Blueprints for Complex Learning: The 4C/ID-Model

☐ Jeroen J. G. van Merriënboer
Richard E. Clark
Marcel B. M. de Croock

This article provides an overview description of the four-component instructional design system (4C/ID-model) developed originally by van Merriënboer and others in the early 1990s (van Merriënboer, Jelsma, & Paas, 1992) for the design of training programs for complex skills. It discusses the structure of training blueprints for complex learning and associated instructional methods. The basic claim is that four interrelated components are essential in blueprints for complex learning: (a) learning tasks, (b) supportive information, (c) just-in-time (JIT) information, and (d) part-task practice. Instructional methods for each component are coupled to the basic learning processes involved in complex learning and a fully worked-out example of a training blueprint for "searching for literature" is provided. Readers who benefit from a structured advance organizer should consider reading the appendix at the end of this article before reading the entire article.

☐ The instructional design enterprise is a bit like an ocean liner—huge, slow, ponderous, and requiring large amounts of energy and a great deal of time to move it even one degree off its current path. Recent discussions and developments in the field concern rapid technological and societal changes and the resulting need for very complex knowledge at work (Berryman, 1993; Cascio, 1995); new constructivist design theories for problem solving (Jonassen, 1994; Reigeluth, 1999a; Schwarz, Brophy, Lin, & Bransford, 1999); arguments for new context and technology-based design (Driscoll & Dick, 1999; Kozma, 2000; Richey, 1998); two decades of systematic design research and development by John Anderson (1983, 1993; Anderson & Lebiere, 1998), and innovative work on "first principles of instruction" by designer-researcher David Merrill (2000). These welcome discussions have at least one important goal in common-the gradual evolution of design theory to accommodate complex learning. Future design theory should support the development of training programs for learners who need to learn and transfer highly complex cognitive skills or "competencies" to an increasingly varied set of real-world contexts and settings. In addition, adequate design for complex skills helps overcome findings that under some conditions, inadequate design may cause learning problems (Clark, 1988).

The 4C/ID-model proposed in this article addresses at least three deficits in previous instructional design models. First, the 4C/ID-model focuses on the integration and coordinated performance of task-specific constituent skills rather than on knowledge types, context or presentation-delivery media. Second, the model makes a critical distinction between supportive information and required just-in-time (JIT) information (the latter specifies the performance required, not only the type of knowledge required). And third, traditional models use either part-task or whole-task practice; the 4C/ID-model recommends a mixture where part-task practice supports very complex, "whole-task" learning.

Novices learn complex tasks in a very different way than they do simple tasks. Evidence for this claim can be found in research on learning concepts (Corneille & Judd, 1999), verbal information (Pointe & Engle, 1990), mathematics (Wenger & Carlson, 1996), visual comparison tasks (Pellegrino, Doane, Fischer, & Alderton, 1991) and a variety of complex work skills (Ackerman, 1990), among others. Most design models emphasize instruction in relatively simple learning tasks and assume that a large, complex set of interrelated tasks are achievable as "the sum of the parts"—by sequencing a string of simplified, component task procedures until a complex task is captured. There is overwhelming evidence that this does not work (see van Merriënboer, 1997, for an in-depth discussion of these issues). Existing design models most often assume that knowledge of simple task performance, once acquired, transfers reliably to novel future problems despite considerable evidence to the contrary (e.g., Clark & Estes, 1999; Perkins & Grotzer, 1997).

These relatively new insights about complex learning are presented in a design theory developed originally by van Merriënboer and others in the early 1990s (van Merriënboer, Jelsma, & Paas, 1992). The complete design system and its psychological backgrounds are described in van Merriënboer (1997; see also van Merriënboer & Dijkstra, 1996, for its theoretical basis). This article presents an overview of the most recent version of the design theory, called 4C/ID. It is a version of the model that currently provides the basis for the development of computer-based design tools in a European project called ADAPT[IT] (Advanced Design Approach for Personalized Training—Interactive Tools).

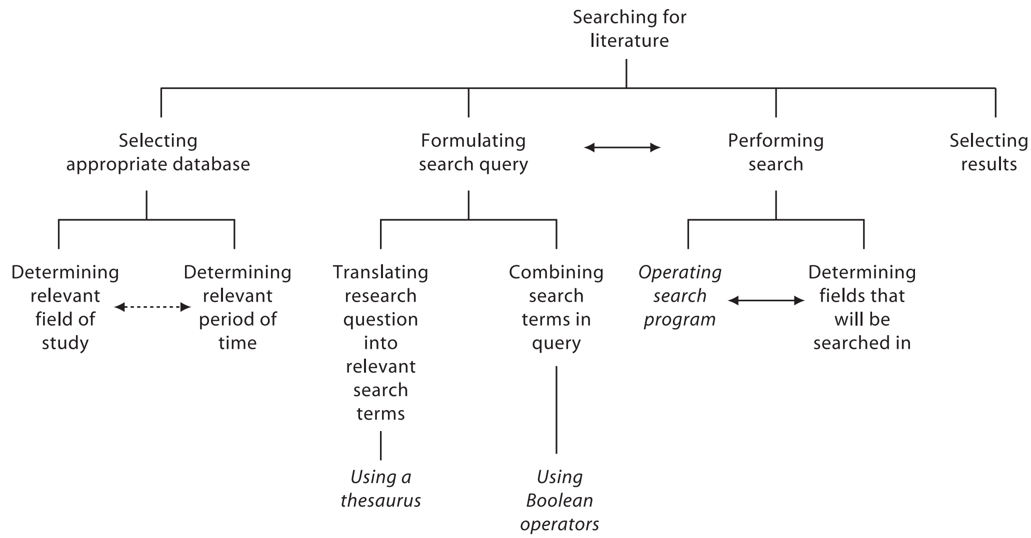
An overview of the 4C/ID-model is given in three parts. First, the elements of complex learning that must be accommodated in design are described conceptually, using a concrete example of the skills necessary to search for documents in a computerized database. Second, a description is presented of the four "blueprint components" (4C) that support complex learning, namely (a) learning tasks; (b) supportive information; (c) JIT information, and (d) part-task practice. Instructional methods are illustrated for each component. Finally, the use of the model for designing adaptive instruction is discussed and some empirical studies that support the effectiveness of the model are briefly reviewed. We will also briefly discuss cognitive task analysis as a method for capturing advance expertise as content for complex training.

COMPLEX LEARNING

Complex learning is always involved with achieving integrated sets of learning goals—multiple performance objectives. It has little to do with learning separate skills in isolation, but it is foremost dealing with learning to coordinate and integrate the separate skills that constitute real-life task performance. Thus, in complex learning the whole is clearly more than the sum of its parts because it also includes the ability to coordinate and integrate those parts. As an illustration, Figure 1 provides a simple description of the *constituent skills* that make up the moderately complex cognitive skill, "searching for relevant research literature." A well-designed training program for complex learning will not aim at trainees' acquiring each of these constituent skills separately, but will instead try to achieve that the trainees acquire the ability to use all of the skills in a coordinated and integrated fashion while doing real-life literature searches.

The skills hierarchy in Figure 1 depicts the two fundamental types of relations between constituent skills that must be taken into account when designing a training program (cf. Gagné's "learning hierarchy," Gagné, Briggs, & Wager, 1992). First, there is a horizontal relationship between coordinate skills that is indicated from left to right. This relationship can be temporal (e.g., you first select an appropriate database and then

Figure 1 ☐ Skills hierarchy for the moderately complex skill "searching for relevant research literature." Nonrecurrent skills are represented in roman font, recurrent skills in italics. Double horizontal arrows with a solid line represent a simultaneous relationship; double horizontal arrows with a dotted line represent a transposable relationship (see text).



formulate the search query for the selected database), simultaneous (e.g., you concurrently formulate a search query and perform the search until you have a relevant and manageable list of results), or transposable (e.g., determining the relevant field of study and determining the relevant period of time can be done in any order or even simultaneously). The second type of relation is the vertical relationship, which is indicated from bottom-to-top between child skills on a certain level and their parent skill one level higher. This relationship signifies that constituent skills lower in the hierarchy enable or are prerequisite to the learning and performance of skills higher in the hierarchy (e.g., you must be able to operate a search program in order to be able to perform a search). In an intertwined hierarchy, additional relations between constituent skills that are important for training design may be added. For instance, similarity relations may indicate constituent skills that are easily mixed up.

Figure 1 also illustrates a typical characteristic of complex learning outcomes. Namely, for expert task performers, there are qualitative differences between constituent skills involved. Some constituent skills are performed in a variable way from problem to problem situation. For instance, formulating a search query involves problem solving and reasoning in order to cope with the specific requirements of each new search. Experts can effectively perform such constituent skills because they have highly complex cognitive schemata available that help them to reason about the domain and to guide their problem solving. Thus, schemata enable *another* use of the same knowledge in a new problem situation, because they contain generalized knowledge, or concrete cases, or both, that can serve as an analogy.

Other constituent skills lower in the hierarchy may be performed in a highly consistent way from problem to problem situation. For instance, operating the search program is a constituent skill that does not require reasoning or problem solving. Experts can effectively perform such constituent skills because their schemata contain rules that directly associate particular characteristics of the problem situation to particular actions. In other words, rules enable the *same* use of identical, situation-specific knowledge in a new problem situation.

Experts may even reach a level of performance where they operate the search program fully *automatically* (unconsciously, without mental effort). Conscious control is then no longer required because the rules have become automated. The result is that trained experts are able to focus their attention on other, nonautomatic constituent skills while operating the search program.

