Schaafstal, A .M. (1991). *Diagnostic skill in process operation: A comparison between experts and novices.* Unpublished Doctoral Dissertation, University of Groningen, The Netherlands.

Schaafstal, A. M. (1993). Knowledge and strategies in diagnostic skill. *Ergonomics, 36,* 1305–1316.

Schaafstal, A. M. & Schraagen, J. M. C. (1992). *A method for cognitive task analysis* (Technical Report IZF 1992 B-5). Soesterberg, The Netherlands: TNO Human Factors Research Institute.

Schaafstal, A. M., & Schraagen, J. M. C. (1993). *Knowledge-based support for the training of troubleshooting in a Sensor, Weapon, and Command subsystem: the* SMR_{μ}. (Technical Report IZF 1993 A-49). Soesterberg, The Netherlands: TNO Institute for Perception (in dutch).

Schraagen, J. M. C. & Schaafstal, A. M. (1991). *Fault diagnosis in a Sensor, Weapon and Command subsystem: the LW08 radar.* (Technical Report IZF 1991 A-36). Soesterberg, The Netherlands: TNO Institute for Perception (in dutch).

Schraagen, J. M. C. & Schaafstal, A. M. (1996). Training of systematic diagnosis: A case study in electronics troubleshooting. *Le Travail Humain, 59*(1), 5–21.

Simon, H. A. (1981). *The sciences of the artificial* (2nd ed.). Cambridge, MA: The MIT Press.

Singley, M.K. & Anderson, J.R. (1989). *The transfer of cognitive skill.* Cambridge, MA: Harvard University Press.

5

DNA: PROVIDING THE BLUEPRINT FOR INSTRUCTION

Valerie J. Shute
Air Force Research Laboratory (AFRL)

Lisa A. Torreano
National Research Council Fellow at AFRL

Ross E. Willis
Galaxy Scientific, Corp.

Cognitive task analysis (CTA) refers to a diverse collection of methods for obtaining information from expert sources. The techniques are united by the common goal of representing the underlying knowledge, skills, and goal structures of a particular domain or task performance (e.g., Dehoney, 1995; Gordon & Gill, 1992; Hall, Gott, & Pokorny, 1995; Schraagen et al., 1997). Recently, conventional CTA methods have come under attack for a variety of reasons. Most notably, the great expense (in time, personnel, and money) associated with conducting a CTA is often cited as a major obstacle to employing them more widely. That is, a typical CTA process involves the support of many individuals, such as instructional designers, cognitive psychologists, knowledge engineers, and subject-matter experts (SMEs), across long periods of time (e.g., Durkin, 1994; Gordon, Schmierer, & Gill, 1993; Hayes-Roth, Waterman, & Lenat, 1983). A more subtle, but important criticism of CTA has to do with a general sense of vagueness and imprecision. For example, in their recent review of the field, Schraagen et al. (1997) concluded, "that little attention is being paid to the conditions under which methods are appropriate, and that often it is unclear how products of cognitive task analysis should be used" (p. 5).

This chapter describes our response to these issues. Specifically, can we automate any part of this process without sacrificing accuracy and do so in a principled manner? Other researchers have tackled this issue of automating knowledge acquisition and representation (for excellent reviews of such tools, see Lethbridge, 1994; Williams & Kotnur, 1993). Generally, these knowledge-elicitation tools have not made it into the mainstream for various reasons–most notably due to limitations in the breadth of data that may be captured (e.g., only conceptual knowledge by Lethbridge's CODE4 program) and being difficult to be used by persons other than programmers. We describe a computer-based cognitive tool that has been designed to aid in knowledge elicitation and organization for instruction and training purposes, capturing a range of data in a user-friendly manner. Our tool is called DNA, which stands for decompose a domain into its constituent elements, network the elements into an inheritance hierarchy, and assess the ensuing knowledge structure for validity and reliability (for a detailed description of the program, see Shute, Torreano, & Willis, in press-a).

DNA's particular target application, in relation to CTA, is to obtain curriculum elements to be used within instructional systems—specifically, intelligent tutoring systems (ITSs) that generally contain three distinct core components: expert model, student model, and instructor model. The expert model contains a curriculum map consisting of expert knowledge and skills. The student model monitors a learner's prior and emergent knowledge and skills and gauges whether he or she has demonstrated mastery of the elements included in the expert model. Basically, the student model is an online report card that indicates whether a student has mastered the curriculum elements contained in the expert model. Finally, the instructor model determines a student's path through the instructional content based on discrepancies between the expert model and the student model. These three components interact to individualize instruction based on a learner's needs. DNA's job, then, is to facilitate expert model development.

The organization of our chapter is as follows. First, we briefly discuss the two previously cited CTA-related issues in relation to DNA design features. We continue with a description of DNA's structure and components, followed by an illustration of how the program works in the context of dealing with a real-world situation (i.e., obtaining elements for a specific curriculum). We conclude with a summary of the general CTA-related issues and our program's fit in the realm of CTA.

CTA ISSUES

As mentioned, the design of DNA was motivated, in part, as a response to some of the problems cited earlier. The following paragraphs discuss DNA in relation to these CTA-related issues.

