools. The idea of a monitoring tool is on the one hand quite general, but on the other hand for actually working with this tool it needs intimate co-operation with the discovery environment. This tight level of integration is a problem because it seems to imply that for each new situation a new monitoring tool must be created. Especially for the many different kinds of simulation environments that exist, each time an interface must be designed to extract the variables and to read and control their value. In research projects this would be manageable, but when deploying discovery environment to schools and practice centers, it would be impossible to support adapting cognitive tools to a multitude of different environments.

**SimQuest, component based design of learning environments**

The way out of this situation is to design discovery environments as component-based systems, with standard interfaces between the various parts, or building blocks, which constitute the learning environment. This approach is taken in the SimQuest authoring system for simulation-based discovery environments (Van Joolingen, King, & De Jong, 1997). SimQuest takes the approach that all applications can be built from building blocks, which then will be glued together to form a complete discovery environment. The basic assumption behind SimQuest is that it is possible to define interfaces between several kinds of cognitive tools and simulations.

Figure 6 displays the main authoring process in SimQuest. The author can pick building blocks from a library and move them to the discovery environment that is under construction. The library elements include parts for all components in the discovery environment, like simulation models, elements of the user interface and more or less complete cognitive tools, like the monitoring tool that was presented above. The building blocks are linked together and configured by the author in order to yield a complete learning environment.

The reason why this works within SimQuest is because the architecture below the mechanism allows this. Because simulations can take many different forms in SimQuest their properties are abstracted to a standardized form for the information that is actually needed by the other components in the learning environment. Every simulation environment uses variables that express its state and represent the inputs and the outputs. The number and nature of the variables and their possible values may change, but this basic way of representing the world is common to all simulations. Also there may be many ways of controlling a simulation, but most of these reduce to starting or stopping time, changing a variable value or changing a parameter, for instance one controlling the time frame.

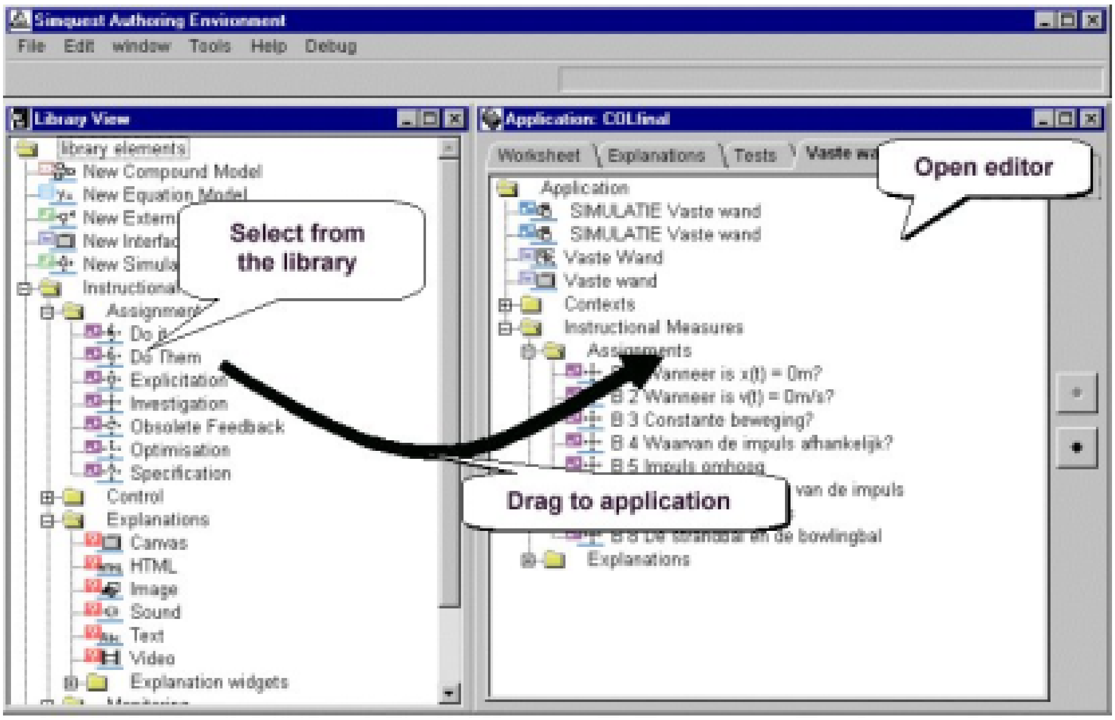


**Figure 6.** The main authoring process in SimQuest. On the left the library of available building blocks, on the right the discovery environment under construction.

Within SimQuest this abstracted information is represented by a *simulation context*. The simulation context is the main means of communication between the simulation (or in fact any other kind of experimental environment) and all other components in the learning environment, such as a monitoring tool. In such a way the simulation context is the glue that keeps the various components together. All elements of the learning environment have a well defined interface to the simulation context, which passes on all relevant information to the components it may concern. For instance, changes in variable values are passed on to those components that have expressed interest in these values. Also, control over the simulation is co-ordinated through the simulation context. The basic architecture of any learning environment created with SimQuest looks like Figure 7. In this figure, instructional measures represent all possible components to support the learner, including cognitive tools.

