

Instructional Control of Cognitive Load in the Training of Complex Cognitive Tasks

Fred G. W. C. Paas^{1,2} and Jeroen J. G. Van Merriënboer¹

Limited processing capacity constrains learning and performance in complex cognitive tasks. In traditional instruction, novices' failure to adequately learn cognitive tasks can often be attributed to the inappropriate direction of attention and the related high or excessive load that is imposed on a learner's cognitive system. An instructional design model for the training of complex cognitive tasks should provide instructional strategies that control cognitive load. We propose such a model and recommend research in which the cognitive load of instructional manipulations is systematically investigated and determined with mental-effort based measures.

KEY WORDS: instruction; cognitive load; complex cognitive tasks.

INTRODUCTION

Learning and problem solving in complex cognitive domains, such as computer programming, mathematics, physics, and science, are typically constrained by the limited cognitive processing capacity of human memory. For novices, training tasks in these domains typically represent situations that are close to the limits of their capabilities, imposing high or excessive load on their cognitive system. Traditional instruction, which is based on the assumption that solving a wide range of goal-specific problems (i.e., conventional problems) is an effective way for novices to build expertise, provides no solution to this problem; it even seems to increase it.

¹Department of Instructional Technology, University of Twente.

²Correspondence should be addressed to Fred G. W. C. Paas, who is now at the Department of Movement Sciences, University of Limburg, P.O. Box 616, 6200 MD Maastricht, The Netherlands.

To solve conventional problems, novices use weak methods such as means-ends analysis (Anderson, 1987; Larkin *et al.*, 1980). In combination with the intrinsic complexity of the information in complex cognitive tasks, a heavy extraneous cognitive load is imposed in conventional problem solving by the search process for differences between problem states and goal states and by the operators for reducing these differences (Sweller, 1988). Consequently, novices cannot pay attention to relevant problem characteristics necessary for learning, such as the problem states previously acquired and the moves associated with those states. The adverse effects of high extraneous cognitive load on learning complex cognitive tasks, related to traditional instruction, are discussed in several recent articles (e.g., Chandler and Sweller, 1991, 1992; Paas, 1992; Paas and Van Merriënboer, 1994; Sweller *et al.*, 1990).

In this article, we argue that one of the primary problems that instructional technologists have to face in the design of training for complex cognitive tasks is to help learners cope with the cognitive load imposed by such tasks. The main cause of this problem is the incomplete knowledge about the quality and quantity of cognitive processes involved in learning these tasks. To remedy this problem, and to make instructional techniques less variable in their success, we propose that the current knowledge base on the interrelationships between instructional variables and cognitive load be extended and implemented in instructional design theories.

Current knowledge of the interrelationships between instructional variables and cognitive load is largely based on Sweller's (1988) cognitive load theory. In this theory, Sweller attributes the failure of traditional instruction for acquiring problem-solving skills to high cognitive load and inappropriate direction of attention. Across several studies, Sweller and colleagues collected data that supported the usefulness of cognitive load theory for instructional applications (Chandler and Sweller, 1991, 1992; Sweller, 1988, 1989; Sweller *et al.*, 1990; Ward and Sweller, 1990). The body of research evidence that resulted from these studies may be useful for solving the problem of developing effective instructional strategies for complex cognitive skills.

In the studies by Sweller and colleagues, cognitive load was determined by interpretations of performance- and task-based measures. In instructional research, it is not common to directly determine cognitive load during the process of instruction. We argue for research in which the cognitive load of instructional manipulations is systematically investigated with mental-effort based measures.

The primary goal of this article is to contribute to the realization of *instructional control of cognitive load* in the training of complex cognitive tasks. The organization is as follows: In the first section, the concept cog-

nitive load and its measurement techniques are described. Furthermore, this section provides an overview of the causes and effects of high or excessive cognitive load in the training of complex cognitive tasks. In the next section, the instructional prescriptions and variables that can be inferred from current knowledge on the relation between instruction and cognitive load are described, and some training strategies that have incorporated cognitive load are discussed. Special attention is paid to the implementation of the factor cognitive load in the design of practice. Finally, the state of affairs with regard to the role of cognitive load in instruction, and possible future research on this topic, are discussed.

THE CONCEPT OF COGNITIVE LOAD

Cognitive load is generally considered a multidimensional construct that represents the load that performing a particular task imposes on the cognitive system of a learner (e.g., Meshkati, 1988; Yeh and Wickens, 1988). Theoretically, the construct consists of causal factors and assessment factors, corresponding to factors that affect cognitive load and factors that are affected by cognitive load, respectively. With regard to its measurement, cognitive load can be conceptualized in the dimensions of mental load, mental effort, and performance (Jahns, 1973). Figure 1 shows a schematic representation of the construct cognitive load and its causal factors (left part of Fig. 1) and assessment factors (right part of Fig. 1).

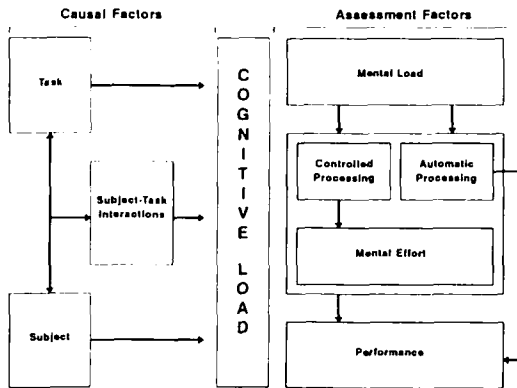


Fig. 1. Schematic representation of the construct cognitive load.

