Examples, Rules, and Strategies in the Control of Dynamic Systems[1]

Wolfgang Schoppek

University of Bayreuth, Germany[2]

Two main types of knowledge are considered relevant to successful control of dynamic systems: input-output knowledge (I-O-knowledge), which represents specific input values together with the corresponding output values, and structural knowledge, defined as general knowledge about the variables of a system and their causal relations. While I-O-knowledge has proven important for the control of small systems, structural knowledge is expected to enhance performance when dealing with more complex systems. In an experiment, structural knowledge about a complex system was manipulated. Although the experimental group had better structural knowledge, the control group was equally successful in reaching new goals. That seems to contradict other studies where effects of structural knowledge on performance have been found. To resolve these contradictions, the consideration of a third type of knowledge - strategic knowledge - is suggested. The postulated effects of different levels of structural and strategic knowledge are explored with a computational model. The three knowledge types are used to interpret the variety of findings within a unitary conceptual framework.

Keywords: Problem solving, dynamic system, knowledge acquisition, cognitive modeling, strategies

---

[1] I wish to thank Christian Lebiere, Wayne Gray, and several reviewers for their helpful comments on earlier drafts of the manuscript.

[2] Address for correspondence: Dr. Wolfgang Schoppek, Lehrstuhl Psychologie, Universität Bayreuth, 95440 Bayreuth, Germany. E-mail: Wolfgang.schoppek@uni-bayreuth.de. Home page: http://www.uni-bayreuth.de/departments/psychologie/wolfgang.htm

Introduction

Controlling dynamic systems is an important requirement in many domains. Flying an aircraft, controlling a central heating system, or intensive medical care are examples of situations in which human operators have to interact with environments with a complex causal structure, and where the state of the environment changes, both autonomously and as a consequence of the operator's actions.

Psychological research on controlling dynamic systems has mainly focused on how people learn to handle systems that are new to them. A central question concerns the knowledge that is necessary - or at least sufficient - for controlling a system. Two main types of knowledge are discussed in the literature: (1) Input-output knowledge (I-O-knowledge) represents specific input values together with the corresponding output values. (2) Structural knowledge is defined as general knowledge about the variables of a system and their causal relations. When interacting with a new system, a subject might try to memorize specific output-input-output sequences of values - then she would acquire I-O-knowledge. A subject might also try to induce rules about the causal relations between the variables - then she would acquire structural knowledge.

The distinction between I-O-knowledge and structural knowledge is an example of the more general distinction between instance vs. abstraction theories of knowledge (Anderson, 1995). These two classes of theories are discussed controversially in many domains of cognitive psychology, e.g. categorization (Medin & Shaffer, 1978; Nosofsky, 1984; Nosofsky, Palmeri, & McKinley, 1994), or causal judgment (Allan, 1993; Anderson & Sheu, 1995; Schoppek, 1999). Like in these areas, in the domain of system control most authors emphasize one type of knowledge, neglecting the importance of the other type. Recently, more authors have emphasized the instance view rather than the abstraction view (Buchner & Funke, 1993; Dienes & Fahey, 1995; Lebiere, Wallach, and Taatgen, 1998; Gibson, Fichman, and Plaut, 1997).

In this paper, I argue for the importance of both types of knowledge. The argument is based on the observation that the data supporting the instance view were collected with rather small systems, whereas experiments with larger systems have provided evidence for the abstraction view. First, I discuss the relation between system size and the usefulness of I-O-knowledge and suggest an estimate of problem size that can be used as a criterion to distinguish between small and large systems. Second, I give an overview of experiments that support either the instance or the abstraction view. Next, an experiment is reported in which the role of structural knowledge in the control of a large system was investigated. Finally, I propose the consideration of a third type of knowledge - strategic knowledge - that can resolve the seemingly contradictory results of studies in which structural knowledge was manipulated.

The systems that were used in the studies reported below are modeled by simultaneous linear difference equations of the following type:

$$y_k(t) = \sum_i a_{ik} \cdot x_i(t) + \sum_j b_{jk} \cdot y_j(t-1) + c_k \cdot y_k(t-1) \qquad [1]$$

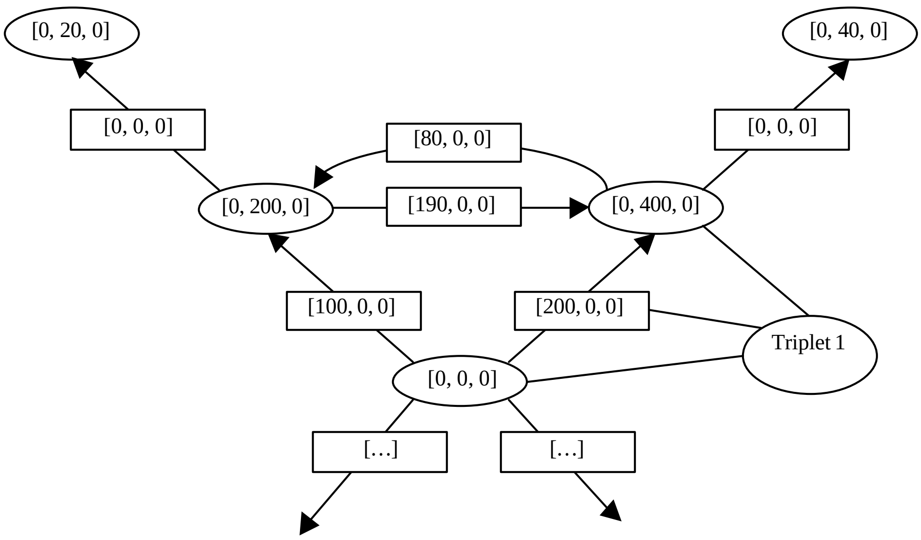
One equation is used for each output variable *y*; *x* denotes input variables. The first term describes the effects of the input variables, each effect characterized by a causal weight a_{ik}. The second term describes the effects of the other output variables, characterized by causal weights b_{jk}. The third term models what could be called momentum, i.e. the effect of the output variable on itself. The dynamic (i.e. the property that the output state can change without a change in the input variables) results from the effects among the output variables, including momentums. Some systems contain equations with the second term, or the third term, or both of them missing.

In experiments, the goal for the subjects usually is to reach and maintain target values in the output variables by entering appropriate values in the input variables. Performance is measured as the proportion of time steps where the output variables are on target, or as solution error, which averages the log absolute differences between actual and target values for all goal variables (Müller, 1993).

All of the systems mentioned in this paper are time discrete and the values of input and output variables are presented as integers. Hence, the problem spaces of these systems can be represented as finite directed graphs with all possible output states (i.e. arrays of values of the output variables) as nodes connected by all possible input states (i.e. arrays of values of the input variables) as edges. Figure 1 shows a very small fraction of the problem space of a system with three input and three output variables. As the example of the input array [0, 0, 0] demonstrates, the application of the same input array to different output states adds to the number of links. The number of links characterizes the size of the problem space or, in other words, the size of the system.

The concept of I-O-knowledge is closely related to the problem space, because it also represents specific values of input and output variables. Ideally, the elements of I-O-knowledge comprise information about the state of the output variables before an input, the input itself, and the state of the output variables after the input. Such an element is the subjective representation of a part of the objective problem space; it corresponds to two nodes connected by one link. This correspondence makes the relation between I-O-knowledge and system size obvious. If the I-O-knowledge of a subject covers a substantial part of the problem space, I-O-knowledge is likely to be useful for controlling the system. Given the current and the desired state of the system, an O-I-O-triplet can be used to determine the appropriate input. The probability that a subject retrieves an appropriate triplet in a given situation depends highly on the size of the system. I don't want to argue that the size of the problem space is generally the appropriate complexity measure for these systems, nor that it is strongly correlated with how difficult it is to control a system. There are a number of alternative complexity concepts that may be superior for predicting difficulty, for example Ashby's (1958) control complexity. However, due to the close correspondence between the problem space and the elements of I-O-knowledge, the size of the problem space is the appropriate measure for estimating the usefulness of I-O-knowledge for a specific system.

Figure 1: Small fraction of the problem space of a system with three input and three output variables. States of the output variables are represented by oval nodes. These states can be transformed by input arrays, depicted as arrows with boxes. All possible input arrays can be applied to each output state (not all are depicted though). "Triplet 1" refers to the mental representation of a part of the problem space which forms the basic unit of I-O-knowledge.



