

# Handling complex real-world data with two cognitive engineering tools: COGENT and MacSHAPA

PENELOPE M. SANDERSON

*University of Illinois at Urbana-Champaign, Urbana, Illinois*

MICHAEL D. MCNEESE

*Wright-Patterson Air Force Base, Ohio*

and

BRIAN S. ZAFF

*Logicon Technical Services Inc., Dayton, Ohio*

The ultimate concern of cognitive engineering is how complex sociotechnical systems might be designed so that humans can work within them and control them safely and effectively. Because of this, large amounts of observational data analysis and knowledge elicitation are incorporated in cognitive engineering. At many points, these two methodologies coalesce. In this paper, we describe two complementary cognitive engineering software tools—MacSHAPA and COGENT—that are being developed alongside each other. MacSHAPA is designed for observational data analysis, and COGENT is designed for knowledge elicitation and cognitive engineering, but both support requirements gathering. We first outline current trends in cognitive engineering that have given rise to the need for tools like MacSHAPA and COGENT. We then describe the two tools in more detail, and point to their similarities and differences. Finally, we show how the two tools are complementary, and how they can be used together in engineering psychology research.

Cognitive engineering is a highly interdisciplinary field that concerns the design of complex human-machine systems in which human cognitive activities are well supported (Rasmussen, Pejtersen, & Goodstein, in press; Woods & Roth, 1988). Disciplines contributing to cognitive engineering are engineering psychology, systems engineering, software engineering, control engineering, and artificial intelligence, plus various aspects of the social and decision sciences. As a branch of psychology,

engineering psychology involves the discovery and articulation of the “first principles” of how people interact with engineered systems—vehicles, organizational information systems, industrial processes, communications systems, and so on.

As a discipline contributing to cognitive engineering, engineering psychology has traditionally focused on understanding the role of the individual human operator in complex work environments, as well as on designing for that role. Understanding the individual is still very important, but with the advent of sophisticated communications technology in the workplace, researchers are increasingly focusing on the entire work context, which consists of multiple workers, multiple tasks, and varied social and motivational pressures (Rasmussen et al., in press). There is an increasing need for engineering psychologists to describe how activities and responsibilities are shared, and how information, expertise, priorities, and meanings are conveyed within a group. In addition, there is a growing appreciation of the influence that the current context or situation has on people’s behavior (Suchman, 1987).

Thus, the type of research that helps cognitive engineers understand and design for today’s work environments tends not to be experimental, but instead tends to rely upon systematic observation and careful elicitation of knowledge. Moreover, the types of data collected are very broad (Olson, Olson, & Kraut, 1992), ranging from behavioral (operator actions, eye movements) to cognitive (verbali-

---

COGENT (Cognitive Engineering Network Technologies) represents confluences from two interrelated streams of research: (1) the Advanced Knowledge And Design Acquisition Methodology (AKADAM; see McNeese, Zaff, Citera, Brown, & Whitaker, in press; McNeese et al., 1990; Snyder, McNeese, & Zaff, 1991; Zaff, McNeese, & Snyder, 1993) and (2) Tools for Automated Knowledge Engineering (TAKE; see Gomes, Lind, & Snyder, 1993). Both are ongoing projects at the Armstrong Laboratory/Human Engineering Division, Wright-Patterson AFB. Further developments and research in TAKE may be obtained by contacting Stephanie Lind, AL/CFHI, Wright-Patterson AFB, OH. COGENT includes the total suite of tools under development in AKADAM and TAKE, spanning across the last 4 years. MacSHAPA is being developed at the Engineering Psychology Research Laboratory, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, with support from NASA-Ames Research Center, Wright-Patterson AFB, through Logicon Technical Services Inc., the Aeronautical Research Laboratory (DSTO, Australia), and the University of Illinois Research Board. Address correspondence and inquiries about MacSHAPA’s status and availability to P. Sanderson, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, 1206 W. Green St., Urbana, IL (e-mail: psanders@psych.uiuc.edu).

zations, concept clusters), organizational (role structures, use of social cues), and cultural (work customs). The two software tools introduced in this paper were designed to integrate these data and help the engineering psychologist and cognitive engineer draw inferences from them.