Causal Factors

Possible causal factors may refer to characteristics of the task (or to the environment in which the task is performed), to characteristics of the subject performing the task, or to interactions between both. With regard to task characteristics, examples of causal factors include task novelty, time pressure, and reward systems. Here, novel tasks, tasks performed under high time pressure, or tasks associated with punishment for incorrect performance are typically associated with high cognitive load. In addition, environmental factors such as high levels of noise or extreme temperatures may increase cognitive load.

Subject characteristics typically pertain to relatively stable factors; that is, factors that are not likely to experience sudden changes as a result of the task (or environment), such as subjects' cognitive capabilities, cognitive style, preferences, and prior knowledge.

Finally, the subject-task/environment interactions can affect cognitive load through relatively instable factors such as internal criteria of optimal performance, motivation, or state of arousal. For instance, the structuring of the task can interact with the cognitive style reflection-impulsivity. Jelsma and Pieters (1989) and Jelsma and Van Merriënboer (1989) compared "blocked" task structuring (i.e., grouping similar subtasks) to "randomized" structuring (i.e., performing subtasks in a random order) and found that reflective learners' cognitive load was relatively unaffected by task structure, whereas impulsive learners' cognitive load was relatively high for a random task structure.

Assessment Factors

With regard to the assessment factors, *mental load* is imposed by the task or environmental demands. This task-centered dimension, considered independent of subject characteristics, is constant for a given task in a given environment. For instance, suppose that there are two maze tasks A and B, and that maze A is more complex than maze B. Then, for all subjects solving the tasks, the mental load related to task A is higher than the mental load related to task B.

The human-centered dimension, *mental effort* refers to the amount of capacity or resources that is actually allocated to accommodate the task demands. Mental effort is believed to reflect the amount of controlled processing the individual is engaged in (Schneider and Shiffrin, 1977; Shiffrin and Schneider, 1977). The amount of invested mental effort comprises all three causal factors: task or environmental characteristics, subject charac-

teristics, and interactions between both. Returning to our two maze tasks, this means that task A will often show higher mental effort than task B because it is related with a higher mental load. But, on the other hand, this need not be true because subjects might brush aside task A and try very hard on task B, resulting in higher mental effort for task B than for task A. Finally, a group of subjects with no prior knowledge of the two mazes will show higher mental effort for task A, whereas a group of subjects with ample experience on task A and no experience on task B will show higher mental effort on task B.

The level of *performance* achieved is an indication of the third measurement dimension; it also reflects all three causal factors. To elaborate on our maze task example, this means that performance will usually be better on task B than task A (i.e., less errors, higher speed). But, by investing more mental effort (e.g., trying harder) subjects might reach equal or even higher performance on task A. In addition, some subjects might reach highest performance on task A and others on task B as a result of differences in experience or prior knowledge.

HIGH COGNITIVE LOAD: CAUSES AND EFFECTS

Several characteristics of complex cognitive tasks that may be responsible for high mental load will often yield high mental effort. In combination, this results in high cognitive load. Two important mental-load characteristics of complex cognitive tasks pertain to the number and nature of *component skills* involved (i.e., subskills that form part of the to-be-learned skill) and the complexity of the *goal hierarchies* of the problems that must be solved in the task domain (i.e., the progression of goals that must be accomplished to reach a solution). Component skills are subskills that form part of the to-be-learned skill; for instance, component skills for computer programming are analyzing problems, designing algorithms, composing solutions, applying the editor, testing implementations, and so forth. Goal hierarchies pertain to the progression of goals that must be accomplished to reach a solution; for example, the goal hierarchy for the simple programming problem to compute the mean of a series of inputs consists of the goals to count the number of inputs, to compute the sum of all inputs, and to divide the sum by the number of inputs. Obviously, subgoals may be further specified to reach each of those goals.

Skills containing many component skills will typically show higher processing demands than skills with less components. For instance, top-down approaches in computer programming are expected to reduce processing demands because less component skills have to be simultaneously

performed in each of the design phases. Moreover, the variability of the component skills involved is important. Complex cognitive tasks consist of both component skills that require consistent performance over problem situations ("recurrent components") and component skills that require variable performance over various problem situations ("nonrecurrent components"). Task demands are higher for skills that require relatively many nonrecurrent components.

For instance, in the field of computer programming recurrent skills involve the use of the editor and the compiler, the selection of the basic language commands, and the application of syntactic rules of the language; nonrecurrent skills may involve decomposition of programming problems into subproblems that are increasingly easier to solve and the composition of a program by putting functional parts of programming codes together. In statistical analyses, recurrent skills mainly involve the application of computational algorithms and the generation of applicable sequences of equations; nonrecurrent skills may involve strategic decisions about which statistical techniques, concepts, and principles to use and the coupling of the information that is present in the problem situations to the proper constants and variables.

A second, somewhat related characteristic that may produce high cognitive load is the complexity of the goal hierarchies of the problem solutions. In general, problem solutions can be viewed as hierarchies in which the top-level goal can only be accomplished by successfully attaining all lower-level (sub-)goals. A simple example of a goal-specific problem in the domain of computer programming is: Write a program to draw a square with side lengths of 50 mm. Writing programming codes to draw a straight line and to make a 90-degree turn can be considered as possible subgoals that have to be attained in order to accomplish the top-level goal of drawing the square.

To solve goal-specific problems, novices typically use cognitive-capacity demanding strategies, such as means-end analysis and hill climbing, when starting problem-solving in a new domain (Chi, Glaser, and Rees, 1982; Larkin *et al.*, 1980). Sweller (1988) has provided an elaborate example of (the cognitive consequences of) the means-ends analysis in the domain of physics. Both the number of subgoals and the number of possibilities to reach lower-level goals in alternative ways further increase the amount of mental effort that must be invested in the task. Using a computational model, Sweller (1988) showed that goal-specific problem-solving in the form of means-ends analysis needs more if-then productions than forward-working problem-solving strategies.