Maximizing Cost–Benefit Ratio

The first issue influencing DNA's design concerns improving the cost–benefit ratio of analyzing a domain. That is, typical CTA methods involve numerous steps and personnel. First, the instructional designer employs a trained cognitive task analyst. The analyst prepares for upcoming interviews with experts by becoming familiar with the target domain (e.g., via document analysis).

Subsequently, the analyst conducts extensive interviews with one or more experts in the field. Following the interviews, the analyst transcribes the protocols and translates and organizes the knowledge and skill elements into individual curriculum elements (CEs). Finally, the instructional designer needs to transform individual CEs into a coherent curriculum. This process typically requires several months and many person hours to achieve (e.g., Gordon, Babbitt, Sorensen, Bell, & Crane, 1993). Thus, the cost–benefit ratio has room for improvement.

DNA attempts to automate the bulk of the interview process, thus improving efficiency by decreasing the personnel resources required to conduct the analysis. Additional time is saved because lengthy transcription sessions are eliminated. That is, DNA immediately stores all SME input in a database of CE records. In addition, the two common phases of CTA–elicitation and organization of knowledge–have been collapsed into a symbiotic process to decrease the time required for the SME. The SME, working with the DNA program, not only identifies various CEs, but also arrays them into a hierarchical structure.

Specifying Purpose, Domain, and Representation

To avoid the vagueness issue cited previously, throughout the design process, we have been careful to explicate DNA's purpose, including suitable domains and desired representation. DNA's primary purpose is to obtain curriculum for use in intelligent instructional software. It was not designed to be an all-purpose tool. Within instruction, there are some domains for which it is appropriate and others for which it is not. The most fitting domains or tasks are those containing structured, explicit knowledge and skills. There are many domains that fall into this category in academic, military, and industrial settings. Domains or tasks where DNA is less suitable include those that place a large emphasis on psychomotor skill and those that contain a lot of implicit knowledge. In the former case, observational techniques such as videotaping on-the-job perfor-mance may be appropriate to capture relevant psychomotor data. In the latter case, acquiring implicit knowledge may be achieved via card sorting and other multidimensional scaling techniques or think-aloud protocols. Finally, we have considered the needed representation given the domains to be analyzed for our purposes of instruction. As is described in greater detail in the next section, because DNA is intended to help build instruction across a variety of topics, it was designed to capture a range of knowledge and skill types. Consequently, DNA represents information in a hybrid structure—combining aspects of production systems (i.e., if–then rules related to procedural knowledge and skills; see Anderson, 1993) as well as conceptual graphs (e.g., semantic nets of concepts and their relationships; see Gordon et al., 1993).

DNA COMPONENTS AND STRUCTURE

DNA is a software program designed to streamline development of ITSs by automating part of the process of constructing the expert model or curriculum map. Although DNA can theoretically provide the foundation for any curriculum, computer administered or not, it was specifically designed to interface with a student-modeling paradigm called *SMART* (Student Modeling Approach to Responsive Tutoring; Shute, 1995).

DNA and SMART are intended to work together. DNA provides the blueprint for instruction by obtaining curriculum elements directly from the responses and actions of SMEs who answer structured queries posed by the computer. That is, it supplies the instructional system with an expert model by eliciting and organizing knowledge and skills from human experts. SMART uses the resulting blueprint (knowledge structure) as the basis for assessment, cognitive diagnosis, and instruction, managing the learning environment based on the discrepancies between the expert and student models. To optimally manage such divergences, SMART requires the identification of individual units of instruction (i.e., the curricular elements) as well as their relationships (hierarchical dependencies). Hierarchical dependencies indicate prerequisite knowledge and skills that are used when developing the instructional path through the curriculum. Given these requirements, the goals for DNA's output are to identify CEs and their relationships (Shute et al., in press-a).

Because they are meant to work collaboratively, SMART influenced key DNA design decisions affecting the intended nature of the tool's knowledge representation or output. In particular, SMART is intended to be useful across a large array of instructional content. Consequently, it recognizes and monitors a variety of knowledge types (symbolic, procedural, and conceptual). In addition, SMART is intended to manage instruction in a principled manner based on dependency relations among knowledge elements. Thus, it uses information from hierarchical knowledge structures to select curriculum units for instruction.

As a result, DNA was designed to capture a range of knowledge types as well as their hierarchical structure. Specifically, DNA operates via a series of interactive modules for decomposing, networking, and assessing experts' knowledge structures in relation to a topic. The Decompose module acquires and classifies experts' knowledge into the knowledge types that are recognized and monitored by SMART. Its focus is to analyze the domain, breaking its knowledge down into constituent elements. The Network module, an organizational tool, allows experts to connect their extracted knowledge components into an integrated hierarchical structure of CEs. Its focus is the synthesis of the knowledge units. Finally, the Assess module (when implemented) will allow the instructional designer to ascertain the validity and reliability of the extracted knowledge structures obtained from SMEs–an important aspect of any CTA approach. Collectively, the DNA-elicited elements and hierarchical structure provide guidance to SMART in terms of how to structure, track, and modify the ensuing curriculum.