Training programs for complex learning should pay attention not only to the coordination and integration of constituent skills, but also to these qualitative differences in desired exit behavior of constituent skills. In order to identify these qualitatively different performance objectives, constituent skills are classified as either nonrecurrent or recurrent. For *nonrecurrent* (novel, effortful) constituent skills the desired exit behavior varies from problem to problem situation, and is guided by cognitive schemata that steer problem-solving behavior (cognitive strategies) and allow for reasoning about the domain (mental models). For *recurrent* (routine) constituent skills the desired exit behavior is highly similar from problem to problem situation, and is driven by rules that link particular characteristics of the problem situation to particular actions. For instance, constituent skills that may be classified as recurrent for searching for relevant research literature are using thesauri, using Boolean operators, and operating the search program (in Figure 1, these recurrent skills are printed in *italics*). This classification is particularly important because the dominant learning processes for nonrecurrent constituent skills are fundamentally different from those for recurrent constituent skills.

For the nonrecurrent aspects of a complex skill and the complex skill as a whole, the main learning processes that must be promoted are related to *schema construction*. From the viewpoint of practice, learners should be encouraged to mindfully abstract away from the concrete experiences that are offered to them. Schemata are then actively (re-)constructed in order to make them more in agreement with concrete experiences. This process of *induction* is central to the design of learning tasks that offer the concrete experiences in a training program for complex learning. From the viewpoint of information presentation, learners should be encouraged to connect newly presented information to already existing schemata, that is, to what they already know. This way, schemata are (re-)constructed and embellished with the new information that is relevant to learning and performing the skill. This process of *elaboration* is central to the design of information that helps learners to perform the nonrecurrent aspects of a complex skill. It is called "supportive information."

For the recurrent aspects of a complex skill, the main learning processes that must be promoted are related to what is called *rule automation*. Automation is mainly a function of the amount and quality of practice that is provided to the learners and eventually leads to automated rules that directly control behavior. Rules are formed in two processes: First *compilation*, which embeds specific knowledge or information in the rules (proceduralization) and chunks rules together that are consistently applied in the same order (composition), and second *strengthening*, which increases the strength of a rule each time it is successfully applied (Anderson, 1983, 1993, Anderson & Lebiere, 1998). The processes of compilation and, especially, subsequent strengthening are central to the design of part-task practice, which offers additional practice for selected recurrent constituent skills in a training program for complex learning. From the viewpoint of information presentation, it is important to present to-be-proceduralized information precisely when learners need it during practice. This process of *restricted encoding* of new information into cognitive rules is central to the design of information that helps learners to learn and perform the recurrent aspects of a complex skill. It is called JIT information and presented to learners during their work on learning tasks and during part-task practice.

To sum up, a training program for complex learning must pay attention to the integration and coordination of all skills that constitute a complex cognitive skill (i.e., integrated objectives), and concurrently promote schema construction for nonrecurrent aspects and rule automation for recurrent aspects of the complex skill. By doing so, the training program aims at transfer of learning—the ability to apply the complex cognitive skill in a wide variety of new real-life situations. The familiar aspects that learners encounter in transfer situations can be dealt with thanks to the availability of rules, which also free up cognitive resources that may be used to handle the unfamiliar aspects of the transfer tasks. This process of rule-based transfer complies with the "component fluency hypothesis" (e.g., Carlson, Khoo, & Elliot, 1990). Furthermore, unfamiliar aspects can be dealt with thanks to the availability of complex cognitive schemata, which may be interpreted in order to guide the problem-solving process and to reason about the domain. This process of schema-based transfer produces reasonably effective behavior for unfamiliar aspects of a problem situation. The combination of the two transfer processes is believed to allow for *reflective expertise* because complex schemata may also be used to monitor and evaluate one's own performance, including a reflection on the quality of solutions reached by the application of rules. This complies with the "understanding hypothesis" (e.g., Ohlsson & Rees, 1991; see van Merriënboer, 1997, for a complete discussion). Building on these assumptions, the next section describes the main blueprint components of training programs aimed at complex learning.

THE FOUR BLUEPRINT COMPONENTS

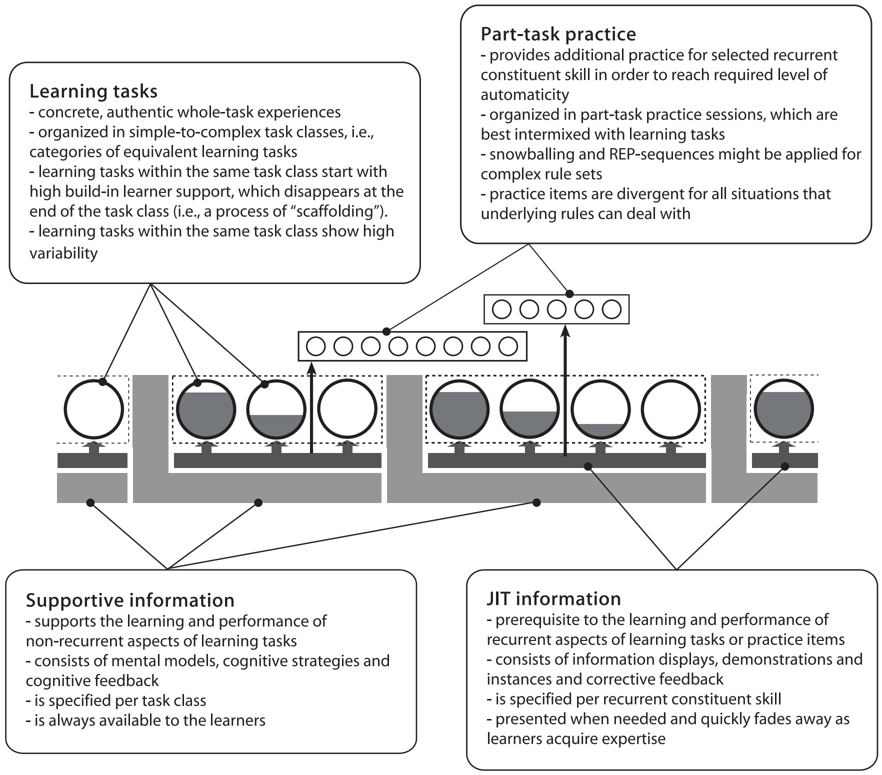
The basic message of the 4C/ID-model is that environments for complex learning can always be described in terms of four interrelated blueprint components. These components are based on the four categories of learning processes that are central to complex learning:

1. Learning Tasks: concrete, authentic, whole-task experiences that are provided to learners in order to promote schema construction for nonrecurrent aspects and, to a certain degree, rule automation by compilation for recurrent aspects. Instructional methods primarily aim at induction, that is, constructing schemata through mindful abstraction from the concrete experiences that are provided by the learning tasks.
2. Supportive Information: information that is supportive to the learning and performance of nonrecurrent aspects of learning tasks. It provides the bridge between learners' prior knowledge and the learning tasks. Instructional methods primarily aim at elaboration, that is, embellishing schemata by establishing nonarbitrary relationships between new elements and what learners already know.
3. JIT Information: information that is prerequisite to the learning and performance of recurrent aspects of learning tasks. Instructional methods primarily aim at compilation through restricted encoding, that is, embedding procedural information in rules. JIT information is not only relevant to learning tasks but also to:
4. Part-task Practice: practice items that are provided to learners in order to promote rule automation for selected recurrent aspects of the whole complex skill. Instructional methods primarily aim at rule automation, including compilation and subsequent strengthening to reach a very high level of automaticity.

Component 1: Learning Tasks

A sequence of learning tasks is the backbone of every training program aimed at complex learning (see Figure 2, which represents the learning tasks as circles). The learning tasks are typically performed in a real or simulated task environment and provide whole-task practice: *ideally*, they confront the learners with all constituent skills that make up the whole complex skill. It is important to stress that learning tasks should engage learners in activities that require them to work with the constituent skills, this as opposed to activities in which they have to study general information about or related to the skills. For the nonrecurrent aspects of the complex skill and the complex skill as whole (which is always nonrecurrent), learning tasks promote schema construction by inductive processing. That is, the learning tasks stimulate learners to construct cognitive schemata by mindfully abstracting away from the concrete experiences that the learning tasks provide. Learning processes like

Figure 2 ☐ A graphical view on the four components: (a) learning tasks, (b) supportive information, (c) just-in-time (JIT) information, and (d) part-task practice.



generalization and discrimination subsequently reconstruct schemata to make them more in accordance with new experiences. To-be-constructed schemata come in two forms: (a) *mental models* that allow for reasoning in the domain because they reflect the way in which the learning domain is organized, and (b) *cognitive strategies* that guide problem solving in the domain because they reflect the way problems may be effectively approached.

Task classes. It is clearly impossible to provide highly complex learning tasks right from the start of the training program because this would yield excessive cognitive overload for the learners, which impairs learning and performance (Sweller, van Merriënboer, & Paas, 1998). Thus, learners will typically start their work on relatively simple learning tasks and progress toward more complex tasks. Complexity is affected by the number of constituent skills involved, the number of interactions between constituent skills, and the amount of knowledge necessary to perform the constituent skills. Task classes are used to define simple-to-complex categories of learning tasks and to steer the process of selection and development of suitable learning tasks (see the dotted lines around the circles in Figure 2). *Task classes* and not the individual learning tasks define the basic sequence of a training program developed according to the 4C/ID-model. Learning tasks within a particular task class are equivalent in the sense that the tasks can be performed on the basis of the same body of knowledge (i.e., mental models and cognitive strategies). A more complex task class requires more knowledge or more elaboration of knowledge for effective performance (cf., The Elaboration Theory, Reigeluth, 1999b; Reigeluth & Stein, 1983). The basic idea is to use a whole-task approach where the first task class refers to the simplest version of whole tasks that experts encounter in the real world. For increasingly more complex task classes the assumptions that simplify task performance are relaxed. The final task class represents all tasks, including the most complex ones that professionals encounter in the real world.