High cognitive load typically has negative effects on learning. For instance, Sweller (1988) showed that a means-ends strategy used by novices

to solve conventional trigonometry problems left little processing capacity for problem states previously acquired and the moves associated with those states. A large amount of the available capacity was required for simultaneously considering and making decisions about the current problem state, the goal state, differences between states, and problem-solving operators to be used to reduce these differences. He concluded that traditional instructional strategies, which stress the solving of many conventional, goal-specific problems, do not provide an effective way to learn. Evidence supporting this conclusion has been obtained in a wide variety of domains: computer programming (Van Merriënboer, 1990a, b; Van Merriënboer and De Croock, 1992; Paas and Van Merriënboer, 1994; Sweller *et al.*, 1990), mathematics (Cooper and Sweller, 1987; Paas, 1992; Sweller, 1989; Tarmizi and Sweller, 1988), and physics (Sweller, 1988; Ward and Sweller, 1990).

To conclude, the mediating role of the construct cognitive load in the relation between instruction and learning of complex cognitive tasks has been acknowledged in instructional research. However, in order to include the construct in instructional design models it is necessary to reliably measure cognitive load. The next section is concerned with this topic.

MEASUREMENT OF COGNITIVE LOAD

The question of how to determine cognitive load is difficult for researchers because of its multidimensional character. The interrelationships between performance, mental load, and mental effort are neither simple nor consistent (Tulga and Sheridan, 1980). This may be illustrated by the observation that students can compensate for an increase in mental load by increasing their mental effort, thereby maintaining performance at a constant level (but, at the cost of an increase in cognitive load). Furthermore, individuals must choose to engage in controlled processing. Therefore, instructional manipulations to change cognitive load can only be effective if subjects are willing to invest mental effort to accommodate task demands and, indeed, actually do so.

As a consequence, mental-effort measurement can reveal important information that is not necessarily reflected in performance and/or mental-load based measures. In combination with performance, measures of mental effort constitute the essence and the best estimator of cognitive load (Hamilton, 1979; Sanders, 1979; Paas, Van Merriënboer, and Adam, 1994). Two promising classes of techniques for measuring the effort expenditure can be identified; namely, techniques that use subjective indices (rating scales) and techniques that use psychophysiological indices (e.g., pupil diameter, heart-rate variability, event-related brain potentials).

Subjective rating-scale techniques are based on the assumptions that subjects are able to introspect on their cognitive processes and can report the amount of mental effort expenditure. Among others, Gopher and Braune (1984) found that subjects can introspect on their cognitive processes and have no difficulty in assigning numerical values to the imposed mental load or invested mental effort.

Psychophysiological techniques are based on the assumption that changes in cognitive functioning are reflected in physiological functioning. For example, controlled processing (i.e., intensity of mental effort) has been found to be related to a specific cardiovascular state that manifests itself in the heart-rate variability power-spectrum band from 0.07 to 0.14 Hz (i.e., mid-frequency band). The variability in this band is determined by a feedback mechanism connected with blood-pressure regulation. Heart-rate variability (i.e., spectral energy in the mid-frequency band) has been found to decrease with increasing mental effort (e.g., Aasman, Mulder, and Mulder, 1987; Mulder, 1988, 1992).

INSTRUCTIONAL CONTROL OF COGNITIVE LOAD

Two characteristics of complex cognitive tasks should be taken into account in the realization of instructional control of cognitive load. First, novices' learning and performance in complex cognitive tasks are severely constrained by their limited cognitive processing capacity. Second, as a rule, problem-solving in complex cognitive domains requires transfer of acquired knowledge and problem-solving skills; that is, the application of these knowledge and skills to problems that differ more or less from the ones trained in. The starting and ending point on this continuum have been referred to as near and far transfer (Mayer and Greeno, 1972).

Two related instructional goals can be derived from these characteristics. First, instruction should optimize cognitive load. For the training of complex cognitive tasks, this means that cognitive load should be substantially decreased. Second, instruction should redirect attention to aspects of learning that facilitate the transfer of acquired knowledge and skills. These two goals will be discussed in the following sections. Some examples also will be given of instructional strategies that may be used to reach these goals.

DECREASING COGNITIVE LOAD

Schneider and Shiffrin (1977; Shiffrin and Schneider, 1977) offer a general psychological model that is relevant to the question of how to de-

crease cognitive load. They make a distinction between two fundamental modes of information processing: automatic processing and controlled processing. The most important issue concerning the distinction between automatic and controlled processing pertains to the attentional requirements of the two processing modes.

Controlled processing can be characterized as slow, serial, and effortful; it typically occurs in novel and inconsistent processing tasks. It is flexible and available to analytic reflection, but because of capacity limitations its efficiency is highly dependent on mental load. Automatic processing, on the other hand, occurs without requiring active or conscious control or attention. It can be characterized as fast, parallel, and effortless; it typically occurs in well-practiced consistent tasks. During automatic processing, performance is believed to be relatively effortless, that is, insensitive to capacity limits.

The term *rule automation* may be used to refer to the transition from controlled to automatic processing; it can be seen as a lengthy, phase-wise process that evolves from purely controlled to purely automatic. Rule automation is mainly a function of practice; that is, less effort is required when a task is more extensively practiced. The underlying idea is that consistent practice builds "rules" or highly task-specific procedures (also called "productions," e.g., Anderson, 1983, 1987) that may directly control problem-solving behavior. However, it should be noted that the processing demands of *complex* cognitive tasks always reflect a combination of automatic and controlled processing components, because such tasks typically contain inconsistent aspects that cannot easily be automated.