We now present more detail on the particular knowledge types and structures that both SMART and DNA distinguish. The purpose is to extend the range of applicability across domains, as well as provide a coherent basis on which to make principled decisions regarding instruction.

Knowledge Types

SMART and DNA recognize three knowledge types: symbolic, procedural, and conceptual (also known as what, how, and why knowledge, respectively). Symbolic knowledge refers to fundamental information including definitions, symbols, formulas, and other basic depictions. Procedural knowledge relates to specific steps and conditionals underlying a particular procedure or way to accomplish a goal. Conceptual knowledge provides the rationale or "big picture" relating lower level elements together into a system or whole (e.g., Kyllonen & Shute, 1989; Shute, 1994). The reason DNA makes these knowledge-type distinctions is because of its interface with SMART, which provides differential instruction and assessment based on the categorization. To illustrate the differential instruction, consider one piece of knowledge—the statistical mean. Symbolic knowledge elements, such as the formula for the mean (e.g., $\Sigma X/N$), are introduced in a straightforward, didactic manner and assessed by requiring the recognition and/or generation of an answer. Instruction and assessment of procedural elements, however, can focus either on the knowledge of the rules of a procedure (procedural knowledge) or the competency with which the procedure is executed (procedural skill). For example, procedural knowledge may require the generation of the procedure's relevant steps (e.g., add all numbers in a distribution of data, then divide by the total sample size). In contrast, procedural skill might be instructed in the context of an interactive, problem-solving environment and assessed via the application of some rule or procedure (e.g., calculate the mean of a given data set). Finally, instruction of conceptual knowledge may use analogies or the explicit presentation of the big picture, whereas assessment requires extrapolation, induction, deduction, and transfer to novel areas. For example, conceptual knowledge of the mean involves understanding and articulating its relationship to other measures of central tendency within various distributions.

As we have shown, the knowledge-type distinction is important to SMART in that it allows for more customized instructional and assessment techniques to be employed. Consequently, DNA is responsible for acquiring and categorizing data in accord with these three knowledge types. This is accomplished via queries embedded within the program that are specifically crafted to focus on eliciting what, how, and why knowledge. We now turn our attention to the importance of knowledge structures.

Knowledge Structures

Although the Decompose module of DNA is intended to elicit and embellish a variety of curriculum elements, the Network module is designed to capture the structure of the experts' knowledge (Chi, Glaser, & Rees 1982; Dochy, 1992; Jonassen, Beissner, & Yacci, 1993). These structures contain inheritance relationships among CEs that can provide the basis for assessment (i.e., What is the current status of a particular CE?), cognitive diagnosis (i.e., What is the source of the problem, if any?), and instruction (i.e., What needs to be taught now?). To enable the structuring of the range of CE types captured during the decomposition, the Network module includes provisions for approximating both a production system framework to represent procedural elements and a conceptual graph structure to reflect conceptual elements, as well as the relationship between the two.

Ultimately, it is structured knowledge that provides the curriculum that we, as educators, research scientists, and/or designers of intelligent instructional software, wish to impart to learners (Jonassen, Beissner, & Yacci, 1993; Reigeluth, 1983). The reason for doing so is straightforward: "Structured knowledge enables inference capabilities, assists in the elaboration of new information, and enhances retrieval. It provides potential links between stored knowledge and incoming information, which facilitates learning and problem solving" (Glaser & Bassok, 1989, p. 26).

Sound CTA approaches consider the nature of the knowledge of the domain being analyzed and use an appropriate representation to reflect it. Given that DNA is designed for developing curriculum for a range of topics, both conceptual and procedural in nature, it is important that this tool be able to accommodate a range of knowledge types. Having been designed to accommodate the knowledge classification used by SMART, and by providing an organizational aid to support structuring of procedural and conceptual knowledge domains, our CTA tool should be readied to accomplish this goal.

DNA: How Does it Work?

The Situation. Two of the authors were posed with the task to develop a 2-hour training program for medical personnel on how to use the Internet to access pertinent data. To develop the curriculum, we used the Decompose module of DNA to collect relevant elements and procedures on this topic. We present our actual interactions with the system to illustrate how DNA operates to obtain relevant symbolic, procedural, and conceptual knowledge for the domain "getting data from the Internet." This illustration should provide a general sense of how the program works. For more details on design and functionality, see Shute, et al. (in press-a).

Interaction Among DNA Program Components

As mentioned, DNA consists of several modules that interact with one another. The instructional designer (ID) begins with the Customize module and answers a fairly short series of questions. During the process, he or she specifies the domain of interest, delineates the learning goals for the course, identifies the incoming knowledge and skills of the learner population (targeted to receive the course), and indicates the desired flavor of instruction (e.g., teaching the curriculum from a predominately procedural perspective). After the ID completes the Customize module, the program automatically generates a customized form letter that is sent to prospective SMEs, in conjunction with the DNA program, initialized to reflect the input by the ID. Table 5.1 shows the actual letter produced by the customize module, which is generated to reflect the requirements of this Internet curriculum.

TABLE 5.1
Edited Output From Customize Module Detailing the Curriculum for the Expert

Dear [Expert's Name],

We're writing today to get your help in designing a course on getting data from the Internet. Before you begin working with the enclosed program, please sit down and think about the critical things that make you good at getting data from the Internet.