A simple illustration of this simplifying-assumptions approach[1] might be given for the moderately complex skill, searching for literature (see also Figure 1). The following task factors can be identified that determine how complex it is to perform this skill: (a) the clearness of the concept definitions within or between domains (ranging from clear to unclear); (b) the number of articles that are written about the topic of interest (ranging from small to large); (c) the number of domains in which relevant articles have been published and hence, the number of databases that need to be searched (ranging from one familiar database to many databases that are relevant for the topic of interest); (d) the type of search (ranging from a search on titles and key words to abstracts and full text); and (e) the number of search terms and Boolean operators used (ranging from few search terms to many search terms that are interconnected with Boolean operators). Given these factors, the assumptions for the first, simplest task class can be defined as follows: A category of learning tasks that confronts learners with situations in which the search is performed in a domain in which the concepts are clearly defined, on titles and keywords in one particular database, with only few search terms and yielding a limited number of relevant articles. The most complex task class is defined as a category of learning tasks that confronts learners with situations where concept definitions within or between domains are unclear and in which full-text searches have to be performed in several relevant databases, with many search terms interconnected by Boolean operators in order to limit the otherwise large number of relevant articles. Additional task classes of an intermediate complexity level can be added in between by varying one or more of the task factors.

Once the task classes are defined, the learning tasks can be selected and developed for each class. For instance, one could ask an experienced librarian to come up with concrete cases in which a successful search has been performed on titles in one particular database, with only few search terms and yielding a limited number of highly relevant articles (i.e., cases that fit within the first task class). The same is done for subsequent, more complex task classes. The cases that are selected for each task class form the basis for the to-be-developed learning tasks. For each task class, enough cases are needed to ensure that learners receive enough practice to reach mastery. It should be noted that the cases or learning tasks within the same task class are *not* further ordered from simple to complex; they are considered to be equivalent in terms of difficulty. However, on this microsequencing level a high *variability* of the learning tasks within the same task class is of utmost importance (e.g., Gick & Holyoak, 1983; Paas & van Merriënboer, 1994). They are best sequenced in random order and should differ from each other in terms of the saliency of defining characteristics, the context in which the task has to be performed, the familiarity of the task, or any other task dimensions that also vary in the real world. This high variability is necessary to promote the development of rich cognitive schemata, which allow for schema-based transfer from the training program to the real world.

Learner support. While there is no increasing difficulty for the learning tasks within one task class, they do differ with regard to the amount of *support* provided to learners. Much support is given for learning tasks early in a task class, and no support is given for the final learning task in a task class. This process of diminishing support as learners acquire more expertise is called "scaffolding." It is repeated for each subsequent task class, yielding a saw-tooth pattern of support throughout the whole training program (see the filling of the circles in Figure 2). A general framework of human problem solving (Newell & Simon, 1972) is used to distinguish support structures. According to this framework, four elements are needed to describe learners' work on a learning task: (a) the given state that a learner is confronted with; (b) the criteria for an acceptable goal state; (c) a

1 More advanced approaches to the specification of task classes are described in van Merriënboer, 1997.

solution, that is, a sequence of operators that enables the transition from the given state to the goal state, and (d) a problem-solving process, which may be seen as the tentative application of mental operations in order to reach a solution. This framework is used to make a distinction between product-oriented and process-oriented support. Product-oriented support only relates to the first three elements: The given state, the goal state, and the solution. Process-oriented support also takes the problem-solving process itself into account.

Product-oriented support is provided in a lesser or higher degree by different types of learning tasks. Highest product-oriented support is provided by a *case study* or worked-out example, which confronts the learner with a given state, a desired goal state, and a solution, intermediate solutions, or both. In order to arouse learner interest, it may be desirable to use case studies that describe a spectacular event, such as an accident, a success story, or a disputed decision that turned out all right. Typically, learners have to answer questions that provoke deep processing and the induction of mental models from the given example materials. By studying examples of intermediate solutions, learners get a clear impression of how a particular domain is organized. At the other extreme, no support is provided by a *conventional* learning task, which provides only a given state and a desired goal state. Learners have to come up with a solution themselves. Table 1 presents some illustrations of other types of learning tasks, roughly ordered from high to low product-oriented support (see van Merriënboer, 1997, for a complete description of types of learning tasks).

Process-oriented support is also directed toward the problem-solving process itself. Highest process-oriented support is provided by a *modeling example,* which confronts the learner with an expert who is performing the task and simultaneously explaining why the task is performed as it is performed. It is essential to present an appropriate role model that has credibility as well as expertise the observer can comprehend. Thinking aloud may be very helpful to bring the hidden mental problem-solving processes of the expert into the open. As for case studies, learners often have to answer questions that provoke deep processing and the induction of cognitive strategies from the given modeling example. By studying the modeling example, they get a clear impression of the systematic approaches and rules of thumb that professionals use.

Process-oriented support may also be provided in the form of performance constraints and performance support structures. Both are based on a cognitive task analysis of strategic knowledge, which yields a description of cognitive strategies as systematic approaches to problem solving (SAPs) that experts use to solve problems in the domain of interest. An SAP distinguishes the successive phases in a problem-solving process and the rules of thumb or heuristics that may be helpful to successfully complete each of the phases. *Performance constraints* typically require learners to complete one phase satisfactorily before they may enter the next phase. *Performance support structures* are less directive and typically take the form of problem-solving support. For instance, in order to guide learners through the problem-solving process, process worksheets that list the main phases and useful rules of thumb for each of the phases may be provided to them. Or as a more advanced approach, computer-based learning tools may invite learners to approach the problem at hand as an expert would do (for an example, see Dufresne, Gerace, Thibodeau-Hardiman, & Mestre, 1992).

Component 2: Supportive Information

Obviously, learners need information in order to work fruitfully on nonrecurrent aspects of learning tasks and to genuinely learn from those tasks. This supportive information provides the bridge between what learners already know and their work on the learning tasks. It is the information that teachers typically call “the theory” and which is often presented in study books and lectures. Because the same body of general knowledge underlies all learning tasks in the same task class, and because it is not known beforehand which knowledge is precisely needed to successfully perform a particular

Table 1 ☐ Examples of different types of learning tasks for the complex skill, searching for relevant research literature. The learning tasks fit the first task class (see text) and are ordered from high product-oriented support (case studies) to no support (conventional learning tasks).

Task Class: *Learners are confronted with situations where concepts in the to-be-searched domain are clearly defined. Only a small number of articles is written about the subject and articles are only written in one field of research. Therefore, the search only needs to be performed on titles of articles in one database from the particular field of research. Only a few search terms are needed to perform the search and the search will yield a limited number of articles.*

Learning Task	*Given(s)*	*Goal(s)*	*Solution*	*Task Description*
Case study	+	+	+	Learners receive a research question, a list with articles, and a search query used to produce the list of articles. They must evaluate the quality of the search query and the list of articles.
Reverse	Predict	+	+	Learners receive a list with articles and a search query used to produce the list of articles. They must predict possible research questions for which the list of articles and search query are relevant.
Imitation	+Analog +	+Analog +	+Analog Find	Learners have a worked-out example available of a research question, a list with articles, and a search query used to produce the list of articles. They receive another research question and the goal to produce a list with a limited amount of relevant articles. By imitating the given example materials, they must formulate the search query, perform the search and make a selection of articles for the new research question.
A-specific goal	+	Define	Find	Learners receive a research question and a highly a-specific goal, for instance to come up with as many search queries as possible that might be relevant to the research question. They must formulate those search queries.
Completion	+	+	Complete	Learners receive a research question, the goal to produce a list with a limited number of relevant articles, and an incomplete search query. They must complete the search query, perform the search and make a selection of articles.
Conventional	+	+	Find	Learners receive a research question and the goal to produce a list with a limited number of relevant articles. They must formulate the search query, perform the search and make a selection of articles.

\+ indicates that this element is provided to the learners.

learning task, supportive information is not coupled to individual learning tasks but to task classes (see the supportive information in Figure 2). The supportive information for each subsequent task class is an addition to or an elaboration of the previous information, allowing learners to do things that could not be done before. Instructional methods for the presentation of supportive information primarily promote schema construction through elaboration, that is, helping students to establish nonarbitrary relationships between newly presented information elements and their prior knowledge. This process of elaboration yields highly complex schemata that should allow for deep understanding.

As discussed in the previous section, the cognitive schemata that may help learners to perform nonrecurrent aspects of a complex task come in two forms. (a) Mental models allow one to reason within the learning domain, and (b) cognitive strategies allow one to systematically approach problems in this domain and use rules of thumb or heuristics that guide the problem-solving process. Supportive information reflects both types of schematic knowledge. For instance, it is known that Tiger Woods makes extensive study of the layout of golf courses around the world (to develop mental models of how the world is organized) and of videotapes of his competitors (to develop cognitive strategies of how to approach problems in this world). Thus even expert task performers further develop their mental models and cognitive strategies in order to improve their performance. The same is true for learners in a training program aimed at searching for relevant research literature. In addition to working on learning tasks as specified in Table 1, learners may, for instance, study how databases are organized in order to develop useful mental models, and they may study how expert librarians develop search queries in order to develop more effective strategies themselves.