The recently described four-component instructional design model of Van Merriënboer, Jelsma, and Paas (1992) builds on the controlled-automatic distinction and offers a general approach that incorporates the factor of cognitive load in the design of training for complex cognitive tasks. In this model (see Fig. 2 for a schematic representation), the process of building a training strategy for complex cognitive skills passes through an analysis phase and a design phase. The goal of the analysis phase is to be able to reduce cognitive load by first breaking down the complex task into component skills and required knowledge. The (most) relevant skills are selected and divided in *recurrent* and *nonrecurrent* component skills. Recurrent components are performed in a similar way over various problem situations; for instance, keyboard use and application of syntactical rules of a programming language are recurrent skills in computer programming because they are not dependent on the type of programming problem one is solving. *Nonrecurrent* components, on the other hand, vary considerably over problem situations; examples of such components in computer programming are decomposition of the programming problem or specification

of data structures. In addition to the description of recurrent and nonre-current components, the knowledge that is required to perform those skills is determined.

Representing the complex task in relevant component skills and knowledge is a *quantitative approach* that is appropriate for reducing cognitive load. A first aim of the design phase is to devise instruction that imposes relatively light cognitive load in such a way that learning and performance of the task are not influenced by cognitive-capacity limitations. To attain this goal, the 4C-model specifies instructional tactics that aim at the fast development of automatic processing for recurrent component skills. The instructional tactics define one instructional strategy (see Reigeluth, 1983) together with the instructional goals and conditions under which this strategy is useful.

A second aim of the design phase is to facilitate transfer. Therefore, a *qualitative approach* is offered to make controlled processing of nonre-current components more efficient, by providing instructional tactics that support the development of a rich knowledge base. Here, it is important to note that there is a complementary relation between the two aims of the design phase: The recurrent component skills must be automatized so that cognitive capacity can be devoted to more cognitive demanding (inconsistent) aspects of the total task (Fabiani *et al.*, 1989; Fisk and Gallini, 1989; Myers and Fisk, 1987; Schneider, 1985; Van Merriënboer, Jelsma,

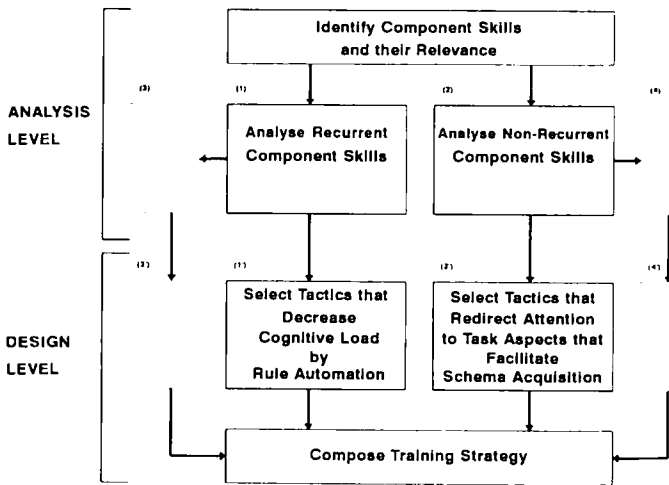


Fig. 2. The 4C instructional design model adapted to cognitive load theory (It should be noted that the dotted blocks represent components of the original 4C model that are not directly relevant in the cognitive load approach).

and Paas, 1992). This process will be described in more detail in the following section.

REDIRECTING ATTENTION

Apart from rule automation, *schema acquisition* can be identified as a process that is important in the acquisition of a complex cognitive skill (Van Merriënboer and Paas, 1990). Acquired schemata provide a rich knowledge base that makes controlled processing more efficient; thus, schemata are particularly important in the performance of nonrecurrent component skills. A schema can be conceptualized as a cognitive structure which has developed through experience with one or more problems and which enables problem solvers to recognize problems as belonging to a particular category requiring particular operations to reach a solution. Thus, schemata provide general knowledge that can be applied to particular cases. Recent research points out that the acquisition and the application of schemata is an effort-demanding process subject to strategic control (e.g., Anderson, 1987; Proctor and Reeve, 1988). Thus, learners must deliberately study problems and their solutions in order to abstract details and develop cognitive schemata; and they must also consciously apply those acquired schemata in order to solve new problems.

During practice, one may either create new schemata or adjust existing schemata to make them more in tune with experience. A more general schema may develop if a set of successful solutions of related problems is available, whereas the availability of a set of unsuccessful solutions of related problems will lead to a more specific schema. As knowledge is acquired in a particular domain, problem-solving becomes more schema driven as opposed to search driven. Throughout the problem-solving process, schemata may be acquired that offer analogies or abstract categories of problems and solutions, and that may guide subsequent problem-solving behavior. The completeness of a schema will influence the confidence in our own abilities to solve new problems. As our schemata enlarge, so our awareness of possibilities is thereby enlarged. So, better schema acquisition is self-rewarding. For example, due to the availability of schemata, expert programmers not only perform familiar programming tasks by the use of highly task-specific procedures, but also interpret unfamiliar situations in terms of their generalized knowledge. For instance, they may rely upon knowledge of computer technology to make their programs more efficient, knowledge of the design process in program development to guide their programming behavior, and knowledge of programming plans to improve problem decomposition and program composition.

To reach transfer of acquired knowledge and skills, both rule automation and schema acquisition should be facilitated. Automated procedures can provide identical elements that may help solve familiar aspects of new problems, and they can free up processing resources that may be devoted to controlled processes. For instance, operators in process control have the availability over a large number of routine procedures to detect particular system faults and to compensate for common system faults, enabling them to direct their attention to nonfamiliar aspects of troubleshooting tasks.