As you go through the enclosed program and respond to our questions, try to respond in terms of how you currently perform the job or think about the particular task. Please don't respond with how you originally learned to get data from the Internet; you have probably developed much better ways of performing this task since then.

The ultimate goals of the course are for our trainees to:

(1) Identify the components used to access the WWW
(2) Identify the syntax and semantics of a URL
(3) Identify various search engines
(4) Know how to connect to the Internet
(5) Know how to bring up various search engines

(continued)

(6) Know how to input keywords for efficient searching (strategies & tricks)
(7) Know how to navigate to a web site from a hot link
(8) Know how to navigate to a web site given any URL
(9) Know how to get information from the WWW in your hands or into your PC
(10) Know how to download files from a web page
(11) Know how to bookmark page(s) for later viewing
(12) Understand the relationship among elements that allow you to view a page on the WWW
(13) Understand how search engines retrieve relevant data

How specific should you get? You can presume that our trainees will have the following knowledge and skills: (1) familiarity with Windows 95/98 environment, (2) basic knowledge of the Internet, and (3) basic typing skills. Therefore, you will not need to define knowledge or skills at a detailed level in relation to these elements.

When answering questions during the program, please make your responses consistent with the following distribution: What: 34%; How: 54%; Why: 12%. As you can see, the curriculum is intended to emphasize procedural knowledge, but also includes "what" and a bit of the "why" knowledge.

Thanks very much for your time.

Sincerely,

[Signature]

The SME receives the letter and begins the program. DNA starts with a 5- to 10-minute orientation that provides a general overview of the program and includes a description of the different knowledge types (introduced as what, how, and why kinds of knowledge). There is also a summary of how the program operates. After the brief orientation, the expert begins the Decompose module.

Decompose Module

The Decompose module asks what, how, and why questions via a semi-structured, interactive dialogue with the expert. A different set of questions is posed to the expert depending on whether symbolic, procedural, or conceptual knowledge is currently being decomposed. That is, to differentiate the knowledge comprising a domain, different paths of interrogation have been designed to capture information relevant to each particular knowledge type. Each path (i.e., What—Symbolic knowledge, How—Procedural knowledge, and Why—Conceptual knowledge) has its own interface to accomplish that goal. For example, to elicit symbolic knowledge, DNA probes for definitions, examples, and supplemental multimedia links. To elicit procedural knowledge, DNA probes for detailed, step-by-step information, logical and temporal relations among steps, and subprocedures. For obtaining conceptual knowledge, DNA focuses on obtaining relational information among concepts. Responses in all paths are made via selecting items from list boxes and typing directly into text boxes.

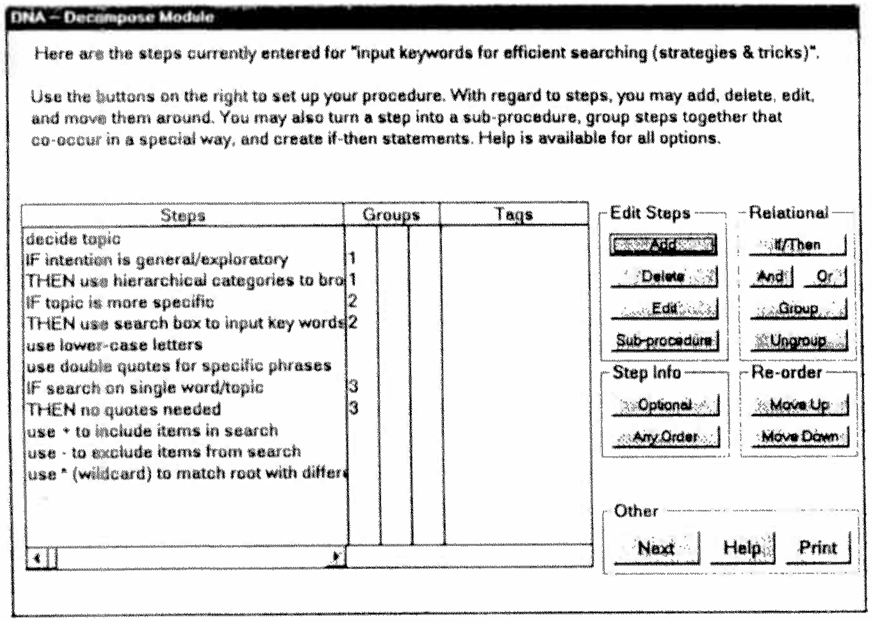
One of the three inquiry paths is launched when the SME selects a question listed in the Main Question Queue–the first screen of the Decompose module. To initialize the list of questions, the program restates the ID-provided learning goals in question formats consistent with the three knowledge types. For example, the fourth goal (see Table 5.1) for the topic "getting data from the Internet" was for students to "know how to connect to the Internet"–a procedural knowledge goal. The restated question listed in the Main Question Queue, which would launch the How path, became: "What are the steps that you go through when you connect to the Internet?"

The following paragraphs illustrate portions of each path in action when surfing the Internet was being decomposed. We start with the How path because the dominant flavor of the curriculum to be achieved is procedural (i.e., 8/13 of the listed learning goals are procedural knowledge).