To calculate the number of links in a problem space, the number of different possible output arrays must be multiplied with the number of different possible input arrays (because each input array can be applied to all possible output states). Put another way, the system size equals the product of the cardinalities of domain and range of the entire system[3]. Since the input variables are independent of each other, the number of different input arrays equals the product of the numbers of possible values of each input variable. The number of possible output states may be constrained by the influences among the output variables, which makes it hard to calculate it. The fact that the number of output states is hard to determine is not problematic here for two reasons. First, it is not reasonable to draw a sharp line between small and large systems because there are virtually no criteria for that. Second, such a line is not necessary too: The following examples demonstrate that even with conservative estimates for the number of output states there are huge

[3] This measure is closely related to the cyclomatic complexity measure for software systems suggested by McCabe (1976), $C = e - n + 2*p$ (e: number of edges of the problem space; n: number of nodes of the problem space; p: number of connected components, here p=1). Whereas in McCabe's equation all equivalent input arrays (i.e. arrays that have the same effect on an output state) are counted as one edge, they are counted individually in the present calculation, because the equivalent operations are likely to be represented individually in I-O-knowledge - at least initially.

differences between the sizes of systems, which makes it easy to classify them as rather small or rather large.

One small system that was used in many studies - called the sugar factory (Broadbent, 1977) - has one input and one output variable with twelve possible states each. The system size of the sugar factory is thus $12 \times 12 = 144$. A typical system of the larger kind has three input variables that can be set to integer values ranging from –10000 to +10000, resulting in $20000^3 = 8 \times 10^{12}$ different input arrays that can be applied to each of the possible states of the output variables. As mentioned above, this number is harder to calculate, because the output variables are not independent of each other. However, it is clear that the size of the problem spaces of systems like that and systems like the sugar factory differ several orders of magnitude. The general prediction is that I-O-knowledge is expected to be sufficient only in the control of small systems. When large systems have to be controlled, additional structural knowledge is necessary.

When Broadbent and his working group started to investigate the control of small systems with two or four variables (e.g. the sugar factory), they expected that verbal instructions about the structure of the system would enhance performance. What they found was that performance was primarily affected by the amount of practice and that amount of practice did not affect verbalizable structural knowledge (Broadbent, 1977; Berry & Broadbent, 1984). In some cases, the instruction to search for the underlying rule even degraded performance. Only if the rule was very simple („salient“), this effect disappeared (Berry & Broadbent, 1984, 1988).

Berry and Broadbent have suggested several explanations that account for their findings. One of them involves the concept of a lookup table that stores correct actions to be taken in specific situations (Broadbent, FitzGerald, & Broadbent, 1986). Because specific values of input and output variables are represented in a lookup table, it is a form of I-O-knowledge. This concept was quite successful in explaining results with small systems. Marescaux, Luc, and Carnas (1989) tested some predictions of a model by Cleeremans (in Marescaux et al., 1989) that used a lookup table to control the sugar factory. If the model encountered a situation for which it had an entry in the table it responded with that input value, otherwise it selected one of the twelve possible responses by chance. If the response led to the desired output, it was entered in the table. The predictions of the model were tested with a „specific-situation task“ where subjects are shown descriptions of situations which they either have encountered before, or not. Subjects are asked to enter the level of workforce that would achieve the target level of sugar production. As predicted by the model, subjects performed better on old situations than on new ones. For situations in which a subject had selected a correct input value during learning, one would expect the same response in the specific-situation task. That was true in 57% of the cases.

Dienes and Fahey (1995) questioned whether an above chance concordance for old situations was sufficient to support the model. This concordance must be compared with the one for situations previously given an incorrect response and with the probabilities to give each response without sensitivity to situations. In two experiments using the sugar factory and the person interaction task (Berry & Broadbent, 1988) they calculated those different concordances and baselevel probabilities. The concordances for previously correctly answered situations were in fact higher than those for incorrectly answered situations. Based on Logan's (1988) instance theory, Dienes and Fahey developed two models, one primarily using a lookup table and one storing instances of different rules. The former model reproduced the pattern of results more closely than the latter, even though it had fewer free parameters.

Lebiere, Wallach, and Taatgen (1998) presented an ACT-R model to explain the same data. That model stores all its experiences with the system as specific instances, not only the successful ones. The instances can be retrieved depending on the similarity to the current situation. Whereas in Dienes and Fahey's model even the rules that guide the first few inputs were translated into specific instances, those rules are implemented as procedural knowledge in the ACT-R model. The model can account for the data as well as Dienes and Fahey's model, but it works with fewer assumptions.

Gibson, Fichman, and Plaut (1997) developed a connectionist model that successfully reproduced their data from an experiment with the sugar factory. Connectionist models usually process input and output values without drawing inferences. Thus, the type of knowledge acquired by this model resembles I-O-knowledge.

All these studies support the view that I-O-knowledge rather than structural knowledge is used in the control of dynamic systems. But the generalizability of this statement is attenuated by the fact that all these studies used small systems. The measure of system size proposed above is 144 for the sugar factory. Another system that was used in these studies, the city transportation system (Berry & Broadbent, 1988), is not dynamic, as the state of the output variables solely depends on the input values. Therefore the number of states of the output variables does not contribute to the size of the problem space. Assuming a plausible range of 1-500 for the two input variables "parking fee" and "time interval", its size is $2{,}5 \times 10^5$.

In the following studies, the relevance of structural knowledge, defined as general knowledge about the variables of a system and their causal relations, was investigated. The systems used in these studies had sizes of at least 10^{13} (at least three input variables with a domain of at least 1000 each and a range of at least 10000 output arrays). It is implausible to assume that I-O-knowledge is sufficient to control systems of that size flexibly, i.e. to reach and maintain arbitrary output states. Unrealistic amounts of I-O-I-triplets would be required to cover a substantial part of the problem space. Structural knowledge, on the other hand, is applicable in every region of the problem space, because causal relations stay the same regardless of specific values.

In a series of experiments, Funke (1993) used systems with three input and three output variables and varied characteristics of the system structure, e.g. the number of different effects of input variables or the amount of momentum (defined as the effect of an output variable on itself). Basically, the results show that the complexity of the system has an impact on performance and on structural knowledge. The latter was assessed by structure graphs subjects had to complete after each round of system control. In most of the experiments, path analyses revealed significant path coefficients between knowledge scores and subsequent performance scores. This supports the hypothesis that structural knowledge may be a cause of successful system control, although the hypothesis has not been tested experimentally.

One of the first attempts to manipulate structural knowledge about a larger system experimentally was undertaken by Putz-Osterloh (1993). She provided one group of subjects with a graphic of the causal structure of the system LINAS (four input variables, seven output variables). The control group explored the system without the graphic. There were no differences in performance in a test phase. But in a transfer phase, where the structure of the system was changed by skipping one output-output relation, the experimental group outperformed the control group. Moreover, the experimental group was better in diagnosing and explaining the change of the system (which was introduced to the subjects as an error). Putz-Osterloh categorized the subjects' strategies into selective and non-selective ones. Using a selective strategy means that only one or two input variables were used to control the system. Analyses of the control group data indicated that a selective strategy was associated with better structural knowledge at the end of the experiment and with a better diagnosis of the error.

Vollmeyer, Burns, and Holyoak (1996) manipulated knowledge acquisition through strategy instruction and variation of goal specificity in the system BIOLOGY-LAB with three input and three output variables. They found that providing subjects with a specific goal state in the learning phase led them to strive for that goal state rather than to acquire general knowledge about the system. Even in the condition where subjects were instructed to use a selective strategy in order to find out the structure of the system, many subjects preferred to violate the instruction and tried to reach the goals. In a test phase when all subjects had to attain these goals, the groups performed equally well. But the groups with strategy instruction and with unspecific goals were more successful in solving a transfer problem with a different goal state (all main effects significant, no interaction effect). Vollmeyer et al. interpret their findings in terms of dual-space theories. Subjects who do not try to reach goals in the learning phase explore both, the instance space and the hypotheses space, whereas subjects who try to reach specific goals mainly explore the instance space. In the terminology used here, the instance space corresponds to I-O-knowledge, and the hypotheses space to structural knowledge. The results are also viewed as supporting Sweller's (1988) claim that problem solving during learning impairs the acquisition of general knowledge.

A problem with Vollmeyer's experiment is that subjects in the condition „specific goal & strategy instruction" were in a conflict. If they tried to reach the goal state, they needed to vary all three input variables and thus violated the strategy instruction. Because most of the subjects preferred to ignore the instruction in order to reach the goals, the specific goal condition was confounded with a strong bias to use strategies that are unfavorable for the acquisition of structural knowledge.