Acquired schemata, on the other hand, can provide analogies which may be used in new problem-solving situations by mapping processes to reach a solution. For instance, a process operator may be confronted with a system fault he has never before encountered; but his generalized knowledge may indicate a certain analogy with previously encountered faults and so guide his trouble shooting behavior. For example, in solving a new problem in the domain of programming the situation will be as follows. First, familiar aspects of the task (i.e., those aspects that are consistent over problem situations), such as proceeding in the programming environment, choosing the correct basic commands, and applying syntax can be performed by task-specific procedures. These procedures can be applied rapidly, without errors, and with little or no demands on processing capacity. Second, new aspects of the tasks can be solved by analogy. Schemata such as programming plans (learned programming language templates) should be available to help find a solution. These schemata can be interpreted, thanks to processing resources that are freed up by automation of the more familiar aspects of the programming task.

The relative importance of schema acquisition increases as the required transfer becomes further; that is, as the procedural overlap between transfer task and training task decreases, the availability of relevant schemata that may offer useful analogies becomes increasingly important (Jelma, Van Merriënboer, and Bijlstra, 1990).

As mentioned in the previous section, the 4C-model offers instructional tactics aimed at developing automatic processing for recurrent component skills in order to decrease cognitive load. With regard to the second aim, the model describes instructional tactics that redirect attention to aspects of the task that facilitate schema acquisition. This is particularly important for novice learners: In contrast to experts and intermediates, they cannot distinguish between relevant and irrelevant information. They cannot selectively allocate processing capacity by focusing attention to relevant aspects while ignoring irrelevant aspects (although intermediates and experts also process irrelevant aspects at least up to the level of recognition units). In this way, a rich knowledge base consisting of schemata will be

developed to support the performance of nonrecurrent components. Both aims are complementary to each other: automation of recurrent components frees up processing resources that may be redirected to task aspects that facilitate schema acquisition, which in turn favors the performance of nonrecurrent components.

EXEMPLARY TRAINING STRATEGIES

This section is primarily concerned with the design for training in complex cognitive tasks. A number of exemplary training strategies will be described that effectively decrease cognitive load and redirect attention to those aspects of learning that are expected to facilitate transfer. In order, they are the hierarchical approach, the emphasis-manipulation approach, goal-free problems, worked-out problems, the completion strategy, and expert-like problem analyses.

Hierarchical Approach

Frederiksen and White (1989) proposed a hierarchical approach to the design of practice. According to this approach, a set of "problem environments" must be devised to focus on particular component skills, concepts, and strategies that are important to the whole skill. Practice in the problem environments takes place in an order that follows from the hierarchical relations among these components. The hierarchical approach falls within the larger class of part-task training strategies. With these strategies individuals learn skill and knowledge components separately, and at the end of the training apply them to the whole task.

Frederiksen and White used the Space Fortress game, a complex task which involves the concurrent and coordinate use of perceptual and motor skills, and conceptual and strategic knowledge, in the service of multiple goals. Based upon the analysis of expert performance, Frederiksen and White developed a sequence of instructional games in which they implemented a hierarchy of primary objectives (e.g., detecting and destroying mines), optimal strategies for each objective (e.g., letting the mine hit the ship), and subskills which enable those strategies (e.g., identifying friend and foe mines). Subjects who were exposed to the hierarchical approach performed better on transfer tasks than control subjects who had to perform the whole task from the beginning of the learning process (see also Fabiani *et al.*, 1989).

Under this approach attention can be directed to the separate component skills and knowledge before training in the whole task. In this way, the training of task parts becomes an effective approach to reduce cognitive load.

Emphasis-Manipulation Approach

Gopher, Weil, and Siegel (1989) proposed an emphasis-manipulation approach, in which only the relative emphasis of selected subcomponents was manipulated, while leaving the whole task intact. The essence of this strategy is training under varying conditions of subcomponent priorities. The emphasis-manipulation approach falls within the larger class of whole-task training strategies. With these strategies learners practice the component skills and knowledge within the whole task context.

As in the experiment by Frederiksen and White (1989), the domain of Gopher and colleagues was the Space Fortress game. Gopher and colleagues conducted experiments with two types of emphasis manipulations: one which focused on the control dynamics of the ship, and one which concentrated on the handling of mines. From an analysis of goals and subgoals in the game and from an additive factor analysis, these two types of emphasis manipulations were identified as the two most important segments of the game.

Although Gopher and colleagues did not use transfer tests, they found that low-ability subjects who worked with the emphasis manipulation approach came close to the achievements of high-ability subjects in the control condition. The manipulation of the relative emphasis of selected subcomponents seems to be an effective way to direct attention and to reduce cognitive load. Gopher and colleagues suggested that training under multiple strategy changes may have given subjects a broader perspective of the task, improved their knowledge about the efficiency of their resources, and given them greater flexibility in adopting modes of response.

GOAL-FREE OR NONSPECIFIC GOAL PROBLEMS

A goal-free problem is one in which the goal is not specified. Whereas a specific goal in kinematics would look like, *What is the final velocity of the car?*, a nonspecific goal would look like, *Calculate the value of as many variables as you can.*

Using kinematics, geometry, and trigonometry problems, Sweller, Mawer, and Ward (1983), and Owen and Sweller (1985) demonstrated that

practice on goal-free problems enhances learning more than practice on conventional problems. In kinematics and geometry, Sweller (1988) found empirical evidence that problems that do not include specific goals reduce cognitive load, facilitate schema acquisition more than conventional problems, and lead to higher transfer of acquired skills.

Using verbal protocols and written solutions, these studies showed that goal-free problems redirect learners' attention from a means-ends strategy to a strategy in which they are forced to work forward from the givens. While working forward, subjects calculate values by substituting given or previously calculated values into equations that contain only a single unknown. If the goal is not specified, it is not possible to work backward from the goal by constructing a chain of subgoals that are connected to the givens. Sweller (1988) also constructed a computational model that provided evidence that cognitive load is higher under a means-ends strategy than under a goal-free strategy. Compared to a goal-free strategy the means-ends analysis results in more decisions and more information that must be considered when making decisions.