How Path. The How path is used to elicit procedural knowledge. The primary interface is the Step Editor (see Fig. 5.1), which the expert uses to build procedures using options to list steps, establish conditional statements (i.e., If/Then), and include necessary connectors (i.e., and–or). Fig. 5.1 presents the steps for finding information on the Internet corresponding to Goal 6 in table 5.1.

Notice that If/Then items are automatically grouped and function as a single unit of information. The numbers (i.e., 1, 2, 3) in the Grouping column indicate that elements sharing identical numbers belong to the same group. For example, the user cannot separate an If statement from its associated Then statement. Other features include tagging steps with additional information, such as whether a step is optional or required.

A series of follow-up questions will be implemented in the next version of the Decompose module. They are intended to clarify procedures that may be ambiguous. For example, if an expert outlined the procedure: DO A AND B OR C (without adequate grouping), the program would ask if the procedure was: {DO (A AND B) OR C} or {DO A AND (B OR C)}. Follow-up questions also require the expert to consider *alternative* conditions and consequences. For instance, suppose the expert created the conditional: IF X, THEN Y. The program would ask if



DNA – Decompose Module

Here are the steps currently entered for "input keywords for efficient searching (strategies & tricks)".

Use the buttons on the right to set up your procedure. With regard to steps, you may add, delete, edit, and move them around. You may also turn a step into a sub-procedure, group steps together that co-occur in a special way, and create if-then statements. Help is available for all options.

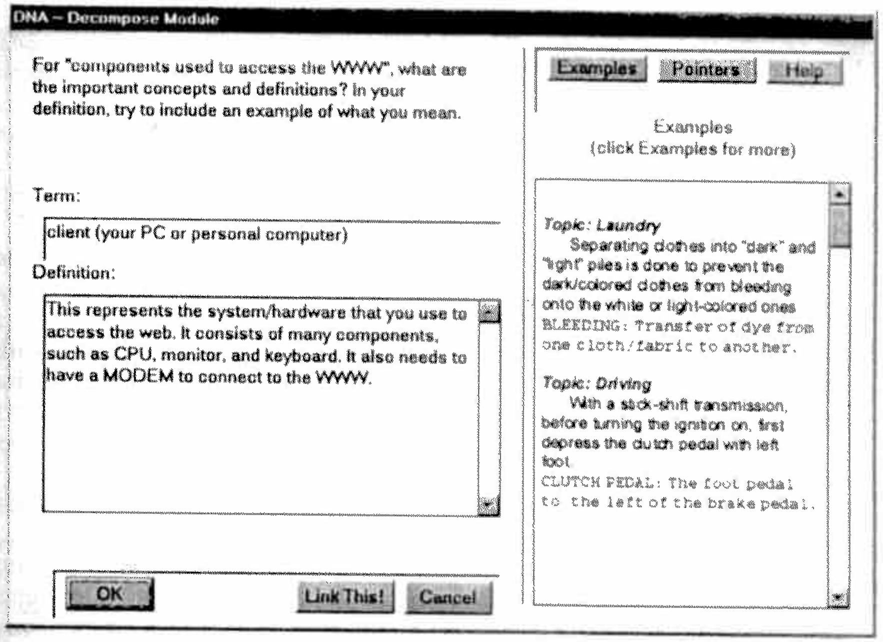
Steps	Groups	Tags
decide topic		
IF intention is general/exploratory	1	
THEN use hierarchical categories to bro	1	
IF topic is more specific	2	
THEN use search box to input key words	2	
use lower-case letters		
use double quotes for specific phrases		
IF search on single word/topic	3	
THEN no quotes needed	3	
use + to include items in search		
use - to exclude items from search		
use * (wildcard) to match root with differ		

FIG. 5.1. The Step Editor interface from DNA's How path illustrating a procedure.

there are other conditions, besides X, that could invoke Y, and whether there are actions, other than Y, that may ensue from X.

With these follow-up questions, in conjunction with the user-friendly interface, we anticipate that even individuals who are naive to control structure (or logic) will be able to provide descriptions that will enable context-dependent reasoning. This is important because it is conceptual knowledge (i.e., the underlying *why* and *how* a step is important toward reaching the goal state) that constrains a procedure. Therefore, having a conceptual understanding of a procedure enables one to make inferences during task performance and thus respond appropriately when faced with novel circumstances.

What Path. The What path is a series of screens used to elicit symbolic knowledge. Fig. 5.2 depicts the interface after the expert has entered some relevant terms and definitions related to the topic being decomposed. For instance, in Fig. 5.2, the expert defined the term "client" and attached a graphics file to this definition (i.e., an image of a personal computer). This associated picture attachment is denoted by the Mona Lisa icon located below the definition. Experts



DNA – Decompose Module

For "components used to access the WWW", what are the important concepts and definitions? In your definition, try to include an example of what you mean.

Term:

client (your PC or personal computer)

Definition:

This represents the system/hardware that you use to access the web. It consists of many components, such as CPU, monitor, and keyboard. It also needs to have a MODEM to connect to the WWW.