Mental models. Mental models are declarative representations of how the world is organized and may contain both general, abstract knowledge and concrete cases that exemplify this knowledge. So, strong models allow for both abstract and case-based reasoning. Mental models may be viewed from different perspectives and can be analyzed as conceptual models, structural models, or causal models. First, conceptual models (*what is this?*) focus on how "things" are interrelated and allow for the classification or description of objects, events or activities. For instance, knowledge about several types of market stocks, and how these differ from each other, helps financial analysts to determine the risk associated with particular portfolios. Second, structural models (*how is this organized?*) describe how plans for reaching particular goals are related to each other. Plans can be distinguished in scripts (what happens when?) that focus on how events are related in time and help to understand and predict behavior, and building blocks or templates (how is this built?) that focus on how objects are related in space and help to understand or design artifacts. For instance in the biology domain, knowledge about routine sequences of events (i.e., scripts) occurring in a particular species of birds enables a biologist to predict and understand the ritual of mating behavior. In the computer-programming domain, knowledge about stereotyped patterns of programming code (i.e., programming templates) and how these patterns fit together helps computer programmers to understand and develop programs. Third, causal models (*how does this work?*) focus on how principles affect each other and help to interpret processes, give explanations for events, and make predictions. For instance, knowledge about how components of a chemical factory function, and how each component affects other components, helps process operators to diagnose malfunctions. Mental models may also combine these three different perspectives and thus allow for qualitative reasoning in a particular domain.

Central to mental models is the existence of many nonarbitrary relationships between knowledge elements. For the presentation of supportive information, it is of utmost importance to stress those nonarbitrary relationships. Table 2 lists some popular instructional methods that help learners to identify relevant relationships. These methods can be used in an expository fashion or in an inquiry fashion. *Expository methods* explicitly present the nonarbitrary relationships to learners. For instance, when learners study a particular piece of machinery one could explicitly indicate to them what the different parts of the machine are (see Method 1 in Table 2). *Inquiry methods,* on the other hand, ask the learners to "discover" the relationships. Thus, in the previous example one should ask the learners to identify the different parts of the machine. Inquiry approaches are time-consuming, but because they directly build on learners' prior knowledge they are very appropriate for interconnecting new information and already existing cognitive schemata. It is a form of *guided* discovery, because the leading questions (e.g., which parts can be distinguished in this machine?) help learners to identify relevant nonarbitrary relationships.

A particularly important relationship is the experiential one, which relates general, abstract knowledge to concrete cases (see Method 2 in Table 2). The 4C/ID-model distinguishes the

Table 2 ☐ Ten popular instructional methods for the presentation of supportive information, stressing nonarbitrary relationships. Adding the prefix "Ask the learners to . . ." indicates an inquiry use of the method.

Instructional method	*Highlighted relationship(s)*
1. (Ask the learners to . . .) analyze a particular idea into smaller ideas	Subordinate kind of or part of relation
2. (Ask the learners to . . .) present a well-known, familiar example or counterexample for a particular idea	Subordinate experiential relation
3. (Ask the learners to . . .) present a more general idea or organizing framework for a set of similar ideas	Superordinate kind of or part of relation
4. (Ask the learners to . . .) compare and contrast a set of similar ideas	Coordinate kind of or part of relation
5. (Ask the learners to . . .) provide a description of a particular idea in its main features or characteristics	Subordinate kind of or part of relation
6. (Ask the learners to . . .) find an analogy for a particular idea	Coordinate similarity relation
7. (Ask the learners to . . .) explain the relative location of elements in space or time	Location relation
8. (Ask the learners to . . .) rearrange elements and predict effects	Location relation
9. (Ask the learners to . . .) explain a particular state of affairs	cause-effect or natural process relation
10. (Ask the learners to . . .) make a prediction of future states	cause-effect or natural process relation

presentation of general information (i.e., a didactical specification of conceptual, structural and causal models) and concrete cases or case studies that illustrate this information. If a conceptual perspective is taken, case studies may describe concrete objects, events, or situations. For models with a structural perspective, case studies may be artifacts designed in order to reach particular goals. And for models with a causal perspective, case studies may illustrate real-life processes. Computer-based simulations provide a powerful approach to the presentation of case studies, because learners are then able to change the settings of particular variables and study the effects of those changes on other variables, that is, explore relationships (de Jong & van Joolingen, 1998). The goal of such "microworlds" is not primarily to practice the complex target skill, but to help learners construct mental models of how the world is organized through active experimentation.

The 4C / ID-model furthermore distinguishes inductive and deductive strategies for presenting supportive information. In an inductive strategy one or more case studies are presented as part of the supportive information; then the general, abstract information is dealt with; and finally the learning tasks are given. In a first type of inductive strategy, the *inductive-inquiry* strategy, one presents one or more case studies and then asks the learners to identify the relationships between pieces of information illustrated in the case. As described above, such a form of guided discovery is time consuming and should be used only if there is enough instructional time available, learners have no experience with the skill whatsoever, and a deep level of understanding is required. A second type of inductive strategy is the *inductive-expository* strategy: One starts with one or more case studies and then explicitly presents the relationships between pieces of information that were illustrated in the cases. The 4C / ID-model suggests using this approach by default, because it is reasonable and time effective, and starting with concrete, recognizable case studies works well for learners with little prior knowledge. The third alternative is a *deductive* strategy, where learners work from the general, abstract information directly toward learning tasks that fulfill the role of case studies. Here, one starts with explicitly presenting relationships between pieces of information (the theory) and then illustrates this general information with one or more learning tasks with maximum product-oriented support (note that many instructors use this method by default!). A problem is that learners without prior knowledge may have severe difficulties with understanding the general information. It should thus be used only if instructional time is limited, learners have already some experience with the skill, and a deep level of understanding is not strictly necessary.

Cognitive strategies. Like mental models, cognitive strategies contain both general, abstract knowledge and concrete cases that exemplify this knowledge. As mentioned before, cognitive strategies may be analyzed as SAPs, describing the successive phases in a problem-solving process and the rules of thumb or heuristics that may be helpful to successfully complete each of the phases. Instructional methods for presenting cognitive strategies closely resemble methods for presenting mental models, and in particular, structural and causal models. For instance, one might ask the learners to explain why one phase should precede another phase (see Method 7 in Table 2), predict the effects of rearranging phases (Method 8), explain how the use of particular rules of thumb brought a particular state of affairs about (Method 9), or predict the effects of the use of particular rules of thumb (Method 10). For cognitive strategies, the experiential relationship refers to concrete cases that take the form of modeling examples, which illustrate how the application of SAPs can help to reach a solution. The modeling examples provide a hinge between the supportive information (where they illustrate cognitive strategies) and the learning tasks (where they may be seen as learning tasks with maximum process-oriented support). As argued before, modeling examples may refer to an expert who is performing a nontrivial task and simultaneously explaining why particular decisions and actions are taken (e.g., by thinking aloud). Preferably, the example is interspersed with questions that require the learners to think critically about the problem-solving process that is modeled. Because of the highly abstract character of cognitive strategies, the 4C/ID-model prescribes only an *inductive-expository* strategy for their presentation. Thus, as a rule one, should start with the presentation of one or more modeling examples and then explicitly present the problem-solving phases and rules of thumb that are illustrated by those examples.

Cognitive feedback. A final part of supportive information relates to feedback that is provided on the quality of performance. This so-called cognitive feedback (Balzer, Doherty, & O'Connor, 1989; Butler & Winne, 1995) refers to the nonrecurrent aspects of performance only and should thus promote schema construction. Because nonrecurrent performance is never "correct" or "incorrect," but only more or less effective, cognitive feedback is only provided after learners have finished one or more learning tasks, or even after they have finished a whole task class. Well-designed feedback should stimulate learners to reflect on the quality of their personal problem-solving processes and found solutions, so that more effective mental models and cognitive strategies can be developed. This central role of reflection is shared with a cognitive apprenticeship model (Collins, Brown & Newman, 1989; see also Kluger & DiNisi, 1998). Debriefing sessions, peer or expert critiques, and group discussions offer a valuable approach. Then, learners' actual problem-solving processes may be compared and contrasted with presented SAPs, modeling examples that illustrated those SAPs, or problem-solving processes reported by other learners. Further, found solutions may be compared and contrasted with presented general information, case studies that illustrated this general information, or solutions found for previous problems or reported by other learners. In such discussions, inquiry methods as presented in Table 2 may well be used as a form of "feedback by discovery."

Component 3: Just-in-Time Information

Whereas supportive information pertains to the nonrecurrent aspects of a complex skill, JIT information pertains to the recurrent aspects, that is, constituent skills that should be performed after the training in a highly similar way over different problem situations. JIT information provides learners with the step-by-step knowledge they need to know in order to perform the recurrent skills. They can be in the form of, for example, directions teachers or tutors typically give to their learners during practice, acting as an "assistant looking over your shoulder." Because the JIT information is identical for many learning tasks, which all require the same recurrent constituent skills, it is typically provided during the first learning task for which the skill is relevant (see JIT information in Figure 2). For subsequent learning tasks, JIT information is quickly faded away as learners gain more expertise (a principle called *fading*). Instructional methods for the presentation of JIT information primarily promote compilation through restricted encoding of situation-specific knowledge into cognitive rules. The JIT information is specified at the entry level of the learners, that is, at a level that is suitable to present to the lowest-level ability learner. It is not critical that the information be embedded in existing schemata in declarative memory. Because of this, during presentation, no particular reference has to be made to related knowledge structures in long-term memory.