Worked-Out Problems

A worked-out problem or worked example is a problem with a written-out, well-structured solution that provides a best example to the students under consideration. Worked examples can be annotated with information about what they are supposed to illustrate (Anderson, Boyle, Corbett, and Lewis, 1986). A worked-out problem can recapitulate the structure of the abstract concept, is an efficient means to make the tacit knowledge of experts explicit, and provides the possibility to extract general principles from a well-structured problem solution. Therefore, worked-out problems can be used as a kind of concrete schemata to map new solutions and at the same time foster schema acquisition.

Practice on worked-out problems is an effective way to optimize training for transfer. In the context of algebra-manipulation problems, Sweller and Cooper (1985) showed that studying worked examples facilitated problem-solving performance more than actually solving the equivalent problems. Using algebra transformation problems, Cooper and Sweller (1987) showed that subjects whose training emphasized worked examples were better able to solve both similar and transfer problems than subjects whose training consisted of conventional examples. Zhu and Simon (1987) showed that a 3-year mathematics course could be completed in 2 years by emphasizing worked examples.

According to Sweller's (1988) cognitive load theory, a critical feature of worked examples is that they appropriately direct attention. In worked-out problems, students are prevented from using weak problem-solving methods. Therefore, they do not have to perform goal-irrelevant operations, which may divert attention from the more relevant aspects of the problem solution. Another critical feature of worked examples is that they impose relatively light cognitive load. Cognitive load is decreased by the redirection of attention to task aspects that facilitate schema acquisition (e.g., stereotyped solutions for particular (sub-)goals, relations between task components, etc.) and by the amount of guidance that can be offered in effectively structured worked-out problems.

Completion Strategy

Completion strategy, introduced by Van Merriënboer (1990b) in the domain of elementary computer programming, is the completion of increasingly larger parts of incomplete solutions. A completion assignment consists of a problem with a partly written out well-structured solution that has to be completed. It can be considered as an intermediate form between conventional and worked-out problems.

In the domain of computer programming, Van Merriënboer (1990a) and Van Merriënboer and De Croock (1992) found that a strategy emphasizing the completion of existing programs results in superior use of programming templates (i.e., a kind of schemata that describes stereotyped solutions in the domain of programming), compared to a strategy that emphasizes the generation of new programs (see Fig. 3 for an example of a completion assignment). In the domain of statistics, Paas (1992) found that a completion strategy results in better performance on a near- and far-transfer test than a conventional strategy.

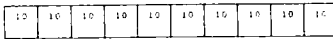
Completion problems can be very similar to worked-out problems. Therefore, the advantages of worked-out problems also apply to completion problems. The completion assignments can be used to direct learners' attention to aspects of the problem that are important for the acquisition of schemata. The extent to which the problem solution is worked out determines the amount of cognitive load.

Expert-like Problem Analyses

With this technique, students are constrained to carry out qualitatively and hierarchically structured problem analyses that consist of answering a series of qualitative questions. The analyses are designed to mimic the types

Problem Specification

Write a program to draw a series of a pre-specified number of squares. The total length of the series must be 100 units. Examples of desired outputs are



Partial Solution

```

(01) PROC series(100:number)
(02)   right(90)
(03)   FOR count := 1 TO number DO
(04)     forward(100:number)
(05)   ENDFOR count 1
(06) ENDPROC
    
```

Information

The partial solution presents another example on the use of the FORE-loop. Remember that it should be used in case you want to repeat one or more program lines a fixed number of times.

- In previous programs, fixed meant a constant number of times (e.g., 4 or 65).
- In the presented partial solution, fixed means a number of times that is pre-specified as an argument to the procedure-call.

Question(s)

The procedure-call series(100/5) on the partial solution yields

- a. one square with side lengths of 20 units
- b. one square with side lengths of 100 units
- c. one line with a length of 100 units
- d. one line with a length of 500 units

Task Instruction(s)

1. Load the partial solution from disk and observe its output for different procedure-calls.
2. Complete the program to meet the given problem specification.

Fig. 3. A simplified example of a completion example, using the programming language Comal-80 (Cristensen, 1982).

of analyses carried out by experts, prior to problem-solving. Questions become increasingly problem specific as the analysis proceeds.

In the domain of elementary classical mechanics, Dufresne *et al.* (1992) presented questions to novice physics students via a computer-based environment called the Hierarchical Analysis Tool (HAT). In this menu-driven environment, questions are ordered according to a hierarchy, whereby the questions become increasingly problem specific as the analysis proceeds. The HAT does not present content matter, tutor, or provide feedback to the user; it allows the user to practice a top-down problem-solving approach. The HAT can be thought of as an elaborate, hierarchical tree-like structure in which the exact path that a user takes to analyze a problem depends on the selections made at each node along the way. Dufresne *et al.* (1992) showed that novices who had performed the expert-like analyses were better able to make expert-like judgments of solution similarity and to solve problems than were novices who involved in conventional problem-solving. Dufresne *et al.* (1992) argued that the analyses could redirect attention from the use of means-ends analysis to higher-order domain knowledge, thereby reducing cognitive load.

CONCLUSIONS AND DISCUSSION

One of the main problems in conventional instructional design for complex cognitive tasks is the lack of control of cognitive load. Although cognitive load has been assigned an important role in the instruction of complex cognitive tasks, its measurement during the process of instruction

is uncommon. Furthermore, research efforts have been almost exclusively concerned with performance- and mental-load based measures. We suggest that the combined use of performance and mental-effort based measures is a more appropriate way to estimate cognitive load and to extend the knowledge base regarding the interrelationships between instruction and cognitive load. Moreover, Paas and Van Merriënboer (1993) have shown that a composite measure of performance z-scores and mental-effort z-scores can be used in instructional research to reflect the efficiency of instructional conditions.