The most intensive attempts to manipulate structural knowledge were recently undertaken by Preussler (1998) in two experiments with the LINAS system (four input variables, seven output variables). In most preceding studies, structural knowledge yielded its effects only after some practice, e.g. in transfer phases (Putz-Osterloh, 1993; Vollmeyer et al., 1996). Preussler argues that structural knowledge can be helpful much earlier, provided it is acquired in a context that is similar to the test situation. Both experiments started with one of two conditions of exploration. Then all subjects had to control the system for six rounds (test phase). Finally, there was a transfer phase with two rounds of controlling the same system with changed target values. After the test phase, structural knowledge was assessed with a judgment task. In that task, subjects are shown pairs of variable names and have to decide whether there was a causal relation between them.

In all conditions of both experiments, subjects were initially shown the momentums of the goal variables and had to answer some questions about the regularities. In Experiment 1, subjects of the experimental group were informed about the causal relations between input and output variables and used that knowledge immediately in small control problems. Those problems were small with regard to the number of variables. One input variable was used to control one output variable (goal variable). Nevertheless, they were complex with regard to the number of effects. Besides the effect of the input on the goal variable other influences had to be considered: influences from other output variables and from the momentum of the goal variable. Only correct input values were accepted; wrong inputs were rejected with a simple error message. Thus subjects were forced to figure out the correct solutions. Control subjects worked on the same problems without any hints (e.g. they could use more than one input variable, and wrong inputs were accepted). As expected, the experimental group outperformed the control group not only in the transfer phase, but also in the test phase. Structural knowledge scores were higher in the experimental group. The expected correlation between structural knowledge and performance in the experimental group was found, but only in the transfer phase ($r = -.42$, $p < .05$, the coefficient is negative, because performance is measured by solution error).

In Experiment 2, subjects of the experimental group were guided through a complex procedure of successively being informed about causal relations between variables, and using that knowledge in several control problems. This time subjects were given the correct solutions after two faults. The control group tried to reach the goal values of the later test phase with two restrictions: they could only use one input variable within each round, and they had to obtain the goal values separately. The number of trials was equal in both conditions. The pattern of results was similar to Experiment 1, but the effects were smaller, and significant correlations between the structural knowledge score and performance score were found in both conditions. This had been expected, because both exploration phases involved instructions to use selective strategies.

Preussler concludes that teaching structural knowledge to subjects can improve performance already in the test phase if this is done in a context that is similar to the later application of the knowledge. The experiments demonstrate the importance of structural knowledge in the control of a large system.

To summarize, there is much evidence that general knowledge about the structure of a rather large system (structural knowledge) can be helpful for controlling it (Putz-Osterloh, 1993; Funke, 1993; Vollmeyer et al., 1996; Preussler, 1998). The results of the control groups of these experiments indicate that most uninstructed subjects do not try to acquire and use structural knowledge. I-O-knowledge, on the other hand, seems to be acquired spontaneously; and it appears to be sufficient for the control of rather small systems (Berry & Broadbent, 1984, 1988; Marescaux et al., 1989; Dienes & Fahey, 1995; Gibson et al., 1997). The tendency of subjects to adopt instance oriented strategies might explain why the performance in large systems is generally very low, in particular in control groups where no specific instructions are given: It is simply unlikely to represent a significant proportion of the huge problem space as I-O-knowledge.

Experiment

In the present experiment, I wanted to explore more direct ways of teaching structural knowledge than providing graphics or instructing certain strategies. I also tried to avoid some confounds that were problematic in the reported studies. First, in order to circumvent the possible effect that problem solving impairs the acquisition of general knowledge (Sweller, 1988), structural knowledge was taught with a minimal amount of problem solving. Second, to avoid the conflict between reaching goal states and adopting a strategy that is favorable to acquire structural knowledge (Vollmeyer et al., 1996), a condition was introduced that allowed subjects to do both.

The experiment aimed at answering the following questions: (1) What effects has structural knowledge on performance in a test and several transfer phases, when it is taught in a supervised learning phase without the demand of solving difficult problems? To serve that purpose, an individualized tutorial about the structure of the system used in the experiment was developed. (2) Does the provision of specific goals in the learning phase really impair the acquisition of structural knowledge if this procedure does not prevent subjects from using appropriate strategies? This question was addressed by using a training condition where subjects could reach target values by selecting single input variables – an appropriate strategy for acquiring structural knowledge. (3) A third question aimed at the replication of an effect found by Preussler (1997) in an experiment involving the same system as the present experiment: The display of additional output variables that did not affect the goal variables enhanced performance compared to a condition showing only the output variables that were causally connected with the goal variables. The effect was mediated by the more extensive use of a selective strategy in the group with the additional variables. So the question is, does the display of redundant output variables enhance performance?

Method

Materials

LINAS: The system used in this experiment is called LINAS (linear-additive system). LINAS consists of four input and seven output variables connected by linear equations between input and output variables as well as between the output variables. Only three of the output variables are dynamic ones, supplied with a memory for their previous state. Following are the equations for these three variables. Output variables are denoted by artificial names, input variables by capital letters.

$$\text{Faifen}_t = A_{t-1} - C_{t-1} + C_{t-2} + D_{t-1} + 0.1 \cdot \text{Sripon}_{t-1} + \text{Faifen}_{t-1} \quad [2]$$

$$\text{Sripon}_t = 2 \cdot A_{t-1} + B_{t-1} + D_{t-1} - 0.9 \cdot \text{Sewenal}_{t-2} + 0.1 \cdot \text{Sripon}_{t-1} \quad [3]$$

$$\text{Sewenal}_t = C_{t-1} + 0.9 \cdot \text{Sewenal}_{t-1} [4]$$

The goal for the subjects is to reach specific values in the output variables Sripon and Faifen at the end of a round. A round consists of six trials where the subjects can assign values to the input variables.

A formal analysis of the system (Hoffmann, 1996) revealed that any goal state could be reached within one trial, when using at least three input variables. When using only one input variable, any state can be reached within three trials.

Linalyse: I developed a computer tutorial, called „Linalyse", which guides subjects to analyze direct effects of the input variables of LINAS, and to use that knowledge in small control problems that involve only one input value and one time-step. Figure 2 shows a screenshot of the program.

In the first part of the tutorial, values have to be assigned to single input variables, starting with input A. The program asks for the effects on the goal variables Sripon and Faifen. The subject has to make judgments about the quality of the relation (positive, negative, or no relation), and about the numerical value of the causal weight. This procedure is repeated for all input variables.

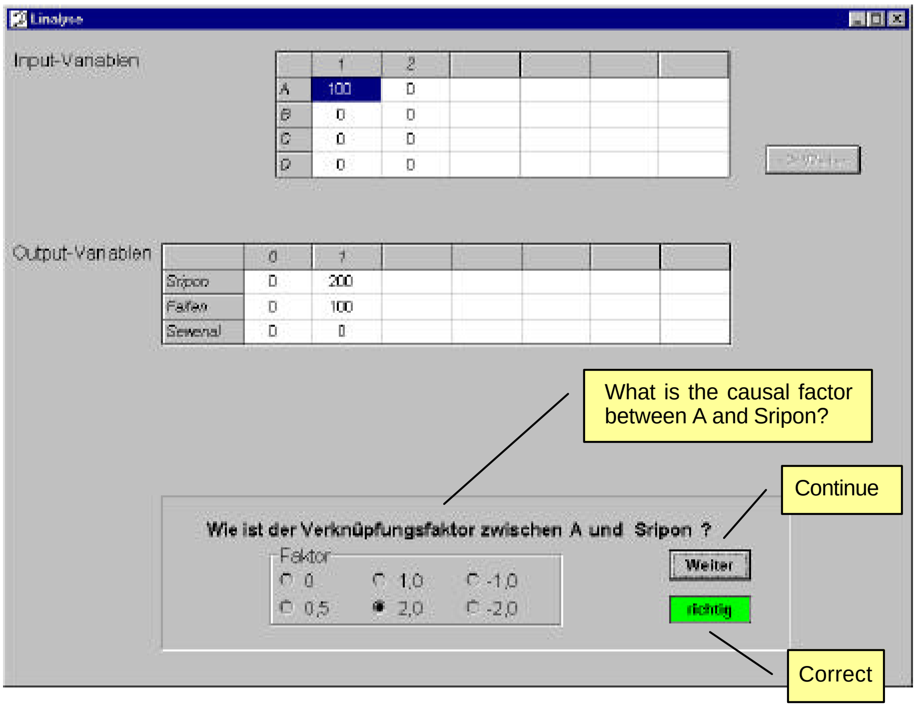
In the second part, subjects are asked to use one input variable to obtain the value of 250 in the two goal variables Sripon and Faifen (one after the other). In the case of an error, the program repeats the relevant section of Part 1. This procedure is repeated for all possible input-output combinations.