Rules that enable learners to correctly perform the recurrent aspects of a complex skill are formed through practice, and this process is facilitated when the information that is necessary for forming the rules is directly available in working memory, precisely when learners need it. This concerns information that describes the rules themselves (or procedures that combine those rules) as well as information describing the knowledge elements (i.e., facts, concepts, plans or principles—the same knowledge elements that make up complex schemata) that are prerequisite to learning and to performing those rules. For instance, when you are learning to play golf, your instructor will preferably tell you how to hold your club, how to take your stance, and how to make your swing out on the driving range while you are making your first drives, and not during a theory lesson in a classroom. The same is true for learners in a training program aimed at searching for relevant research literature. For the recurrent aspects of this complex skill, such as operating the search program, procedural directions for operating the program are also best presented during practice, precisely when learners need it. The next section discusses the design of such information displays.

Information displays. JIT information is organized in small units, called information displays. Organization in small units is considered to be essential because only the presentation of relatively small amounts of new information at the same time can prevent processing overload during practice. Information displays include a didactical specification of the rules that describe correct performance as well as the knowledge that is prerequisite to a correct application of those rules. For instance, a rule may state that "in order to start the machine, you must first switch it on" and also indicate that the on-off switch is located on the back of the machine (i.e., a fact that is prerequisite to a correct application of the rule). Or in the context of searching relevant research literature, a rule for operating the search program may state that "in order to search on keywords, select the choice FIELDS in the menu SEARCH and enter the desired search terms in the field that is labeled KW." The same information display may give a definition of the concept FIELD (i.e., a concept that helps to understand the given rule). These examples make clear that information displays may best be characterized as how-to instruction or rule-based instruction (Fisk & Gallini, 1989).

A traditional approach to JIT information presentation lets learners memorize the information before they start to work on learning tasks, so that it may be activated in working memory when needed. This approach is *not* recommended for the simple reason that memorization is a dull activity and has no advantages to more active JIT approaches. As a regular approach, one should directly present information displays when the learners need this information to work on the recurrent aspects of a particular learning task. It is thus connected to the first learning task for which it is relevant, and for subsequent learning tasks it fades away. However, a requirement of this approach is that the designer have some control over the learning tasks that confront the learner;

otherwise it is not feasible to connect the information displays to the learning tasks. If training takes place on the job, the designer often lacks this control. Learning aids such as on-line help systems, checklists, and manuals then provide a good alternative. While the JIT information is not directly presented when it is needed for the learning tasks, it is at least easily available and readily accessible. In the field of minimalism (Carroll, Smith-Kerker, Ford, & Mazur-Rimetz, 1988) guidelines for the design of "minimal manuals" are in full agreement with this approach (e.g., van der Meij & Carroll, 1995).

Demonstrations and instances. Most of the elements in information displays are general statements about the recurrent skill, or, *generalities* (Merrill, 1983, 1999). For instance, rules are general in that they can be applied in a variety of situations, and prerequisite concepts are general in that they refer to a category of objects or events. It is often desirable to present examples that illustrate or exemplify those generalities. For rules, such examples are called *demonstrations;* for concepts, plans, and principles, they are called *instances.* The 4C/ID-model suggests providing demonstrations and instances in the context of the learning tasks. This should allow learners to place the recurrent skill in the context of the whole task. Thus, demonstrations of the recurrent aspects of a complex skill ideally coincide with suitable learning tasks such as modeling examples, and instances of prerequisite knowledge elements ideally coincide with suitable learning tasks such as case studies. This is a deductive-expository approach, where the generalities (i.e., the information displays) are presented *simultaneously* with the examples (i.e., demonstrations and instances), which are part of the same learning task as the information display is connected to.

Two examples illustrate this principle. Suppose that a complex skill in the process control domain requires the execution of a standard procedure (i.e., a recurrent constituent skill) to detect possible out-of-bound situations. Connected to the first learning task for which the recurrent skill is relevant, an information display will provide a general description of the procedure as well as its prerequisite knowledge. Then, the procedure is best demonstrated as part of a modeling example, where the learner's attention is focused on the recurrent aspects that the demonstration is about. Another example is in the domain of computer programming. Suppose that completion tasks are used asking the learners to complete increasingly larger parts of partial, well-structured computer programs. When a particular programming plan (i.e., a stereotyped pattern of programming code, such as an assignment plan, a looping plan, etc.) is used for the first time in the given part of a to-be-completed program, an information display may be presented with a rule describing when and how to use this plan, as well as a general description of the plan that is prerequisite to the application of the rule. Note that at the same time, a concrete instance of this programming plan is presented in the given part of the to-be-completed computer program. In Completion ASsignment COnstructor (CASCO), an intelligent tutoring system for teaching introductory programming (e.g., van Merriënboer & Luursema, 1996), the programming code that exemplifies the information display (i.e., the instance) is highlighted in the to-be-completed program.

Corrective feedback. A final part of JIT information relates to feedback that is provided on the recurrent aspects of performance. Like all JIT information, this feedback should promote compilation. If rules that algorithmically describe effective performance are not correctly applied, the learner is said to make an "error." Corrective feedback on such errors is preferably presented *immediately* after misapplication of a particular rule. It is necessary for the learner to preserve information about the conditions for applying a particular rule in working memory until feedback (right-wrong) is obtained. Only then can a rule be compiled that attaches the correct action to its critical conditions. Obviously, any delay of feedback may hamper this process.

The 4C/ID-model does not propagate the idea of errorless learning. On the one hand, it is practically impossible to prevent errors when learners are working on rich learning tasks. But even more important, for many recurrent aspects of a complex skill it is considered important that learners learn to recognize their own errors and how to recover from them. Well-designed feedback should then inform the learner *why* there was an error and provide a suggestion or *hint* of how to reach the goal. Such a hint will often take the form of an example or demonstration. It is important not to simply give the correct action because this does not allow for practice, which is critical for compilation to occur. Furthermore, it may be necessary to indicate to the learner how to recover from the results of the error that has been made.

Component 4: Part-task Practice

Learning tasks are designed in such a way that they primarily promote schema construction, but they also facilitate compilation for recurrent aspects of the complex skill. This process is driven by the repeated practice of recurrent constituent skills in the learning tasks. Often, learning tasks provide enough opportunity to practice both the nonrecurrent and the recurrent aspects of the complex skill. This is possible because one can take care of the different nature of underlying learning processes for recurrent and nonrecurrent constituent skills in the context of information presentation. JIT information presentation aims at restricted encoding of newly presented information in rules; supportive information presentation aims at elaboration of existing schemata with new information. However, if a very high level of automaticity of particular recurrent aspects is required, the learning tasks may provide insufficient repetition to provide the necessary amount of strengthening. Only then, it is necessary to include additional part-task practice for those selected recurrent aspects in the training program (see part-task practice in Figure 2). But in general, an overreliance on part-task practice is not helpful to complex learning.

Part-task practice promotes the compilation of procedures or rules and especially their subsequent strengthening, which is a very slow process that requires extensive amounts of practice items. Well-known examples of part-task practice are drilling children on multiplication tables and playing scales on musical instruments. In training design, part-task practice is typically applied for recurrent constituent skills that are critical in terms of safety, for instance, detecting dangerous air traffic situations from a radar screen in the context of air traffic control. But if available instructional time allows, it may also be used for recurrent constituent skills with relations in the skills hierarchy indicating that they (a) enable the performance of many other skills higher in the hierarchy (which is a central idea of Gagné's learning hierarchy, Gagné et al., 1992), or (b) have to be performed simultaneously with many other coordinate skills. It is critical to start part-task practice within an appropriate cognitive context because it has been found to be effective only *after* exposure to a simple version of the whole complex skill (Carlson et al., 1990; Schneider & Detweiler, 1988). One should thus identify the first task class for which performance of the recurrent aspect is required, and initiate part-task practice during this task class—preferably after case studies or other learning tasks with ample learner support have already been worked on. This allows learners to identify the activities that are required to integrate the recurrent aspect in the learning tasks. The following sections briefly discuss the design of practice items for part-task practice and methods for overtraining.

Practice items. Compared to the specification of learning tasks, the specification of practice items for part-task practice is a pretty straightforward process. For learning tasks, simple-to-complex task classes first guide the process of selecting concrete cases, which are then transformed into meaningful learning tasks that require the learners to perform several constituent skills in a coordinated fashion. For part-task practice, however, there is only *one* relevant recurrent constituent skill or objective whose effective performance can be algorithmically described in terms of rules. Practice items should then invite learners to repeatedly perform the recurrent constituent skill. The saying, practice makes perfect, is actually true for part-task practice. It is important that the whole set of practice items be divergent, meaning that it be representative for all situations that can be treated by the rules. This is necessary to develop a broad set of situation-specific rules that may subsequently yield optimal rule-based transfer to new problem situations.

Only for highly complex algorithms, represented by large rule sets, it may be necessary to work from simple to complex practice items. The whole algorithm is then decomposed into parts, and learners are extensively trained on each part separately before they begin to practice the whole recurrent skill. This form of sequencing is fundamentally different from sequencing learning tasks. In order to facilitate schema construction, a whole-task approach is used to sequence simple-to-complex task classes; the learning tasks within the same task class exhibit a high variability, and each learning task requires the integration and coordination of constituent skills involved. In contrast, breaking the task down in parts that are separately trained and then gradually combined toward the whole task (i.e., a part-whole approach) yields a lower variability and facilitates a rapid automation of rules.