This article was concerned with the design of strategies that effectively enable control of cognitive load in the training for transfer of complex cognitive skills. Two related goals were identified for its realization: (a), the optimization (decreasing) of cognitive load by the automation of recurrent component skills; and (b), the redirection of attention to those task aspects that facilitate schema acquisition. The 4C instructional design model of Van Merriënboer *et al.* (1992), adapted to Sweller's (1988) cognitive load theory, offers a useful approach for specifying training strategies for complex cognitive tasks. We presented examples of such strategies in terms of their effects on learning and transfer and their possibilities to redirect attention and reduce cognitive load. The presented framework offered the opportunity to explain research results and to make predictions about the effects of training methods on transfer.

The positive effects of training strategies on transfer can be explained by their ability to redirect attention and decrease cognitive load. Although these strategies have similar effects, they differ in how their effects come about, in their priority for redirecting attention or reducing cognitive load, and in their operating level. With respect to, how their effects come about, a division can be made between "*process approaches*" and "*product approaches*."

Process approaches, such as the hierarchical approach and expert-like problem analyses, try to attain the goal of transfer by analyzing experts' performance, and then constraining novices to mimic experts' behavior.

Product approaches, such as worked examples and the completion strategy, focus on attaining the end product of transfer, without considering the experts' approach. Part-whole task training strategies redirect attention and the remaining strategies reduce cognitive load.

Another difference between the strategies is their operating level. The part-whole training strategies and the expertlike problem analyses are global or high-level approaches. Goal-free or worked examples are more specific or low-level approaches that may form part of, for example, part-whole strategies.

We have shown that interrelationships between instruction and cognitive load can result in innovative practice strategies. In particular, these practice strategies facilitate transfer by (a) redirecting attention to task aspects that facilitate schema acquisition, and by (b) reducing cognitive load. However, there is often no uniformity in the design of these individual strategies. For instance, Sweller *et al.* (1990) and Ward and Sweller (1990) showed that worked examples that required students to mentally integrate mutually referring, disparate sources of information, were no better than conventional examples. So, cognitive load theory needs further specification.

Despite the growing body of research evidence for Sweller's cognitive load theory, instruction for complex cognitive tasks will remain equivocal as long as the cognitive load of instructional manipulations is not systematically determined on the basis of mental-effort and performance measures. This conclusion should be a challenge for future instructional research. More insight in the cognitive costs of instruction would imply more control of cognitive load. Then, training strategies can be adapted to the cognitive demands of a task, situations of cognitive overload and misdirection of attention can be prevented, and predictions about the strategies can be made that aim at maximizing performance efficiency. Eventually, a cognitive-load oriented theory for the instruction of complex cognitive tasks may emerge from this research.

REFERENCES

- Aasman, J., Mulder, G., and Mulder, L. J. M. (1987). Operator effort and the measurement of heart-rate variability. *Hum. Factors* 29: 161-170.
- Anderson, J. R. (1983). *The Architecture of Cognition*, The Harvard University Press, Cambridge.
- Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychol. Rev.* 94: 192-210.
- Anderson, J. R., Boyle, C. F., Corbett, A., and Lewis, M. (1986). Cognitive Modelling and Intelligent Tutoring. Carnegie-Mellon University, Psychology Department, Pittsburgh, PA (Tech. Rep. No. ONR-86-1).
- Chandler, P., and Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognit. Instr.* 8: 293-332.
- Chandler, P., and Sweller, J. (1992). The split-attention effect as a factor in the design of instruction. *Brit. J. Educat. Psychol.* 62: 233-246.
- Chi, M., Glaser, R., and Rees, E. (1982). Expertise in problem-solving. In Sternberg, R. (ed.), *Advances in the Psychology of Human Intelligence*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 7-75.
- Cooper, G., and Sweller, J. (1987). Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *J. Educat. Psychol.* 79: 347-362.
- Cristensen, B. R. (1982). *Beginning Comal*, Ellis Horwood, Chichester.