The third part deals with the momentums of the goal variables. For each input variable, the subject is guided to set an impulse (input=100 in the first time step, input=0 in the following time steps) and is asked for qualitative judgments of the development of the output values.

Finally, all causal weights between the four input and the two goal variables are requested twice in random order (including the „no-relations" with a weight of 0). If at least 75% of the answers are correct, the program is done; otherwise Parts 1 and 2 are repeated. Thus, the training phase is individualized depending on fixed criteria.

Using Sweller's conception, Linalyse should be a condition where resources can be allocated mainly to the acquisition of general knowledge. Therefore, the demand for solving problems was minimized in Linalyse. When problems had to be solved at all, the solution required nothing but the knowledge that was acquired shortly before. Thus knowledge acquisition and use were arranged so that they were clearly related to each other. There was no training of integrating several effects.

Figure 2: Screen of the Linalyse tutorial



Goal-oriented exploration: The contrasting condition was a goal-oriented exploration of the system (GOE). Subjects were instructed about the variables and the potential relations between them. Then they should try to reach target values in a single output variable at the end of a round. Subjects were instructed to use as few input variables as possible. The goal values were as follows: Sripon = 250 in four trials (two rounds), Faifen = 250 in four trials (2 rounds), Sripon = 500 in four trials (1 round), Faifen = 500 in four trials (1 round), Sripon = 500 in six trials (1 round), Faifen = 500 in six trials (1 round). Subjects had to do a considerable amount of problem solving. If the cognitive load hypothesis (Sweller, 1988) is valid, the goal-oriented exploration should result in less structural knowledge, as compared with the Linalyse group. In contrast to Vollmeyer's experiment, the problems could be solved using only one input variable. Thus problem solving did not inevitably preclude the use of strategies that support the acquisition of structural knowledge. This arrangement avoids the confound of problem solving and strategy use.

The I-O-knowledge subjects acquire in the GOE should not be applicable in the later phases, because the goal values in the GOE were quite different to those in the test and transfer phases. Therefore, a lower level of performance can be expected compared to the LILY condition. On the other hand, in the GOE condition subjects have a good chance to become acquainted with the dynamics of the goal variables.

Structural Knowledge Questionnaire: At the end of the training phase, all subjects had to answer 16 questions about the relations between the input variables and the two goal variables. In each question they had to decide if there was a causal relation between two variables, and to choose the correct mediating factor among six presented values. In the Linalyse (LILY) condition, this was a part of the tutorial.

One-Step Problems: After the transfer phase, all participants had to solve four control problems where they should attain specific values of Sripon and Faifen within a single time step. To solve these problems, no knowledge about the relations between output variables is needed, because the effects of the output variables among each other do not come into play until the second time step. Each problem was counted as correct only if both values were attained exactly.

Participants

Eighty students (40 female, 40 male) studying different subjects at the University of Bayreuth participated for payment and were randomly assigned to the conditions. With that sample size and $\alpha = 0.1$, the expected medium sized effects could be detected in an ANOVA with a power of 0.72 (power analysis calculated with G-Power by Faul & Erdfelder, 1992).

Design and Procedure

Two between-subject factors with two levels were combined, resulting in four experimental conditions. The first factor referred to the learning phase (Linalyse vs. goal-oriented exploration), the second to the number of output variables shown on the screen (three vs. seven). In the condition with three output variables, Sripon, Faifen, and Sewenal were shown. In the condition with seven output variables, four additional variables were shown that were redundant in the sense that they had no effects on the goal variables.

The experiment started with general instructions about the control of complex systems and a description of the specific system that had to be controlled. Subjects were randomly assigned to work through either the Linalyse program or the goal-oriented exploration. The learning phase ended with a structural knowledge questionnaire. The subsequent test phase consisted of six rounds in which all subjects were asked to reach a specific goal state at the end of each round (Faifen = 5000, Sripon = 200). After the test phase, subjects could take a little break. A transfer phase was then provided in which all subjects had two rounds to attain a new goal state (Faifen = 2000, Sripon = 2000)[4]. In a second transfer phase, subjects were asked to solve four one-step problems presented in random order. Finally participants were debriefed and paid according to time and performance. The entire session lasted about two hours on average.

[4] The goal states in the test and transfer phases had to be reached at the end of each round, i.e. in time step 6. To prevent subjects from leaving the system in the neutral state (all variables equal 0) until time step 5, a third goal was provided saying that Faifen must grow in every time step.

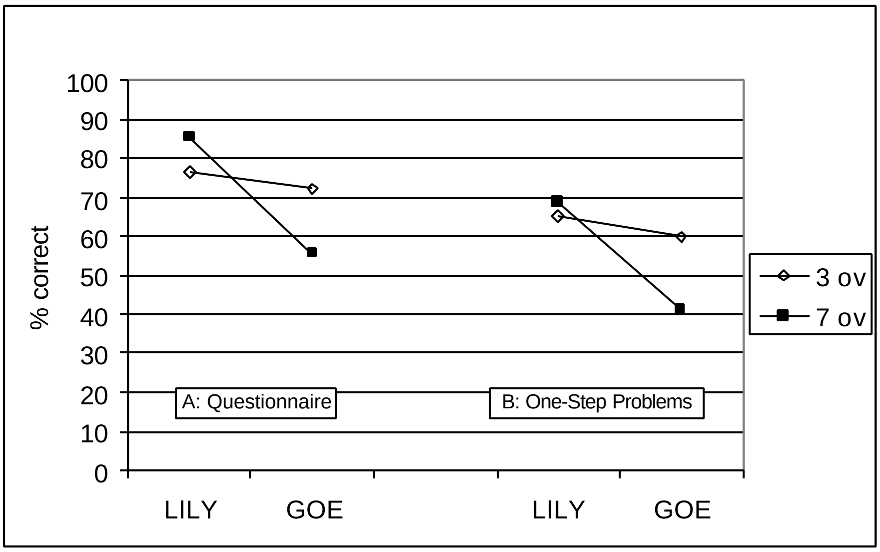
Hypotheses

If the cognitive load hypothesis is true, then LILY provides optimal conditions for acquiring general knowledge. Therefore I expected the LILY groups to have more structural knowledge at the end of the learning phase. Because structural knowledge is assumed to be important for the control of a large system like LINAS, the LILY groups are also expected to be more successful in controlling the system in the test and transfer phases, especially in the one step problems of the second transfer phase. These problems are similar to the Linalyse tutorial because of their static character.

The GOE is a challenging control condition for these hypotheses, because the subjects can - and are encouraged to - use strategies in the GOE that are favorable for acquiring structural knowledge. Therefore, it might turn out that the amount of structural knowledge is equal in both conditions. In that case, the GOE groups would be expected to outperform the LILY groups in the test phase and the first transfer phase, because in the GOE the system was handled under conditions similar to these phases. The I-O-knowledge that can be acquired during the GOE is not expected to be useful for the test and transfer phases, because the goal values of the GOE were markedly different from those of the later phases. Hence, a different region of the problem space is explored in the GOE.

According to Preussler (1997), showing more output variables on the screen enhances performance. The question here was if the effect could be replicated under somewhat different conditions.

Figure 3: Proportions of correct answers in the structural knowledge questionnaire (Panel A) and the one-step problems (Panel B). (3 ov: 3 output variables, 7 ov: 7 output variables)



Results

Performance of system control was measured by the solution error, which sums the natural logs of the absolute differences between the goal values and the actual values at the end of a round. In order to enhance the reliability of the measures, I selected the best out of two consecutive rounds. This results in three variables for the test phase (test 1&2, test 3&4, test 5&6), and one for the transfer phase (transfer 1&2).

Structural Knowledge

Panel A of Figure 3 shows the mean proportions of correct answers in the structural knowledge questionnaire. The effects of the independent variables on the structural knowledge score were examined by a 2×2 ANOVA. There is a significant main effect of training condition ($F = 9.98$, $p < .01$), and a significant interaction between training condition and number of output variables ($F = 5.76$, $p < .05$), i.e. only in the condition with seven output variables, there is a substantial difference between the structural knowledge scores of the two training groups. As expected, structural knowledge was higher in the LILY groups.