Examples
(click Examples for more)

Topic: Laundry
Separating clothes into "dark" and "light" piles is done to prevent the dark/colored clothes from bleeding onto the white or light-colored ones.
BLEEDING: Transfer of dye from one cloth/fabric to another.

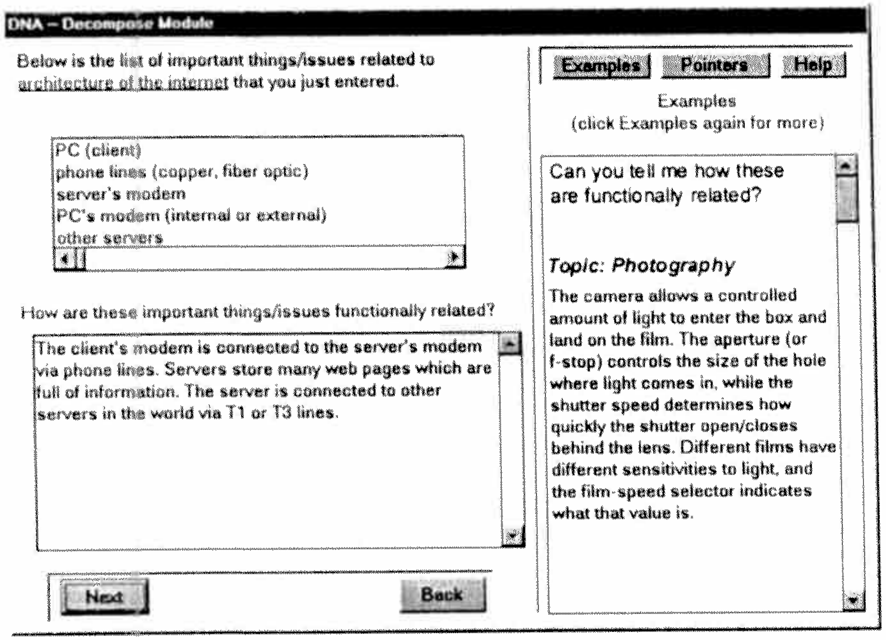
Topic: Driving
With a stick-shift transmission, before turning the ignition on, first depress the clutch pedal with left foot.
CLUTCH PEDAL: The foot pedal to the left of the brake pedal.

FIG. 5.2. Interface from DNA's What path defining an Internet-related term.

are free to attach sound files, videos, and other multimedia embellishments in computer or hard-copy format.

Additional features in the What path, shown on the right-hand side of Fig. 5.2, include examples to help guide the expert when composing definitions. Additionally, the expert can view answering pointers that explain how to best phrase responses so the wording in future questions posed by DNA will flow smoothly. Similar support is available throughout each path of inquiry.

Why Path. The Why path is used to elicit conceptual knowledge. The interface consists of a series of questions that asks for information that enables the expert to successively build up the concept currently in focus. The first question asks the expert to list the main components underlying the current concept or topic being decomposed (e.g., architecture of the internet). The expert identified several components that comprise the Internet, such as: PC (personal computer, or client), phone lines (copper, fiber optic), server's modem, PC's modem (internal, external), and so on. Fig. 5.3 shows the second screen of the Why path. The list of components entered on the first screen are presented and the



Below is the list of important things/issues related to architecture of the internet that you just entered.

PC (client)
phone lines (copper, fiber optic)
server's modem
PC's modem (internal or external)
other servers

How are these important things/issues functionally related?

The client's modem is connected to the server's modem via phone lines. Servers store many web pages which are full of information. The server is connected to other servers in the world via T1 or T3 lines.

Examples
(click Examples again for more)

Can you tell me how these are functionally related?

Topic: Photography

The camera allows a controlled amount of light to enter the box and land on the film. The aperture (or f-stop) controls the size of the hole where light comes in, while the shutter speed determines how quickly the shutter open/closes behind the lens. Different films have different sensitivities to light, and the film-speed selector indicates what that value is.

FIG. 5.3. Interface from DNA's Why path describing relationships among components.

expert is asked to describe how these components are functionally related to one another. The experts accomplish this via input to the text box that can hold up to 16,000 characters.

The third question/screen asks, "Why is having a functional understanding of Internet architecture (i.e., the current topic of query) important in relation to getting data from the Internet" (i.e., the overall domain being decomposed)? One expert response was, "The Internet is comprised of millions of individual web pages, each connected to its own server, and each server is connected to all other servers. This provides a framework for understanding the notion of 'the Internet' and thus how a search engine can travel to so many sites in response to a database inquiry."

All CE-related information is stored in a Microsoft Access 7.0 database containing multifarious fields such as the CE's description, unique number, knowledge type, higher order relations, and so on. When the expert has answered the initial queries from the main question queue, as well as those generated from the various paths, he or she may begin the network module.

Network Module

The Network module represents CEs from the Decompose module as graphical nodes and lets the expert arrange and link them, thereby resulting in a CE hierarchy. New CE nodes can be added, deleted, and edited, and the changes are communicated to the Decompose module via a shared database. Alternatively, if the expert wants to return to the Decompose module to make CE-related changes, that is also possible.