With regard to learner support, there is also a striking difference between support for learning tasks and support for practice items in part-task practice. The performance of recurrent aspects *cannot* be described as the tentative application of mental operations in order to find a solution (i.e., problem solving). Applying the rules simply *is* the solution and ensures that the desired goal state is reached. It is thus the application of rules that is of importance, rather than the search for a solution. So, performance support for part-task practice takes the form of *procedure support*. Special practice items may be relevant if algorithms leave learners error prone, or if different algorithms may easily be mixed up. For instance, a well-known strategy for ordering practice items is the recognize-edit-produce sequence (REP, Gropper, 1983), which starts with items that require learners to recognize which rules to apply, continues with items for which learners have to edit incorrect applications of the rules, and ends with conventional items for which learners have to apply the rules in order to produce the solution. Performance constraints for part-task practice may take the form of "training wheel interfaces" (Carroll et al., 1988) which resemble the use of training wheels on children's bikes. If particular rules leave learners error prone, one may make the actions related to those rules "unreachable" for the learners early in the training process. Such a training-wheels approach may also be used to support the learning of recurrent aspects during whole-task practice on learning tasks (e.g., see Leutner, 2000).

JIT information for part-task practice. JIT information is obviously not only relevant for learning tasks, where it pertains to the recurrent aspects of the task, but also to part-task practice, where it relates to the single recurrent skill that is practiced. Compared to JIT information presentation for learning tasks, one may now carry forward the principle of presenting JIT even further and provide the information that is relevant for applying a particular rule and its prerequisite knowledge precisely at the moment that this one rule has to be applied by the learner. This is known as single-step or step-by-step instruction (Landa, 1983). Furthermore, demonstrations of rule application and instances of prerequisite knowledge cannot be provided as part of a learning task (i.e., in their whole-task context), but are provided separately and simultaneously to the information displays. For instance, if part-task practice is provided for training special emergency procedures, information displays provide a step-by-step description of the procedure and may define prerequisite concepts such as "alarm limit" or "emergency setting." A demonstration should clearly indicate the desired outcome of the procedure, the materials and other pieces of equipment that will be manipulated (i.e., provide concrete instances of the prerequisite knowledge), and should show the actual execution of the procedure using these materials. Feedback on the quality of performance should be provided during practice, ideally, immediately after performing a particular step in a procedure or applying a particular rule.

Overtraining. Part-task practice as discussed above will lead to accurate performance of a recurrent skill. However, extensive amounts of overtraining may be necessary to make the skill fully automatic. Then, the main underlying learning process is no longer compilation but strengthening. For skills that need to be performed highly automatically, the ultimate goal is not always highest accuracy. More often, the goal is to obtain acceptable accuracy, combined with high speed and the ability to perform the skill together with other skills, and ultimately, in the context of the whole task. In order to reach this, the recurrent skill (which is already being performed to the required level of accuracy) is first practiced under speed stress. After speed criteria have been reached, the skill is practiced under time-sharing conditions simultaneously with other effort-demanding skills. And finally, the skill is practiced in the context of the whole task. So, performance criteria gradually change from (a) accuracy, to (b) accuracy combined with speed, to (c) accuracy combined with speed under time-sharing conditions or high overall workload (Salisbury, Richards, & Klein, 1985).

Relatively short, spaced periods of part-task practice or overtraining (i.e., distributed practice) yield better results than long, concentrated periods of part-task practice (i.e., massed practice). Therefore, part-task practice is best intertwined with the learning tasks because this provides distributed practice and also enables the learners to relate the recurrent constituent skill to the whole complex skill. The same principle of *intermix training* is applied if part-task practice is provided for more than one recurrent constituent skill. Practice on those skills is also intertwined in order to distribute practice and to facilitate the perception of interrelationships between constituent skills (cf., Schneider, 1985).

This section ends the description of the four components, which have been graphically interconnected to each other in Figure 2. Table 3 gives a simplified blueprint of a training program for searching for relevant research literature in a textual format. This blueprint also illustrates the four components. First, three task classes are described, each containing several types of *learning tasks*. The task classes show an increase in complexity while the learning tasks within each task class show a decrease in learner support. Second, *supportive information* is specified for each task class. In the first task class, an inductive-expository strategy is illustrated: Learners first receive a modeling example before they start to study supportive information that was illustrated in the modeling example. In the second task class, an inductive-inquiry approach is illustrated: Learners work on a case study and have to identify and discover the relationships between different plans (i.e., templates for search queries using Boolean operators) that are illustrated in the case. In the third task class, a deductive approach is illustrated: Supportive information is made available before learners start to work on the first learning task. At the end of the second and third task class, learners receive cognitive feedback on their work on a conventional learning task. The third component, *JIT information*, is specified for each learning task where it is relevant. And fourth and last, it is specified where to initiate additional *part-task practice*. In this blueprint, it is assumed that learners receive some additional part-task practice in the use of Boolean operators, starting parallel to the first learning task that illustrates their use.

DISCUSSION

In this article, we presented a description of the four blueprint components that are the basic building blocks for training programs for complex learning designed according to the 4C/ID-model, and their major theoretical foundations in cognitive psychology. The four blueprint components refer to (a) learning tasks; (b) supportive information; (c) JIT information, and (d) part-task practice. These components and their associated instructional methods were described in detail and an example of a training blueprint was given for the moderately complex skill searching for relevant research literature (see Table 3).

The 4C/ID-model should be used to develop training programs for complex skills and when transfer is the overarching learning outcome. Such training programs have a typical length of weeks, months or even years. The model is not developed for teaching conceptual knowledge or procedural skills per se. It also is not very useful for designing very short programs that only take an instructional time of hours or a few days (e.g., traditional lesson design or design of short workshops). If the model is used, a blueprint as presented in Table 3 may not always provide enough detail to start the actual development of

Table 3 ☐ Simplified example of a training blueprint for the moderately complex skill, searching for relevant research literature.

Task Class 1: Learners are confronted with situations where the concepts in the to-be-searched domain are clearly defined. Only a small number of articles is written about the subject and articles are only written in one field of research. Therefore, the search needs only to be performed on titles of articles in one database from the particular field of research. There are only a few search terms needed to perform the search and the search will yield a limited number of articles.	
Supportive Information: Modeling example Learners watch an expert who performs a literature search and explains his or her actions while doing so.	
Supportive Information: Presentation of cognitive strategies • Systematic approach to problem solving (SAP) of the four phases involved in performing a literature search: (a) selecting an appropriate database, (b) formulating a search query, (c) performing the search, and (d) selecting results. • SAPs for quickly scanning the relevance of scientific articles. *Supportive Information: Presentation of mental models* • Conceptual model of literature search concepts • Structural model of how databases are organized and can be used. • Conceptual model of different types of scientific articles and how they are organized.	
Learning Task 1.1: Case study Learners receive three worked-out (good) examples of literature searches. Each example describes a different research questions in the same subject matter domain, the search query and the produced list of articles. The learners have to study the examples and explain why the different search queries produced the desired results.	
Learning Task 1.2: Completion Learners receive a research question and an incomplete search query that produces a list containing irrelevant items. They must refine the search query using additional search terms, perform the search and select the relevant articles.	*JIT* information presentation* • Procedures for operating the search program • Procedures for using a thesaurus
Learning Task 1.3: Conventional Learners receive a research question. They have to perform a literature search for the 10 most relevant articles.	*JIT information presentation* • Procedures for operating the search program (fading) • Procedures for using a thesaurus (fading)
Task Class 2: Learners are confronted with situations where the concepts in the to-be-searched domain are clearly defined. A large number of articles is written about the subject, but only in one field of research. Therefore, the search needs only to be performed on titles of articles in one database from the particular field of research. However, many search terms need to be interconnected with Boolean operators to limit the otherwise large number of articles the search can yield.	
Supportive Information: Case study Learners receive three worked-out (good) examples of literature searches. Each example contains an elaborate search query in which Boolean operators are used.	

*JIT = Just-in-time

Table continues

Table 3 ☐ *continued*

Supportive Information: Inquiry for mental models

- Learners are asked to identify templates of search queries describing Boolean combinations of search terms that can be used to make search queries more specific.

Learning task	JIT information presentation	Part-task Practice
Learning Task 2.1: Imitation + constraint Learners have a worked-out example of a research question available, a list of articles and an elaborate Boolean search query to produce the list of articles. They receive a similar research question, and a goal to produce a list with a limited number of relevant articles. By imitating the given example they must formulate the search query, perform the search and select relevant articles. They can only perform the search after the search query is approved.	*JIT information presentation* • Syntax for specifying Boolean search queries	*Part-task Practice* • Applying Boolean operators (continue as needed)
Learning Task 2.2: Completion Learners receive a research question and a list of search terms. They have to formulate a search query by combining the given search terms using Boolean operators.	*JIT information presentation* • Syntax for specifying Boolean search queries (fading)	
Learning Task 2.3: Conventional Learners receive a research question. They have to perform a literature search for the 10 most relevant articles.	*JIT information presentation* • Syntax for specifying Boolean search queries (fading)	

Supportive Information: Cognitive feedback

Learners receive feedback on their approach to solve the problem in Learning Task 2.3.