Performance in the Test Phase

In order to test the effects of the two between-subject factors and learning effects, a repeated measures mixed ANOVA was calculated. From each of two consecutive rounds (1 and 2, 3 and 4, 5 and 6), the best result was selected, forming the within-subjects factor "time" with three levels. The only significant result of the ANOVA is the main effect of the time factor ($F = 3.79$, df = 2, $p < .05$). Over all conditions, subjects improved their performance. Neither the effect of learning condition ($F = 1.77$, df = 1, $p = .19$), nor that of number of output variables ($F = 2.13$, df = 1, $p = .15$) was significant. There was no significant interaction. Thus, the hypothesis about the positive effect of the Linalyse tutorial on performance in the test phase is not supported by the data. Also the effect that showing more output variables on the screen enhances performance (Preussler, 1997) was not replicated. (Actually, the numerical values of the means in the conditions with three vs. seven output variables point in the opposite direction than expected). The mean solution errors and standard deviations from the test and transfer phases are displayed in Table 1.

Table 1: Solution errors for the training conditions Linalyse (LILY) and goal-oriented exploration (GOE). Low solution errors indicate high performance.

	LILY (n = 40)		GOE (n = 40)	
	mean	*s*	*mean*	*s*
Test:				
Round 1&2	5.58	2.06	5.08	2.43
Round 3&4	5.50	2.08	4.65	2.50
Round 5&6	4.96	2.39	4.47	2.66
Transfer:				
Round 1&2	5.12	2.12	4.47	2.35

As I stated earlier, subjects under the GOE condition might have taken advantage of the experience to handle the system under conditions that were similar to those of the test phase. That experience could have compensated for the better structural knowledge of the LILY groups. To test this, an analysis of covariance (ANCOVA) was conducted with the structural knowledge score as a covariate. All other factors were as in the ANOVA described above. Besides the significant effect of the covariate ($F = 15.4$, $df = 1$, $p < .001$), the ANCOVA revealed a significant main effect of learning condition ($F = 4.9$, $df = 1$, $p < .05$). Estimated marginal means indicate that subjects under the GOE condition performed better than subjects under the LILY condition, when structural knowledge is controlled. No other effects were significant. Particularly, the time effect disappeared ($F < .1$).

Performance in the Transfer Phase

In the transfer phase, subjects performed at about the level of the last two rounds of the test phase (test 5&6: 4.72 transfer 1&2: 4.79). There were no significant effects of the experimental conditions in an ANOVA. Again, if the structural knowledge score is entered as a covariate, there are significant main effects of the covariate ($F = 17.2$, $df = 1$, $p < .001$) and of learning condition ($F = 4.8$, $df = 1$, $p < .05$), indicating that the GOE group is better when structural knowledge is controlled.

One-Step Problems

The means of correct solutions in the one-step problems show a similar pattern as the structural knowledge score (see figure 3, panel B). There is a main effect of training condition, too ($F = 3.65$, $p < 0.05$), but the interaction is not significant ($F = 1.75$, $p < .2$). Subjects in the LILY condition were more successful in solving the one-step problems than subjects in the GOE condition. This - together with the correlations reported below - is in line with the assumption that the differences in structural knowledge remained throughout the experiment.

Relationship Between Structural Knowledge and Performance

Correlations were calculated to determine how well the measure of structural knowledge at the end of the training phase predicted success in the problem-solving phases. If structural knowledge was applied in system control, then the measure should correlate inversely with the level of performance (high levels correspond to low numbers). There were only slight differences between the coefficients in the several conditions. In the whole sample, the structural knowledge score is correlated with performance in the test phase (best round), $r = -.32$, $p < .01$, with performance in the transfer phase (best round), $r = -.35$, $p < .01$, and with performance in the one-step problems, $r = .63$, $p < .01$. This means that structural knowledge about input-output relations did not contribute very much to dynamic problem solving, but predicts performance in the more static one-step problems quite well.

Rather low, yet significant correlation coefficients give no hints about the kind of deviation from linearity. To analyze that, I formed dichotomous variables indicating high vs. low structural knowledge (< vs. ≥ 75% correct answers) and high vs. low performance (all three goals attained vs. only a subset of goals attained). The resulting cross tables can indicate if structural knowledge is rather necessary or sufficient for successful system control. (Table 2).

Table 2: Number of subjects with low vs. high structural knowledge and low vs. high performance in the test phase, for the training conditions Linalyse (LILY) and goal-oriented exploration (GOE)

	LILY:		GOE:	
	low perf.	high perf.	low perf.	high perf.
low SK	11	1	11	8
high SK	16	12	11	10

In the LILY condition, almost all subjects who controlled the system successfully also had high scores in structural knowledge, whereas several subjects with high structural knowledge performed weakly, though. This pattern can be interpreted that structural knowledge is necessary, but not sufficient for system control. But that interpretation seems to be too simple, because in the GOE condition, subjects are distributed differently. There are eight subjects with low structural knowledge who perform well. So is structural knowledge not necessary for successful control? This conclusion is not imperative as LINAS can be controlled with just one or two input variables. If subjects in the GOE condition have discovered that property of LINAS already in the training phase, they might have acquired only that fraction of structural knowledge that was relevant for their strategy.

In both conditions there are many subjects who clearly recognized the effects of the input variables, but were not successful in solving the problems. This finding supports the view that declarative knowledge about causal relations is not sufficient to control the system successfully.

Strategies

There is another inconspicuous, but more powerful predictor of performance than structural knowledge: It is the number of input variables used in the first round of the test phase. All correlations between that number and the solution errors in the subsequent rounds (2-6) are significant with a median of r = .40, $p < .01$.

In the GOE condition, the strategy of selecting input variables can be detected already in the training phase, because subjects were free to use one or more input variables. It is very interesting that the use of this strategy predicts problem solving performance throughout the experiment. Table 3 shows the correlation coefficients between the mean number of input variables used in the GOE, the mean solution error in the GOE, and measures of subsequent success (the two variables GOE strategy and GOE performance are also correlated, $r = .47$, $p < .01$).

Using few input variables per round is associated with higher structural knowledge and predicts performance in the test phase. Moreover, the amount of structural knowledge learned in the GOE is related to successful problem solving in that phase. This challenges the conclusion of Vollmeyer et al. (1996) that problem solving during learning impairs the acquisition of general knowledge.

Table 3: Correlation coefficients between a measure of strategy (n of variables) and performance in the goal-oriented exploration (solution error), and measures of structural knowledge and performance in later phases.

	structural knowledge score	solution error (test)	solution error (trans)	one step problem score
n of variables (GOE)	-.35*	-.31*	-.25	-.11
solution error (GOE)	-.43**	.52**	.54**	.47**

* p < .05   ** p < .01   n = 40

Once more, the results demonstrate the importance of strategies. In LINAS, a selective strategy is not only useful for analyzing the relations between variables in the learning phase. Significant correlations between performance and the number of input variables used in later rounds indicate that the reduction of complexity by selecting input variables is a good control strategy too.

Discussion

The comparison between the adaptive tutorial Linalyse and a goal-oriented exploration phase revealed the expected differences in structural knowledge. Subjects trained with Linalyse had higher structural knowledge scores at the end of the training phase. They were also more successful in solving the static one-step problems at the end of the experiment, which indicates that the differences were lasting. However, when the task is to control a dynamic system, subjects trained with Linalyse were not better than subjects trained in the goal-oriented exploration. These results seem to contradict other studies (Putz-Osterloh, 1993; Vollmeyer et al., 1996; Preussler, 1998) where manipulations of structural knowledge had effects on performance, at least in transfer phases. The fact that the manipulation of structural knowledge sometimes yields differences in performance, sometimes not, demonstrates that the concept of structural knowledge is not powerful enough to explain the variety of results.

With respect to the question if providing specific goals in the learning phase impaired learning of general knowledge the data support this assumption in a limited sense: Subjects in the GOE condition had less verbalizable structural knowledge and performed worse in the one-step problems. But they were equally successful in controlling the system towards new goal states, where specific I-O-knowledge from the exploration phase was not applicable. Hence, subjects in the GOE condition must have gained some other type of general knowledge that has compensated their lower structural knowledge. Instead of interpreting the differences in structural knowledge as an effect of specific goals that *impaired* learning, they can alternatively be interpreted as an effect of Linalyse, which selectively *enhanced* learning of structural knowledge.