Nodes differ by knowledge type. That is, symbolic knowledge elements are represented as squares, procedural elements by circles, and conceptual ones as rounded rectangles. Links between nodes differ by type of association (e.g., is a, part of, parent of), strength of association (weak, moderate, and strong), and directionality (uni, bi, or none), which is shown by labels, thickness of line, and arrowheads, respectively. Finally, nesting can occur where one can zoom in or out from a particular node to see more or less detail. Rightclicking successively expands nodes that have other nodes (i.e., CEs) subsumed within.

We view one of the primary functions of the network representation as an organizational aid—to both IDs and SMEs. In general, we expect that this graphical array will make errors salient to an expert viewing the array. Problem areas may then be repaired either in the Network or Decompose module. The other main function of the graph is in terms of providing an indication of knowledge structure–whether it resembles a production system or conceptual graph dependent on the underlying content. The dependency relationships, as embodied in the hierarchical structure, should provide the basis for subsequent instructional decisions.

Assess Module

This last module of DNA, once implemented, is intended to validate conceptual graphs in terms of content, relationships, and associated material generated by different experts on the same topic. This module can be implemented in a number of ways, each offering different strengths and weaknesses. To illustrate, after the initial expert (SME-1) has created a database of elements and associated network, he or she sends the output (database and graphical array) back to the ID who sends that out to the subsequent expert (SME-2). SME-2 uses SME-1's data as the point of departure to modify, clarify, and validate the externalized knowledge structure. Alternatively, the program can be sent to multiple experts in parallel, who complete the Decompose and Network modules independently and return the knowledge structures to the ID. The former option could potentially result in an anchoring bias to the initial expert's output. Although the latter option avoids this potential anchoring problem, it offers other difficulties in the synthesis of multiple knowledge structures. Indeed, a combination of methods may prove to be the best option; multiple experts independently use the tool to represent their knowledge, followed by a series of reviews of each output by other experts for checks on consistency and validity. In short, the assessment module will allow experts to review and edit the database listing of curricular elements and graphs of other experts.

Although we have outlined assessment criterion based on consensus, we believe that additional checks of the output should also be conducted where appropriate. For example, procedural knowledge, after having been validated by a panel of experts, can be further assessed by executing the procedures outlined and determining whether, in fact, the actions produce the desired effect or accomplish the intended goal. Successful real-world implementation of outlined procedures offers independent, empirical validation of the knowledge structure.

We have attempted to provide a cursory depiction of DNA's functionality in the context of developing a real-world curriculum. Although acquisition and organization of knowledge elements constitute just a part of the overall development of the expert model, they are substantially important. Furthermore, we believe they are good candidates to be rendered more efficient. Thus we continue our attempts to automate these processes.

CONCLUSION

In conclusion, we have discussed issues that have motivated DNA's design. Specifically, our goal is to increase efficiency in the acquisition and organization of knowledge and to do so in a principled way. Our approach has been to create a new cognitive tool that automates the bulk of the interview, transcription, and organization processes, thus enhancing efficiency.

We have also responded to the criticisms of vagueness and imprecision within the CTA field by specifying the conditions under which DNA is appropriate and how its products should be used. Specifically, DNA was designed to fill a particular niche in the realm of CTA—that of providing the knowledge structure (or domain expertise) for ITSs. In contrast to the typical CTA focus on expert performance in relation to some task, DNA's purpose is to develop curriculum for a broad range of topics, procedural or conceptual in nature. The consideration of DNA's purpose and intended topics of analysis led to the implementation of principled design features (e.g., capturing a variety of knowledge types and structuring them in a hybrid representation). This infrastructure, in addition to being principled, contributes to efficiency as it was designed to fit with an existing student modeling paradigm that should streamline ITS development.

The Decompose module has now been used to elicit data from experts across various domains, including using the email software program Microsoft Exchange, descriptive statistics, and surfing the Internet. We have completed one formative evaluation of the Decompose module (see Shute, Torreano, & Willis, in press-b) and the results are quite encouraging. However, we are aware, with each new research study and subsequent version of the program, that the road to automating these processes is not on the map.

REFERENCES

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

Chi, M. T. H., Glaser R., & Rees, E. (1982). Expertise in problem solving. In R. Sternberg (Ed.), *Advances in the psychology of human intelligence* (Vol. 1, pp. 7–75). Hillsdale, NJ: Lawrence Erlbaum Associates.

Dehoney, J. (1995). *Cognitive task analysis: Implications for the theory and practice of instructional design*. Proceedings of the 1995 annual national convention of the Association for Educational Communications and Technology (AECT), Anaheim, CA.

Dochy, F. J. R. C. (1992). *Assessment of prior knowledge as a determinant of future learning*. Utrecht, The Netherlands: Lemma.

Durkin, J. (1994). Knowledge acquisition. In J. Durkin (Ed.), *Expert systems: Design and development* (pp. 518–599). New York: Macmillan.

Glaser, R., & Bassok, M. (1989). *Learning theory and the study of instruction* (Technical Report No. 11). Pittsburgh, PA: University of Pittsburgh Press.

Gordon, S.E., Babbitt, B.A., Sorensen, H.B., Bell, H. H., & Crane, P. M. (1993, November–December). *Cognitive task analysis for development of an intelligent tutoring system*. Proceedings of the 15th Interservice/Industry Training Systems and Education Conference, Orlando, FL.