- Dufresne, R. J., Gerace, W. J., Hardiman, P. T., and Mestre, J. P. (1992). Constraining novices to perform expertlike problem analyses: Effects on schema acquisition. *J. Learning Sci.* 2: 307-331.
- Fabiani, M., Buckley, J., Gratton, G., Coles, M. G. H., Donchin, E., and Logie, R. (1989). The training of complex task performance. *Acta Psychol.* 71: 259-299.
- Fisk, A. D., and Gallini, J. K. (1989). Training consistent components of tasks: Developing an instructional system based on automatic/controlled processing principles. *Hum. Factors* 31: 453-463.
- Frederiksen, J. R., and White, B. Y. (1989). An approach to training based upon principled task decomposition. *Acta Psychol.* 71: 89-146.
- Gopher, D., and Braune, R. (1984). On the psychophysics of workload: Why bother with subjective measures? *Hum. Factors* 26: 519-532.
- Gopher, D., Weil, M., and Siegel, D. (1989). Practice under changing priorities: An approach to the training of complex skills. *Acta Psychol.* 71: 147-177.
- Hamilton, P. (1979). Process entropy and cognitive control: Mental load in internalized thought processes. In Moray, N. (ed.), *Mental workload: Its Theory and Measurement*, Plenum, New York, pp. 289-298.
- Jahns, D. W. (1973). A Concept of Operator Workload in Manual Vehicle Operations. Forschungsinstitut Anthropotechnik, Meckenheim, West Germany (Technical report No. 14).
- Jelsma, O., and Pieters, J. M. (1989). Practice schedule and cognitive style interaction in learning variants of a maze task. *J. Appl. Cognit. Psychol.* 3: 73-83.
- Jelsma, O., and Van Merriënboer, J. J. G. (1989). Contextual interference: Interactions with reflection-impulsivity. *Percept. Motor Skills* 68: 1055-1064.
- Jelsma, O., Van Merriënboer, J. J. G., and Bijlstra, J. (1990). The ADAPT design model: Towards instructional control of transfer. *Instr. Sci.* 19: 89-120.
- Larkin, J., McDermott, J., Simon, D., and Simon H. (1980). Models of competence in solving physics problems. *Cognit. Sci.* 4: 317-348.
- Mayer, R. E., and Greeno, J. G. (1972). Structural differences between learning outcomes produced by different instructional methods. *J. Educat. Psychol.* 63: 165-173.
- Meshkati, N. (1988). Toward development of a cohesive model of workload. In Hancock, P. A., and Meshkati, N. (eds.), *Human Mental Workload*, pp. 305-314.
- Myers, G. L., and Fisk, A. D. (1987). Application of automatic and controlled processing theory to industrial task training: The value of consistent component training. *Hum. Factors* 29: 255-268.
- Mulder, L. J. M. (1988). Assessment of Cardiovascular Reactivity by Means of Spectral Analysis. Unpublished doctoral dissertation, University of Groningen, Groningen, The Netherlands.
- Mulder, L. J. M. (1992). Measurement and analysis methods of heart rate and respiration for use in applied environments. *Biol. Psychol.* 34: 205-236.
- Owen, E., and Sweller, J. (1985). What do students learn while solving mathematics problems? *J. Educat. Psychol.* 77: 272-284.
- Paas, F. G. W. C. (1992). Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *J. Educat. Psychol.* 84: 429-434.
- Paas, F. G. W. C., and Van Merriënboer, J. J. G. (1993). The efficiency of instructional conditions: An approach to combine mental-effort and performance measures. *Hum. Factors* 35: 737-743.
- Paas, F. G. W. C., and Van Merriënboer, J. J. G. (1994). Variability of worked examples and transfer of geometrical problem-solving skill: A cognitive-load approach. *J. Educat. Psychol.* 86: 122-133.
- Paas, F. G. W. C., Van Merriënboer, J. J. G., and Adam, J. J. (1994). Measurement of cognitive load in instructional research. *Percept. Motor Skills* 79: 419-430.
- Proctor, R. W., and Reeve, T. G. (1988). The acquisition of task-specific productions and modification of declarative representations in spatial pre-cuing tasks. *J. Exp. Psychol.: General* 117: 182-196.

- Reigeluth, C. M. (1983). Instructional design: What is it and why is it? In Reigeluth, C. M. (ed.), *Instructional Design Theory and Models*, Erlbaum Associates, Hillsdale, NJ, pp. 3-36.
- Sanders, A. F. (1979). Some remarks on mental load. In Moray, N. (ed.), *Mental Workload: Its Theory and Measurement*, Plenum, New York, pp. 41-77.
- Schneider, W. (1985). Training high-performance skills: Fallacies and guidelines. *Hum. Factors* 27: 285-300.
- Schneider, W., and Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. Detection, search, and attention. *Psychol. Rev.* 84: 1-66.
- Shiffrin, R. M., and Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychol. Rev.* 84: 127-190.
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognit. Sci.* 12: 257-285.
- Sweller, J. (1989). Cognitive technology: Some procedures for facilitating learning and problem-solving in mathematics and science. *J. Educat. Psychol.* 4: 457-466.
- Sweller, J., and Cooper, G. A. (1985). The use of worked examples as a substitute for problem-solving in learning algebra. *Cognit. Instr.* 2: 59-89.
- Sweller, J., Chandler, P., Tierney, P., and Cooper, M. (1990). Cognitive load as a factor in the structuring of technical material. *J. Exp. Psychol.: General* 119: 176-192.
- Tarmizi, R. A., and Sweller, J. (1988). Guidance during mathematical problem-solving. *J. Educat. Psychol.* 80: 424-436.
- Tulga, M. K., and Sheridan, T. B. (1980). Dynamic decisions and workload in multitask supervisory control. *IEEE Trans. Syst., Man, Cybernet.* 10: 217-232.
- Van Merriënboer, J. J. G. (1990a). Strategies for programming instruction in high school: Program completion vs. program generation. *J. Educat. Comp. Res.* 6: 265-287.
- Van Merriënboer, J. J. G. (1990b). *Teaching Introductory Computer Programming: A Perspective from Instructional Technology*, Bijlstra and Van Merriënboer, Enschede, The Netherlands.
- Van Merriënboer, J. J. G., and De Croock, M. B. M. (1992). Strategies for computer-based programming instruction: Program completion vs. program generation. *J. Educat. Comp. Res.* 8: 365-394.
- Van Merriënboer, J. J. G., and Paas, F. G. W. C. (1990). Automation and schema acquisition in learning elementary computer programming: Implications for the design of practice. *Comp. Hum. Behav.* 6: 273-289.
- Van Merriënboer, J. J. G., Jelsma, O., and Paas, F. G. W. C. (1992). Training for reflective expertise: A four-component instructional design model for complex cognitive skills. *Educat. Technol. Res. Devel.* 40: 23-43.
- Ward, M., and Sweller, J. (1990). Structuring effective worked examples. *Cognit. Instr.* 7: 1-39.
- Yeh, Y.-Y., and Wickens, C. D. (1988). Dissociation of performance and subjective measures of workload. *Hum. Factors* 30: 111-120.
- Zhu, X., and Simon, H. (1987). Learning mathematics from examples and by doing. *Cognit. Instr.* 4: 137-166.