One question for the experiment was if the detrimental effect of goal specificity could be avoided if the attempt to reach specific goals did not preclude the use of appropriate strategies - as it was the case in Vollmeyer et al.'s (1996) experiment. If that confound is dissolved, successful problem solving during learning does not only predict further success, but even the amount of structural knowledge (see correlation coefficients in Table 3). Thus, Sweller's (1988) findings that problem solving during learning impairs the acquisition of general knowledge cannot simply be generalized to the domain of complex dynamic tasks.

The effect found by Preussler (1997) that presenting additional output variables enhanced performance could not be replicated. In Preussler's study, subjects with more output variables adopted better input strategies. That was not the case in the present experiment. Eliciting a selective strategy by information overload seems not to be a robust effect. But after all it is remarkable that the groups with additional variables did not perform worse.

Strategic Knowledge

The joint consideration of the studies of Vollmeyer et al. (1996), Preussler (1998), and the present experiment has shown that structural knowledge alone cannot account for the results, because in all three studies the manipulation of structural knowledge was successful but yielded different effects on performance. Although all manipulations in these experiments targeted at structural knowledge, it is possible that some of them affected more than the knowledge that is used to answer questions about causal relations. In fact, most of the findings reported above can alternatively be explained by a co-development of reportable structural knowledge and another form of knowledge that may be equally important for system control.

There are many hints that this type of knowledge has to do with strategies: In some experiments with large systems the effects were mediated by strategies (Preussler, 1997; Putz-Osterloh, 1993); strategy manipulations resulted in performance differences (Vollmeyer et al., 1996); strategy measures are correlated with performance (results of the present experiment). The idea that strategies are important to explain performance in the control of dynamic systems is not new (Funke, 1992; Putz-Osterloh, Bott & Houben, 1988). But the notion of strategies is notoriously ill defined and there are no clear assumptions about how strategic knowledge might be represented in memory. Often strategies are characterized by a single idea, as e.g. "vary one thing at a time". A closer look on what such an idea means in terms of behavior reveals that there is much more to implementing a strategy than having an idea. In the following, I propose a conception of strategic knowledge that was inspired by the GOMS methodology by Card, Moran, and Newell (1983). The usefulness of this conception shall be demonstrated with a computational model that controls LINAS with varying levels of structural and strategic knowledge.

The following examples of strategies can be used to find the lowest common denominator for characterizing a strategy.

a) Acquire knowledge about the system $\rightarrow$ apply the knowledge in order to attain goal states
b) Vary one input variable $\rightarrow$ infer effects $\rightarrow$ set input variable back to zero $\rightarrow$ infer more effects
c) Calculate expected value in an output variable $\rightarrow$ calculate difference between expected and desired value $\rightarrow$ select free input variable $\rightarrow$ calculate input value

Although on different levels of abstraction, all examples specify a sequence of steps. The types of steps range from goals, as in Example a, down to single operators, as in Example b (third step). Given that the strategies are the result of some underlying knowledge, this "strategic knowledge" can be defined as knowledge about how to proceed in order to accomplish a task. Strategic knowledge contains information about a sequence of steps.

This notion of strategic knowledge closely resembles the *method* in GOMS (Card, Moran, and Newell, 1983). In GOMS, methods consist of sequentially ordered steps that can be operators, goals, or selection rules. Like GOMS methods, strategic knowledge can be rather concrete, when it consists of operators, or rather abstract, when it consists of goals and selection rules. According to Card et al. (1983), I assume a continuum between problem solving strategies and skilled behavior in the development of strategic knowledge[5].

The examples also demonstrate the co-development of strategic knowledge and the other knowledge types. On the abstract level of Example a, the strategy does not imply the use of I-O-knowledge or structural knowledge. Example b describes a strategy to acquire structural knowledge, but it does not preclude incidental learning of I-O-knowledge. Example c, however, describes a strategy of applying structural knowledge. This strategy is useless if only I-O-knowledge is available.

How are the three knowledge types related to the distinction between declarative and procedural knowledge, as it is defined in Anderson's (1983, 1993) ACT theories? At first glance it might appear that I-O-knowledge and structural knowledge are declarative and strategic knowledge is procedural. But the declarative-procedural distinction is defined representationally, whereas the present knowledge classification is defined in terms of content. A sequence of steps can be represented either declaratively or procedurally. Since declarative knowledge cannot act and productions are eventually needed to make a model do something, strategic knowledge always contains procedural knowledge. But strategic knowledge should not be identified with procedural knowledge. Also, I-O-knowledge and structural knowledge might contain procedural elements, particularly when they are highly practiced. "If the goal is to bring the sugar production to 8000 and the current production is 6000 then set work force to 7" is an example for I-O-knowledge represented by a production rule.

---

[5] This also implies a continuum between strategies and tactics in a way that strategies are more abstract methods, whereas tactics are more concrete methods. The shift from problem solving (strategic) to skilled (tactical) behavior is characterized by the development and consolidation of concrete low-level methods. Because of the futility of defining a sharp line between abstract and concrete methods, I use the term "strategic knowledge" for both.

A computational model

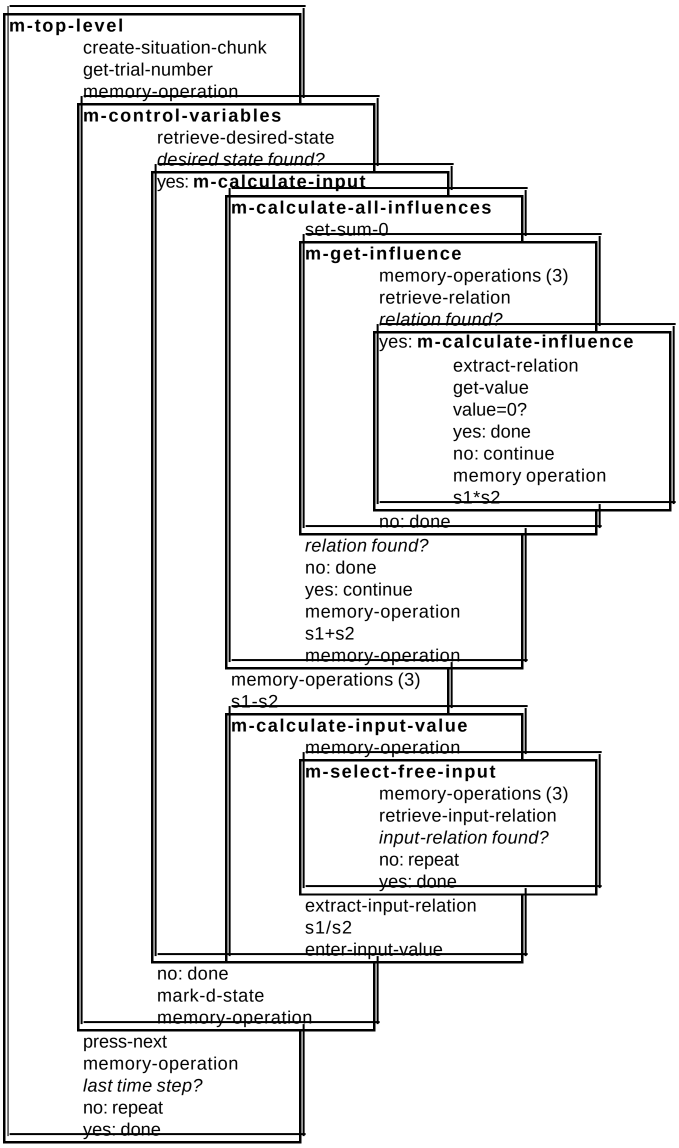
The model that is presented in this section simulates the use of structural knowledge in the control of the dynamic system LINAS. The model is written in ACT-R 4.0 (Anderson & Lebiere, 1998) and was developed on top of a core model that allows translating GOMS like task analyses into ACT-R (Schoppek, Boehm-Davis, Diez, Hansberger & Holt, 2000). A computational model has several advantages over a pure GOMS analysis: The assumptions of the analysis are tested more strictly because their effectiveness is proven through direct interaction with the task; assumptions about the representation of knowledge must be made explicit; predictions of variations in the assumptions can be made directly through simulation.

In the core model, strategic knowledge is represented as methods, consisting of steps that are stored in declarative memory. Steps are linked through associations, i.e. each step acts as a cue for the retrieval of the next step, which is most likely – but not necessarily - the correct one. Steps can trigger internal or external operations, selection rules, or initiate subordinate methods. Internal operations and selection rules are represented as productions in procedural memory. External operations are delegated to some LISP code that interacts with the task environment. Another feature of the model is that hierarchical methods are executed without a deep goal stack. Every time a subgoal is set, the old goal does not remain on a goal stack, but is stored in declarative memory, where it has to be retrieved after the subordinate method has been finished. This feature, together with the possibility that wrong next steps can be retrieved, allows to predict errors similar to those that can be observed in humans when they execute hierarchical plans (Altmann & Trafton, 1999).