Gordon, S. E., & Gill, R. T. (1992). Knowledge acquisition with question probes and conceptual graph structures. In T. Lauer, E. Peacock, & A. Graesser (Eds.), *Questions and information systems* (pp. 29–46). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gordon, S. E., Schmierer, K. A., & Gill, R. T. (1993). Conceptual graph analysis: Knowledge acquisition for instructional systems design. *Human Factors, 35*, 459–481.

Hall, E. P., Gott, S. P., & Pokorny, R. A. (1995). *A procedural guide to cognitive task analysis: The PARI methodology* (Report No. AL/HR-TR-1995-0108). Air Force Materiel Command, Brooks Air Force Base, TX.

Hayes-Roth, F., Waterman, D. A., & Lenat, D. B. (1983). *Building expert systems*. Reading, MA: Addison Wesley.

Jonassen, D. H., Beissner, K., & Yacci, M. A (1993). *Structural knowledge: Techniques for representing, conveying, and acquiring structural knowledge*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kyllonen, P. C., & Shute, V. J. (1989). A taxonomy of learning skills. In P. L. Ackerman, R. J. Sternberg, & R. Glaser (Eds.), *Learning and individual differences* (pp. 117–163). New York: Freeman.

Lethbridge, T. C. (1994). *Practical techniques for organizing and measuring knowledge*. Unpublished doctoral dissertation, Computer Science Department, University of Ottawa, Ontario, Canada.

Reigeluth, C. M. (1983). Meaningfulness and instruction: Relating what is being learned to what a student knows. *Instructional Science, 12*, 197–218.

Schraagen, J. M. C., Chipman, S. E., Shute, V. J., Annett, J., Strub, M., Sheppard, C., Ruisseau, J. Y., & Graff, N. (1997). *State-of-the-art review of cognitive task analysis techniques*. Deliverable Report of NATO Defense Research Study Group 27 on Cognitive Task Analysis. Soesterberg, The Netherlands: TNO Human Factors Research Institute (Report TM-97-B012).

Shute, V. J. (1994). Learning processes and learning outcomes. In T. Husen & T. N. Postlethwaite (Eds.), *International encyclopedia of education* (2nd ed., pp. 3315–3325). New York: Pergamon.

Shute, V. J. (1995). SMART: Student modeling approach for responsive tutoring. *User Modeling and User-Adapted Interaction, 5*, 1–44.

Shute, V. J., Torreano, L. A., & Willis, R. E. (in press-a). DNA: Towards an automated knowledge elicitation and organization tool. In S. P. Lajoie (Ed.), *Computers as cognitive tools* (Vol. 2). Hillsdale, NJ: Lawrence Erlbaum Associates.

Shute, V. J., Torreano, L. A., & Willis, R. E. (in press-b). Preliminary evaluation of an automated knowledge elicitation and organization tool. *International Journal of AI and Education.*

Williams, K. E., & Kotnur, T. G. (1993). *Knowledge acquisition: A review of manual, machine-aided and machine-learning methods*. Technical report of Office of Naval Research. Blacksburg, VA: Virginia Polytechnic Institute and State University (Interim Technical Report N00014-91-J-5-1500).

6

DISCOVERING SITUATED MEANING: AN ECOLOGICAL APPROACH TO TASK ANALYSIS

John M. Flach
Wright State University

"Experts perceive large meaningful patterns in their domain."
—Cooke, (1992, p. 33).

What does it mean for a pattern to be "meaningful"? It is surprising to me that the issue of meaning is often overlooked in studies of cognition and expertise. For example, *meaning* is not included as an index item in Hoffman's (1992) edited volume on the psychology of expertise. Further, it is not included in the indexes of other cognitive texts (Anderson, 1995; Gardner, 1985; Mayer, 1992). Halpern (1996) has two references to meaning. In one citation, Halpern claimed, in a discussion of memory, that "we can add meaning to information by elaborating on it" (p. 47). The other reference included the claim that "meaningful information is more easily remembered than nonmeaningful information" (p. 80). Solso (1995) has two index entries to meaning. Both entries refer to the interpretation or meaning of words.

It seems that, in cognitive psychology, the term *meaning* is synonymous with interpretation. The meaning of something is the interpretation of a particular agent. In this context, it makes sense to talk about *adding meaning* by the act of interpretation. Meaning comes into being when an observer relates one observation to previous experiences. This use of meaning probably has its roots in the linguistic and verbal learning traditions of cognitive psychology, where the stimuli were often words. In this context, the statement at the beginning of this section might suggest that the *meaningful patterns* exist in the mind of the expert. In other words, meaning is added to the pattern as a result of the expert's knowledge—meaning is the product of information processing.

An alternative perspective would be to define *meaning* as synonymous with *significance*. Meaningfulness might be defined independent of any particular observer or interpretation. For example, the pattern may be significant because of its implications for some function. The pattern might signify some threat or opportunity relative to the functional goals within a work domain. In this sense, the pattern would be meaningful regardless of whether that meaning was appreciated by any particular observer. One would expect that patterns that