Task Class 3: Learners are confronted with situations where the concepts in the to-be-searched domain are not clearly defined. Identical terms are used for different concepts, and identical concepts are described with different terms. A large number of articles is written about the subject and articles are written in several fields of research. Therefore, next to searching on titles of articles, the search also needs to be performed on abstracts and texts. Also, databases from different fields of research have to be searched. Many search terms need to be interconnected with Boolean operators to make sure that all relevant articles (using different terminology) are found and that irrelevant articles (using the same terminology as relevant ones) are excluded.

Supportive Information: Presentation of cognitive strategies

- SAP for determining the number of databases to search and whether to also search on abstracts and full texts.

Supportive Information: Presentation of mental models

- Structural model of templates of search queries describing Boolean combinations of search terms that can be used to search for articles about ill-defined subjects.
- Conceptual model of different types of databases for different fields of study, describing structure, special search requirements, etc.

Learning task	JIT information presentation	Part-task Practice
Learning Task 3.1: Completion + Reverse Learners receive a research question and an elaborate search query. They have to predict which databases should be used and then perform the query. They then have to refine the query and select relevant articles.	*JIT information presentation* • Procedures for searching specific databases	*Part-task Practice* • Applying Boolean operators (continue as needed)
Learning Task 3.2: Conventional Learners receive a research question. They have to perform a literature search for the 10 most relevant articles.	*JIT information presentation* • Procedures for searching specific databases (fading)	

Supportive Information: Cognitive feedback

Learners receive feedback on their approach to solve the problem in Learning Task 3.2.

instructional materials. Especially for the development of computer-based or self-instructional materials, more concrete and specific descriptions may be needed of the supportive and JIT information that should be made available to the learners, as well as the practice items that make up part-task practice and the support structures that are available for those practice items and the learning tasks. According to the 4C/ID-model, a process of cognitive task analysis is then needed to flesh out the blueprint (van Merriënboer, 1997).

With respect to the design of individualized or adaptive training programs it should be noted that it is not always desirable to specify all aspects of the blueprint beforehand. In our literature example (Table 3), the order for presenting learning tasks, practice items, and different types of information is fixed and identical for all learners. However, to be able to adapt instruction to differences in learner progress, it is required that sequencing and timing for the presentation of information and practice opportunities can be dynamically adjusted. The 4C/ID-model allows for this. For example, instead of designing a fixed sequence of learning tasks with learner support diminishing at a fixed rate, one could design sets of learning tasks where each set contains several versions of the same learning task but with different amounts of learner support. During training either a human tutor or a computer-based system can select and present learning tasks with an optimal amount of learner support, based on learner performance on previous learning tasks. Only when the required level of performance for a particular task class has been reached does the learner continue to the next task class. In more advanced training environments this dynamic approach can be taken one step further. In CASCO, (van Merriënboer & Luursema, 1996), fuzzy-logic algorithms are used to model learner skill development and to dynamically generate new learning tasks that best suit individual learner needs.

With regard to instructional methods, the 4C/ID-model typically applies a mix of constructivist and instructivist approaches. The basis for the design of a training program is whole-task practice, offering nontrivial, realistic and increasingly more authentic task classes and learning tasks to the learners. Schema construction by induction and mindful abstraction from concrete cases are assumed to be key learning processes, reflecting a strong constructivist approach. With regard to the presentation of information, an inductive-inquiry strategy or (guided) discovery approach also reflects a constructivist viewpoint. This strategy is prescribed for the presentation of supportive information for which deep understanding is required—provided that the available training time allows for it. But for the reason of instructional efficiency, the 4C/ID-model also has some clear instructivist features. For the presentation of JIT information, a deductive-expository strategy is recommended. For the presentation of supportive information, an inductive-expository strategy is recommended by default, and a deductive-expository strategy is recommended if the available time for instruction is limited and learners already have some relevant experience. Thus, in order to make the training process more efficient, it is sometimes necessary to provide learners with prespecified, general knowledge that may be helpful and offer guidance to solve the problems in a particular domain.

It is important to realize that blueprints developed according to the 4C/ID-model mark the transition from the design phase to the production or development phase. The model does not provide detailed guidelines for this development phase. Important elements such as overviews of content structure, summaries, transitions and so forth are not dealt with. The main reason for this is that the guidelines for the development of learning environments and the production of instructional materials are often media specific. After a blueprint for a training program has been finished, a final decision should be made as to the primary and secondary media that will be used. This selection of media is influenced by many factors not discussed in the 4C/ID-model, such as constraints to implement the design (time and money), special task requirements, and characteristics of the target group. After the final media selection, specialized instructional design models that provide media-specific guidelines for materials development should be consulted.

While the 4C/ID-model does not include guidelines for *final* media selection, it does limit the available media options for each of the four components (see van Merriënboer, 1997). This is because each of the four components corresponds to another category of learning processes, and particular learning processes are best supported by particular media. According to the 4C/ID-model, the primary medium is always related to performing the learning tasks and will thus typically involve a real or simulated task environment. Consequently, most instructional systems based on the 4C/ID-model can be characterized as problem-based, simulation-based, simulator-based, case-based, or scenario-based learning environments. The secondary media are related to supportive information (suitable media include books, hypertext systems, lectures, etc.); JIT information (including on-line help systems, job-aids, pop-up menus, balloon help, etc.), and part-task practice (including drill-and-practice computer programs, part-task trainers, etc.).

With regard to the effectiveness of developed training programs, the most important claim is that the 4C/ID-model helps to develop training programs that lead to higher transfer performance than conventional instruction, and that this effect increases as transfer tasks differ more from the original training tasks. This prediction has been tested in several domains by comparing performance after training (and especially, transfer performance) of training strategies developed according to the 4C/ID-model with conventional real-world strategies or strategies developed according to other models. For example, in a range of studies in the computer programming domain, including classroom studies (van Merriënboer, 1990a, 1990b) and computer-based training studies (Schuurman, 1999; van Merriënboer & de Croock, 1992; van Merriënboer, Schuurman, de Croock, & Paas, 2002), 4C/ID strategies yielded higher transfer performance than control strategies, and this superiority became more evident on far transfer problems for which learners had to design and construct new computer programs that required solutions not encountered before. Also in other domains including statistical analyzing (Paas, 1992, 1993), computer numerically controlled programming (Paas & van Merriënboer, 1994), and fault management in process industry (de Croock, 1999; de Croock, van Merriënboer, & Paas, 1998; Jelsma, 1989) training strategies that followed 4C/ID principles were designed and tested, and results supported the main predictions of the 4C/ID-model.

At present, more studies are being carried out that investigate in greater detail important aspects of the 4C/ID-model, including the timing of information presentation (Kester, Kirschner, van Merriënboer, & Bäumer, 2001), modalities of information presentation (Tabbers, Martens, & van Merriënboer, 2001), and optimal step sizes in process worksheets for learning tasks (Nadolski, Kirschner, van Merriënboer, & Hummel, 2001). It is our strong conviction that only research-based models and methodologies may be strong enough to move the huge, slow, and ponderous ocean liner of instructional design a few degrees off its current path. ☐

Jeroen J. G. van Merriënboer [jeroen.vanmerrienboer@ou.nl] is Professor of Educational Technology at the Open University of the Netherlands, Heerlen, The Netherlands.

Richard E. Clark is Professor of Education at the Rossier School of Education, University of Southern California.

Marcel B.M. de Croock is Assistant Professor of Educational Technology at the Open University of the Netherlands, Heerlen, The Netherlands.

Part of the work presented here was carried out within the ADAPT[IT] project, sponsored by the European Commission under contract IST-1999-11740.

Correspondence concerning this article should be addressed to Jeroen J. G. van Merriënboer, Open University of the Netherlands, Educational Technology Expertise Center (OTEC), P.O. Box 2960, NL-6401 DL Heerlen, The Netherlands.

REFERENCES

Ackerman, P.L. (1990). A correlation analysis of skill specificity: Learning, abilities and individual differences. *Journal of Experimental Psychology: Learning, Memory and Cognition, 16,* 883–901.

Anderson, J.R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

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

Anderson, J.R., & Lebiere, C. (1998). *The atomic com-*

ponents of thought. Mahwah, NJ: Lawrence Erlbaum Associates.

Balzer, W.K., Doherty, M.E., & O'Connor, R. (1989). Effects of cognitive feedback on performance. *Psychological Bulletin, 106,* 410–433.

Berryman, S.E. (1993). Learning in the workplace. In L. Darling-Hammond (Ed.), *Review of research in education* (Vol. 19, pp. 343–401). Washington, DC: American Educational Research Association.

Butler, D.L., & Winne, P.H. (1995). Feedback and self-regulated learning: A theoretical synthesis. *Review of Educational Research, 65,* 245–281.

Carlson, R.A., Khoo, H., & Elliot, R.G. (1990). Component practice and exposure to a problem-solving context. *Human Factors, 32,* 267–286.

Carroll, J.M., Smith-Kerker, P.L., Ford, J.R., & Mazur-Rimetz, S.A. (1988). The minimal manual. *Human-Computer Interaction, 3,* 123–153.

Cascio, W.F. (1995). Whither industrial and organizational psychology in a changing world of work? *American Psychologist, 50,* 928–939.

Clark, R.E. (1988). When teaching kills learning: Research on mathemathanics. In H. Mandl, E. DeCorte, N. Bennett, & H.F. Friedrich (Eds.), *Learning and instruction: European research in an international context* (Vol. 2, pp. 1–22). Oxford, UK: Pergamon Press.