For the model that controls LINAS, additional specific assumptions had to be made. One of them concerns the representation of structural knowledge. Each causal relation that is present in the system is represented by a declarative memory element (a "chunk" in ACT-R terminology) that contains information about the causing variable, the dependent variable, and the causal factor between the two variables. The strengths of these chunks in memory - and thus the probabilities of retrieval - can be varied by making assumptions about their history of use: The more often a chunk has been used in more recent times, the stronger it is. Also, the strengths of all chunks are changing according to their use during a simulation run (baselevel learning). Other assumptions have been made about task specific internal and external operators; one of the most important being the internal operator "retrieve-relation" that tries to retrieve a structural knowledge chunk depending on the variable that is currently in the focus of attention. Many other internal operators that perform memory operations, like moving retrieved information into working memory, or storing results in memory, are part of the core model. Finally, the strategies or methods had to be defined. The present model works with predefined strategies and does not learn new ones. This is an important restriction of the model that has to be overcome in future versions.

Figure 4 shows one strategy that was used by the model in the simulations. Methods are denoted in bold, selection rules in italics. The strategy's main submethods are **m-calculate-all-influences**, which calculates the predicted next state of the system without the new input, and **m-calculate-input-value**, which determines an input variable, calculates and enters a value that removes the

Figure 4: A strategy that can be used to control systems like LINAS. Method steps are denoted in bold, selection rule steps are denoted in italics. Several memory operations are subsumed under that label and their number indicated in parentheses.



difference between the predicted next state and the goal state. It is obvious that the strategy is quite complex. But for optimal performance, even more considerations would be necessary (the described strategy does not consider possible side effects of the last input). Although the strategy was developed based on verbal protocols of expert subjects, it is only one of many possible strategies. Compared to others, it is a general one that can be used to control any system of the linear-additive type. It does not rely on I-O-knowledge but on structural knowledge. Although I stated earlier that I-O-knowledge is much less important in controlling large systems than in controlling small systems, the exclusion of I-O-knowledge is a simplification. Vollmeyer et al,'s (1996) results have shown that I-O-knowledge can play a role in the control of large systems, too (high performance of the "specific goals & no strategy instruction" group in the test phase with striking drop of performance in the transfer phase). This should be taken into account in future models.

In a simulation with the model, the effects of different levels of structural and strategic knowledge were investigated. The simulation was not intended to fit a specific data set, but to explore if the proposed notion of strategic knowledge actually produces the effects that were postulated in the previous section. For each knowledge type there were two levels, “high” and “low”. Structural knowledge was varied by defining different strengths for the chunks that represent causal relations[6]. Strategic knowledge was varied by using strategies of different complexity. In the “high strategic knowledge” condition, the method shown in Figure 4 was used. In the “low strategic knowledge” condition, the method **m-calculate-all-influences** with all its subordinate methods was replaced by a single operator that estimates the influences from the state of the system by setting them to 500. This reduces the use of structural knowledge, but it does not completely eliminate it. Attempts to retrieve structural knowledge are still present in the method **m-select-free-input**.

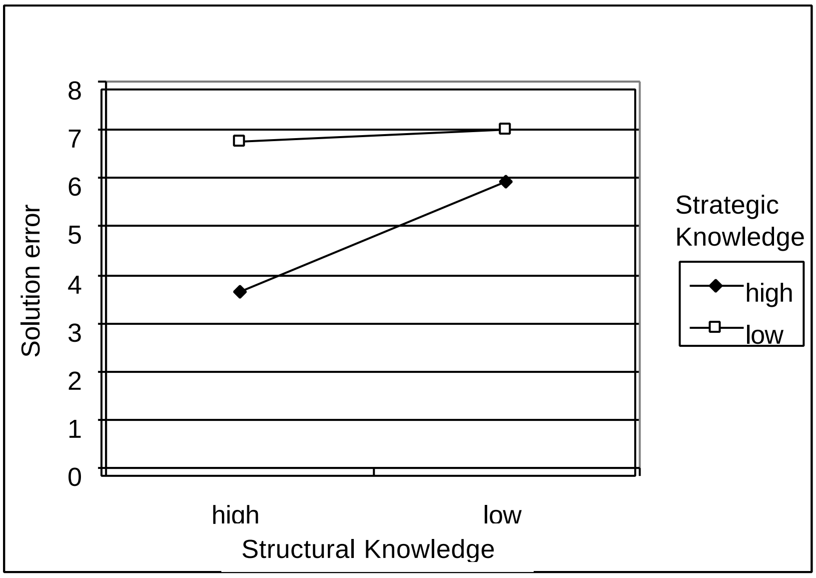
The two factors structural and strategic knowledge were crossed in a simulated experiment with 82 cases. The means of the predicted solution errors for each condition are shown in Figure 5. In line with the verbal interpretations, the simulation predicts that the level of structural knowledge only makes a difference in performance when strategic knowledge is high. Also, a main effect of strategic knowledge is predicted. So the simulation supports the view that strategic knowledge plays an important role for the explanation of results obtained with relatively large dynamic systems.

Comparing the means of the solution errors between the experiments and the simulation reveals a closer fit for the “high structural knowledge & high strategic knowledge” condition (Preussler, 1998, Exp. 1: 2.99, simulation: 3.63) than for the “high structural knowledge & low strategic knowledge” (LILY group: 4.96, simulation: 6.74). One reason for that may be that in experiments there are usually a few subjects who develop good strategies even under difficult conditions, whereas in the simulation all cases in the “low strategic knowledge” conditions use the same poor strategy. But the model generally underestimates performance. This might

[6] Initial baselevels in the “high” conditions were 1.042 (60 references in 5000 s); in the "low" conditions they were 0.735 (2 references in 100 s). The retrieval threshold was set to -0.4. See the source code of the model for further details. It can be obtained at http://hfac.gmu.edu/~wschoppek/linas-model.html

result from the model not using I-O-knowledge at all, whereas subjects benefit from this type of knowledge – under certain conditions.

Figure 5: Results of a simulated experiment with n=82 cases. Structural and strategic knowledge were varied.



Analysis of previous experiments

With the concepts of I-O-knowledge, structural knowledge, and strategic knowledge, the results that were reported above can be interpreted within an integrative conceptual framework. This shall now be demonstrated with a joint interpretation of the experiments by Vollmeyer et al. (1996), Preussler (1998), and the one reported here. For Vollmeyer's experiments, this interpretation resembles the original dual space interpretation, with the instance space corresponding to I-O-knowledge, and the rule space corresponding to structural knowledge. In the conditions that supported the analysis of single causal relations, more structural knowledge was acquired, which is well transferable to new goal states. (Note that the group without strategy instruction and with no specific goal also used the strategy to vary one thing at a time to the amount of 50%). In the conditions with specific goal states the subjects acquired I-O-knowledge that was applicable in the test phase, because the goal states were the same. However, the knowledge-space account cannot explain the improvement of the participants with no specific goal in the transfer phase (Experiment 2). Structural knowledge in these groups had been stabilized on a high level already in the exploration phase; and the I-O-knowledge acquired in the test phase could not be used in the transfer phase (because the changed goal state would have required I-O-knowledge about a different part of the problem space). The continued refinement of strategic knowledge can explain this improvement in face of constant structural knowledge. This interpretation is supported by the huge performance difference the model predicts in the “high structural knowledge” condition between low and high levels of strategic knowledge.

Differences in strategic knowledge could also be the reason why the procedure to teach structural knowledge used by Preussler (1998) resulted in better performance, whereas the procedure in the present experiment did not. Preussler holds that the knowledge was learned in a context similar to the application context. In terms of the proposed knowledge types this means that structural knowledge was learned together with strategies how to use it. In the practice problems, subjects had to reach certain values in one output variable exactly. This was only possible if at least two influences on the output variable were considered: the influences from the output variable itself, from an input variable, and in some of the problems a third influence from another output variable. Thus, subjects were forced to develop a sequence of considering the influences from the current state of the system and calculating an input value that would compensate for these influences (see Example c above). In the Linalyse tutorial, however, virtually no strategies were trained. None of the questions that had to be answered in Linalyse required sequences of more than two interdependent steps (one retrieval and one calculation). So the subjects in Preussler’s experimental group are represented by the “high structural knowledge & high strategic knowledge” condition of the simulation, whereas the subjects of the LILY group are represented by the “high structural knowledge & low strategic knowledge” condition. The GOE group of the present experiment is harder to associate with one of the simulated groups. It might be best characterized by having intermediate levels of structural and strategic knowledge - a condition that was not included in the simulation.