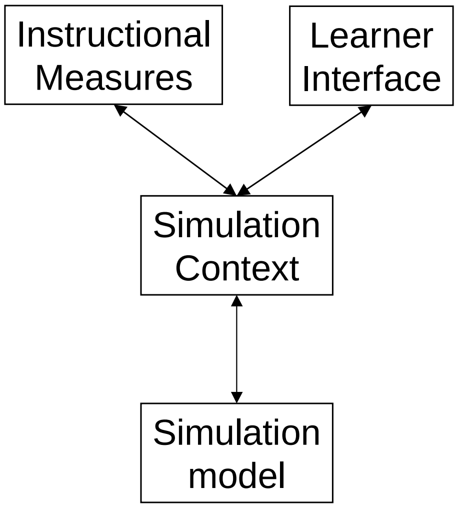


**Figure 7.** The basic architecture of a SimQuest learning environment

The SimQuest Architecture has been evaluated in various contexts. More than fifteen simulation based learning environments have been created with this system in which components like the monitoring tool, assignments, and explanations have been reused in each of the environments. The simulation models in these environments varied from simple static and dynamic systems to complex systems developed externally of SimQuest. These environments now find their way to practice and are used in experimental studies. For instance, see De Jong, et al. (1998, 1999).

## CONCLUSIONS

Cognitive tools as discussed in this article form a valuable addition to open learning environments like discovery learning environments. Cognitive tools were defined as instruments that support cognitive processes by relieving working memory or presenting a structure for performing the process. Also cognitive tools can perform a cognitive process automatically, allowing the learner to concentrate on another process first.

Using cognitive tools as hooks for adaptively and intelligently supporting learning processes is a promising means of solving the problem of adaptive support for learners in open learning environments. Because of their nature of supporting learning processes by making them explicit and visible, intelligent ways of generating support become possible. The kind of support that can be generated includes adaptive feedback, stimulative measures like providing the learner with new information on a search space and constraining measures, limiting the learner’s search spaces. For the latter kind of support, the cognitive tools themselves can be an excellent means of implementation. For instance, the hypothesis space can be constrained by manipulating the variables and relations that are visible.

Finally, we presented an architecture for component-based design of discovery learning environments. The central element in this architectures is the simulation context which glues together the components present in the learning environment, the experimental environment, often a simulation, learner interfaces and learner support measures, including the cognitive tools that have been discussed in this article. The architecture enables generic design of cognitive tools that become pluggable into any discovery environment following this architecture.

In the future we may see more and more tools supporting all kinds of learning processes, including collaboration between learners, higher order skills, and advanced experimentation. The learning environment of tomorrow will allow learners to customize their way of working and receive the support they really need. Cognitive tools like these discussed here offer a wide range of possibilities in controlling the balance between guiding the learner and allowing for enough freedom for real exploration. In such a way the learner and the learning environment can find a position on the continuum between expository and discovery learning.

## REFERENCES


Jong, T. de, & Joolingen, W.R. van (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research*, *68*, 179-202.

Jong, T. de, & Joolingen, W.R. van (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 - 186) Amsterdam: IOS.Joolingen, W.R., van, & Jong, T. de (1991). Supporting hypothesis generation by learners exploring an interactive computer simulation. *Instructional Science*, *20*, 389-404.

de Jong, T., van Joolingen, W.R., Swaak, J., Veermans, K., Limbach, R., King, S., & Gureghian, D. (1998). Self-directed learning in simulation-based discovery environments. *Journal of Computer Assisted Learning*, *14*, 235-246.

de Jong, T., Martin, E., Zamarro J-M., Esquembre, F., Swaak, J., & van Joolingen, W.R. (1999). The integration of computer simulation and learning support; an example from the physics domain of collisions. *Journal of Research in Science Teaching*, *5*, 597-615.

Joolingen, W.R. van, & Jong, T. de (1993). Exploring a domain through a computer simulation: traversing variable and relation space with the help of a hypothesis scratchpad. In D. Towne, T. de Jong & H. Spada (Eds.), *Simulation-based experiential learning* (pp. 191-206). (NATO ASI series). Berlin: Springer

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

Joolingen, W. R. van, & Jong, T. de (1997). An extended dual search space model of scientific discovery learning. *Instructional Science*, *25*, 307-346.

Joolingen, W.R. van, King, S., & Jong, T. de (1997). The SimQuest authoring system for simulation-base discovery environments. In B. du Boulay & R. Mizoguchi (Eds.), *Knowledge and media in learning systems* (pp. 79-87). Amsterdam: IOS.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science*, *12*, 1-48.

Lajoie S. & Derry S. (1993). *Computers as cognitive tools*. Hillsdale, NJ: Erlbaum

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* (pp. 390-400). Berlin: Springer Verlag.

Newell, A., & Simon, H.A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

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

Qin, Y., & Simon, H.A. (1990). Laboratory replication of scientific discovery processes. *Cognitive Science*, *14*, 281-312.

Simon, H.A., & Lea, G. (1974). Problem solving and rule induction: a unified view. In L.W. Gregg (Ed.), *Knowledge and Cognition*. Hillsdale, New York: Lawrence Erlbaum.

Suthers, D., Weiner, A., Connelly, J., & Paolucci. M. (1995). Belvedere: Engaging students in critical discussion of science and public policy issues. AI-Ed 95, the 7th World Conference on Artificial Intelligence in Education, August 16-19, 1995, Washington DC, pp. 266-273

Veermans, K., & Joolingen, W. R. van. (1998). Using Induction to Generate Feedback in Simulation Based Discovery Learning Environments. In: *Lecture Notes in Computer Science*, Vol. 1452 , 1998, 196-205 (10)