Clark, R.E., & Estes, F. (1999). The development of authentic educational technologies. *Educational Technology, 39*(2), 5–16.

Collins, A., Brown, J.S., & Newman, S. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L.B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Corneille, O., & Judd, C.M. (1999). Accentuation and sensitization effects in the categorization of multifaceted stimuli. *Journal of Personality and Social Psychology, 77,* 927–941.

de Croock, M.B.M. (1999). *The transfer paradox: Training design for troubleshooting skills.* Unpublished PhD Thesis. Enschede, The Netherlands: University of Twente.

de Croock, M.B.M., van Merriënboer, J.J.G., & Paas, F.G.W.C. (1998). High vs. low contextual interference in simulation-based training of troubleshooting skills: Effects on transfer performance and invested mental effort. *Computers in Human Behavior, 14,* 249–267.

de Jong, T., & van Joolingen, W.R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68,* 179–201.

Driscoll, M., & Dick, W. (1999). New research paradigms in instructional technology: An inquiry. *Educational Technology Research and Development, 47*(2), 7–18.

Dufresne, R.J., Gerace, W.J., Thibodeau-Hardiman, P., & Mestre, J.P. (1992). Constraining novices to perform expertlike problem analyses: effects on schema acquisition. *The Journal of the Learning Sciences, 2,* 307–331.

Fisk, A.D., & Gallini, J.K. (1989). Training consistent components of tasks: Developing an instructional system based on automatic-controlled processing principles. *Human Factors, 31,* 453–463.

Gagné, R.M., Briggs, L.J., & Wager, W.W. (1992). *Principles of instructional design* (4th ed.). New York, NY: Holt, Rinehart and Winston.

Gick, M., & Holyoak, K.J. (1983). Schema induction and analogical transfer. *Cognitive Psychology, 15,* 1–38.

Gropper, G.L. (1983). A behavioral approach to instructional prescription. In C.M. Reigeluth (Ed.), *Instructional design theories and models* (pp. 101–161). Hillsdale, NJ: Lawrence Erlbaum Associates.

Jelsma, O. (1989). *Instructional control of transfer.* Enschede, The Netherlands: Bijlstra & van Merriënboer.

Jonassen, D. (1994). Thinking technology: Toward a constructivist design model. *Educational Technology, 34*(4), 34–37.

Kester, L., Kirschner, P.A., van Merriënboer, J.J.G., & Bäumer, A. (2001). Just-in-time information presentation and the acquisition of complex cognitive skills. *Computers in Human Behavior, 17,* 373–391.

Kluger, A., & DiNisi, A. (1998). Feedback interventions: Toward the understanding of a double-edged sword. *Current Directions in Psychological Science, 7*(3), 67–72.

Kozma, R. (2000). Reflections on the state of educational technology. *Educational Technology Research and Development, 48*(1), 5–15.

Landa, L.N. (1983). The algo-heuristic theory of instruction. In C.M. Reigeluth (Ed.), *Instructional design theories and models* (pp. 163–211). Hillsdale, NJ: Lawrence Erlbaum Associates.

Leutner, D. (2000). Double-fading support-a training approach to complex software systems. *Journal of Computer Assisted Learning, 16,* 347–357.

Merrill, M.D. (1983). Component display theory. In C.M. Reigeluth (Ed.). *Instructional design theories and models* (pp. 278–333). Hillsdale, NJ: Lawrence Erlbaum Associates.

Merrill, M.D. (1999). Instructional transaction theory: Instructional design based on knowledge objects. In C.M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory* (Vol. 2). Mahwah, NJ: Lawrence Erlbaum Associates.

Merrill, M.D. (2000, October). *First principles of instruction.* Paper presented at the international conference of the Association for Educational Communications and Technology (AECT), Denver, CO.

Nadolski, R.J., Kirschner, P.A., van Merriënboer, J.J.G., & Hummel, H.G.K. (2001). A model for optimizing step size of learning tasks in competency-based multimedia practicals. *Educational Technology Research and Development, 49(3),* 87–103.

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

Ohlsson, S., & Rees. E. (1991). The function of conceptual understanding in the learning of arithmetic procedures. *Cognition and Instruction, 8,* 103–179.

Paas, F.G.W.C. (1992). Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive load approach. *Journal of Educational Psychology, 84,* 429–434.

Paas, F.G.W.C. (1993). *Instructional control of cognitive load in the training of complex cognitive tasks.* Unpublished PhD Thesis. Enschede, The Netherlands: University of Twente.

Paas, F.G.W.C., & van Merriënboer, J.J.G. (1994). Variability of worked examples and transfer of geometrical problem solving skills: A cognitive load approach. *Journal of Educational Psychology, 86,* 122–133.

Pellegrino, J.W., Doane, S.M., Fischer, S.C., & Alderton, D. (1991). Stimulus complexity effects in visual comparisons: The effects of practice and learning context. *Journal of Experimental Psychology: Human Perception and Performance, 17,* 781–791.

Perkins, D.N., & Grotzer, T.A. (1997). Teaching intelligence. *American Psychologist, 52,* 1125–1133.

Pointe, L.B., & Engle, R.W. (1990). Simple and complex word spans as measures of working memory capacity. *Journal of Experimental Psychology: Learning, Memory and Cognition, 16,* 1118–1133.

Reigeluth, C.M. (Ed.) (1999a). *Instructional design theories and models: A new paradigm of instructional theory* (Vol. 2). Mahwah, NJ: Lawrence Erlbaum Associates.

Reigeluth, C.M. (1999b). The elaboration theory: Guidance for scope and sequence decisions. In C.M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory* (Vol. 2). Mahwah, NJ: Lawrence Erlbaum Associates.

Reigeluth, C.M., & Stein, F. (1983). The elaboration theory of instruction. In C.M. Reigeluth (Ed.), *Instructional design theories and models* (pp. 335–381). Hillsdale, NJ: Lawrence Erlbaum Associates.

Richey, R. (1998). The pursuit of useable knowledge in instructional technology. *Educational Technology Research and Development, 46*(4), 7–22.

Salisbury, D.F., Richards, B.F., & Klein, D. (1985). Designing practice: A review of prescriptions and recommendations from instructional design theories. *Journal of Instructional Development, 8*(4), 9–19.

Schneider, W. (1985). Training high-performance skills: Fallacies and guidelines. *Human Factors, 27,* 285–300.

Schneider, W., & Detweiler, M. (1988). The role of practice in dual-task performance: Toward workload modeling in a connectionist/-control architecture. *Human Factors, 30,* 539–566.

Schuurman, J.G. (1999). *On the proper treatment of learning and transfer: A study on introductory computer programming.* Unpublished PhD Thesis. Enschede, The Netherlands: University of Twente.

Schwartz, D., Brophy, S., Lin, X., & Bransford, J. (1999). Flexibly adaptive instructional design: A case study from an educational psychology course. *Educational Technology Research and Development, 47*(2), 39–60.

Sweller, J., van Merriënboer, J.J.G., & Paas, F.G.W.C. (1998). Cognitive architecture and instructional design. *Educational Psychology Review, 10,* 251–296.

Tabbers, H.K., Martens, R.L., & van Merriënboer, J.J.G. (2001, August). *The modality effect in multimedia instruction.* Paper presented at the 23rd Annual Conference of the Cognitive Science Society (Cogsci), Edinburgh, Scotland.

van der Meij, H., & Carroll, J.M. (1995). Principles and heuristics for designing minimalist instruction. *Technical Communication, 42,* 243–261.

van Merriënboer, J.J.G. (1990a). Strategies for programming instruction in high school: Program completion vs. program generation. *Journal of Educational Computing Research, 6,* 265–287.

van Merriënboer, J.J.G. (1990b). *Teaching introductory computer programming: A perspective from instructional technology.* Enschede, The Netherlands: Bijlstra & van Merriënboer.

van Merriënboer, J.J.G. (1997). *Training complex cognitive skills: A four-component instructional design model for technical training.* Englewood Cliffs, NJ: Educational Technology Publications.

van Merriënboer, J.J.G., & de Croock, M.B.M. (1992). Strategies for computer-based programming instruction: Program completion vs. program generation. *Journal of Educational Computing Research, 8,* 365–394.

van Merriënboer, J.J.G., & Dijkstra, S. (1996). The four-component instructional design model for training complex cognitive skills. In R.D. Tennyson & F. Schott (Eds.), *Instructional design: International perspectives. Theory and research* (Vol. 1, pp. 427–445). Hillsdale, NJ: Lawrence Erlbaum Associates.

van Merriënboer, J.J.G., Jelsma, O., & Paas, F.G.W.C. (1992). Training for reflective expertise: A four-component instructional design model for training complex cognitive skills. *Educational Technology Research and Development, 40*(2), 23–43.

van Merriënboer, J.J.G., & Luursema, J.J. (1996). Implementing instructional models in computer-based learning environments: A case study in problem selection. In T.T. Liao (Ed.), *Advanced educational technology: Research issues and future potential* (NATO ASI Series F, Vol. 145, pp. 184–206). Berlin, Germany: Springer Verlag.

van Merriënboer, J.J.G., Schuurman, J.G., de Croock, M.B.M., & Paas, F.G.W.C. (2002). Redirecting learners' attention during training: Effects on cognitive load, transfer test performance and training efficiency. *Learning and Instruction, 12,* 11–37.

Wenger, J.L., & Carlson, R.A. (1996). Cognitive sequence knowledge: What is learned? *Journal of Experimental Psychology: Learning, Memory and Cognition, 22,* 599–619.