These interpretations can be summarized in the following statements: Structural knowledge alone is not sufficient to control a large system. Structural knowledge together with proper strategic knowledge is sufficient (probably even necessary) for the control of large systems. The strategies that are needed for controlling large systems with structural knowledge need to be developed or refined along with the acquisition of structural knowledge.

For small systems like the sugar factory, it has been shown with several models that I-O-knowledge is sufficient to control them (Dienes & Fahey, 1995; Gibson et al., 1997; Lebiere et al., 1998); strategic knowledge seems to play a minor role: None of the models contains assumptions about specific strategies. Probably, the knowledge that is needed to translate I-O-knowledge into actions is part of the general knowledge of most subjects.

Perspectives

The concept of strategic knowledge cannot only be used to explain existing results, but also implies new predictions. Although the more sophisticated forms of strategic knowledge depend on structural knowledge, the former should bridge greater transfer distances than the latter. Structural knowledge can be transferred to problems with different goal states within one system. Strategic knowledge can be transferred across systems of the same type (e.g. systems modeled by simultaneous linear equations). Even strategic knowledge that depends on the availability of structural knowledge can be transferred across systems when the references to structural knowledge are variable. For example, when a strategy requires considering all dependences of an output variable it does not matter how many dependences there are and how the causal weights are.

In future research, questions about the development of strategic knowledge should be addressed. The effects of strategy instructions found by Vollmeyer et al. (1996) indicate that strategies can be taught, but the effects are fragile. Guiding subjects to use a strategy that is not declared as such is not a good means to teach it. In a recent experiment (Schoppek, in prep.), subjects who were guided to put impulses into small systems in order to explore their dynamics did not use that strategy in a transfer phase where they had to explore a larger system. It appears to be important to establish cues that signal that an acquired strategy can be applied in a given context (Preussler, 1998). Little is known about the processes by which subjects develop and refine strategies on their own.

Further research is required to investigate the role of I-O-knowledge in the control of large systems. For this question, the modeling approach is promising. The hypothesis that I-O-knowledge is not sufficient to control large systems could best be challenged by a model that controls a large system with I-O-knowledge only.

I will end this paper with a few considerations concerning possible applications of this area of research. It has been criticized that the tasks used in the described experiments are quite different from those of persons who operate real dynamic systems. In industrial settings, operators have to monitor dynamic processes rather than controlling them actively. Another difference is that subjects in experiments start with very little knowledge about the systems they are controlling and do not handle the systems long enough to gain expertise, whereas operators have to know much about the processes they are monitoring and usually do their job for a long time.

However, technology changes rapidly and there are situations in which operators must learn to handle a new system. For example, when pilots learn to operate the Flight Management System (FMS) of modern aircraft, they are facing a task environment that resembles the systems used in the research described above in many respects. Like these, the FMS is highly dynamic and entails many variables that affect each other. And the complaints of instructors about pilots applying procedures with a lack of understanding, which I heard recently, strongly reminded me of certain strategies observed in our subjects. For the improvement of training procedures in domains like aviation, it is certainly valuable to know how to overcome both, the tenacious tendency of people to use I-O-knowledge, and the inertia of purely declarative structural knowledge.

References

Allan L. G. (1993). Human contingency judgments: Rule based or associative? *Psychological Bulletin, 114,* 435-448.

Altmann, E. & Trafton, J. G. (1999). Memory of goals: An architectural perspective. *Proceedings of the 21st Annual Meeting of the Cognitive Science Society* (pp. 19 –24). Hillsdale, NJ: Erlbaum.

Anderson J. R. (1983). *The architecture of cognition.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson J. R. (1995). *Cognitive psychology and its implications.* New York: Freeman.

Anderson, J. R. & Lebiere, C. (1998). *Atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson J. R., & Sheu C.F. (1995). Causal inferences as perceptual judgments. *Memory & Cognition, 23,* 510-524.

Ashby, W. R. (1958). Requisite variety and its implications for the control of complex systems. *Cybernetica, 1,* 83 -99.

Berry D. C., & Broadbent D. E. (1984). On the relationship between task performance and associated verbalisable 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.

Broadbent D. E. (1977). Levels, hierarchies and the locus of control. *Quarterly Journal of Experimental Psychology,* 181-201.

Broadbent D. E., Fitzgerald P., & Broadbent M. H. P. (1986). Implicit and explicit knowledge in the control of complex systems. *British Journal of Psychology, 77,* 33-50.

Buchner, A. & Funke, J. (1993). Finite-state automata: Dynamic task environments in problem solving research. *The Quartely Journal of Experimental Psychology, 46A*, 83-118.

Card, S. K., Moran, T. P., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Dienes Z., & Fahey R. (1995). Role of specific instances in controlling a dynamic system. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21,* 848-862.

Faul, F. & Erdfelder, E. (1992). GPOWER: A priori, post-hoc, and compromise power analyses for MS-DOS [Computer program]. Bonn, FRG: Bonn University, Dep. of Psychology.

Funke J. (1993). Microworlds based on linear equation systems: a new approach to complex problem solving and experimental results. In G. Strube & K.F. Wender (Eds.), *The cognitive psychology of knowledge*, pp. 313-330. Amsterdam: North-Holland.

Gibson, F. P., Fichman, M., & Plaut, D. C. (1997). Learning in dynamic decision tasks: Computational model and empirical evidence. *Organizational Behavior & Human Decision Processes*, *71,* 1-35.

Hoffmann, D. (1996). *Das System LINAS*. Forschungsbericht des Lehrstuhls für Psychologie der Universität Bayreuth.

Lebiere C., Wallach D., & Taatgen N. (1998). Implicit and explicit learning in ACT-R. In F.E. Ritter & R. M. Young (Eds.), *Proceedings of the Second European Conference on Cognitive Modelling (ECCM-98)* (pp. 183-189). Nottingham, UK: Nottingham University Press.

Logan G. (1988). Towards an instance theory of automatization. *Psychological Review,* 492-527.

Marescaux P.-J., Luc F., & Karnas G. (1989). Modes d´apprentissage selectif et nonselectif et conaissances acquises au controle d´un processus: Evaluation d´un modele simule. *Cahiers de Psychologie Cognitive, 9,* 239-264.

McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering,* 308 -320.

Medin D. L., & Shaffer M. M. (1978). Context theory of classification learning. *Psychological Review,* 207-238.

Müller, H. (1993). *Komplexes Problemlösen: Reliabilität und Wissen.* [Complex problem solving: Knowledge and reliability]. Bonn: Holos.

Nosofsky R. M. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition,* 104-114.

Nosofsky R. M., Palmeri T. J., & McKinley S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review, 101,* 53-79.

Preussler W. (1997). Effekte des Kontexts auf den Wissenserwerb bei der Steuerung eines dynamischen Systems. *Sprache & Kognition, 16,* 48-59.

Preussler W. (1998). Strukturwissen als Voraussetzung für die Steuerung komplexer dynamischer Systeme. *Zeitschrift für Experimentelle Psychologie, 45,* 218-240.

Putz-Osterloh W. (1993). Strategies for knowledge acquisition and transfer of knowledge in dynamic tasks. In G. Strube & K. F. Wender (Eds.), *The cognitive psychology of knowledge,* pp. 331-350. Amsterdam: Elsevier.

Schoppek, W. (in prep.). *Context dependence of strategies in the control of dynamic systems.*

Schoppek, W. (1999). Rules and instances in the control of a static system - Modelling the influence of causal interpretation. In I. Wachsmuth & B. Jung (Eds.), *KogWiss99*, pp. 146-151. Sankt Augustin: Infix.

Schoppek, W., Boehm-Davis, D. A., Diez, M., Hansberger, J., & Holt, R. W. (2000). *Letting ACT-R fly - A model of the interaction between trained airline pilots and the flight management system.* Seventh Annual ACT-R Workshop 2000, Carnegie Mellon University, Pittsburgh, PA. (http://www.uni-bayreuth.de/departments/psychologie/ACTR-2000/index.htm)

Sweller J. (1988). Cognitive load during problem solving: effects on learning. *Cognitive Science, 12,* 257-285.

Vollmeyer R., Burns B. D., & Holyoak K. J. (1996). The impact of goal specificity on strategy use and the acquisition of problem structure. *Cognitive Science, 20,* 75-100.