In the following text we discuss cognitive engineering as a user-centered, participatory approach to design, then describe the two software tools. First, concept-mapping approaches to cognitive engineering are outlined, and the COGENT software is introduced. Second, we outline how MacSHAPA supports observational data analysis. In the final section we highlight the complementary relationship between COGENT and MacSHAPA and discuss how this might be exploited.

### A COGNITIVE ENGINEERING FRAMEWORK

To successfully design a complex sociotechnical system, a cognitive engineer must understand the cognitive, social, organizational, and cultural forces that play out when the system is in operation. Such forces become apparent when humans talk, act, and interact with each other and with technology in real-world work contexts, whether in the normal course of events or in response to emergencies. To fully understand the complexities of such interactions, cognitive engineers must involve experts and system users directly in the process of participatory design. The expert's knowledge of his/her work is the basis for beginning the cognitive engineering process; this is what is meant by a "user-centered" approach. The design process is also "participatory"—domain practitioners become involved in the specification and design of the tools that they will eventually use (Woods, 1987). Thus, cognitive engineering can be described as an ecologically based activity (see Zaff, McNeese, Brown, Citera, & Selvaraj, 1993, for further details).

When cognitive engineers observe experts at work and interact with them to elicit their knowledge and their needs, there are four interrelated activities that need to occur: (1) generation, (2) preservation, (3) exploration, and (4) interpretation. *Generation* means that knowledge must be brought forth into some explicit form; *preservation*—that this knowledge must be recorded or stored so that it is available for examination; *exploration*—that investigators must be able to manipulate and query the knowledge; and *interpretation*—that the implications of this knowledge for design or redesign must be inferred. Each activity is iterative, using as well as producing external knowledge records. External knowledge records are the forms that knowledge can take as it is externalized, such as concepts that are written and linked on a white board, video segments preserved on tape, designs explored through protocol analysis, or comments interpreted through classification schemes. Other examples of knowledge records may be observation based or field based, such as verbal transcriptions, operator actions, eye movements, cultural artifacts, documented cases, or physiological data. In some instances, these records may be sequential streams of human behavior, and in others they may be nonlinear

representations of human thought. Regardless of how external knowledge records are represented, the generate-preserve-explore-interpret ("g-p-e-i") iteration is fundamental to cognitive engineering practice and needs to be supported with appropriate methodological tools.

Although the use of external knowledge records is not new in corroborating human understanding and assisting in the design of complex systems, many of these records have only been captured in isolation or without the full benefit of the g-p-e-i iteration. Thus, cognitive engineers need to be able to integrate different external knowledge records and to use different cognitive engineering techniques on these records with the goal of converging on sound recommendations for design. The methods and tools described in the present paper are presented with this objective in mind.

### SOFTWARE ENVIRONMENTS FOR COGNITIVE ENGINEERING

In the following text, we describe COGENT and MacSHAPA in detail. Briefly, both tools support the identification of specifications and requirements for design. COGENT is a set of methods and software tools that allows designers, software engineers, users, and others to collaborate on the identification of needs, constraints, options, important questions, and criteria that need to be considered in design. MacSHAPA is a software tool for observational data analysis that aids the systematic review and interpretation of recordings that have been taken of people at work in either existing or prototype systems. Both have arisen out of pressing practical needs, so some of this background will be provided where needed.

#### COGENT: Tools and Procedures for Concept Mapping

COGENT (Cognitive Engineering Network Technologies) is a knowledge-acquisition technique that is based on theories of generative learning and user-centered design. Traditional knowledge-acquisition tools tend to be structured around a knowledge engineer's needs rather than around the needs of users and domain experts<sup>1</sup> (Zaff, McNeese, & Snyder, 1993). However, COGENT uses a participant-centered technique called *concept mapping* (K. M. Fisher, Faletti, & Quinn, 1990; McFarren, 1987; Novak & Gowin, 1984). In this approach, a knowledge engineer helps domain experts express their knowledge as a concept map—a network of nodes (concepts) and labeled arcs (relations between concepts). We have found that concept mapping is an effective way of eliciting knowledge from experts, either individually or in groups, so that this information can be used in system design (McNeese, Zaff, Brown, Citera, & Wellens, 1992; Zaff, Hughes, McNeese, Brown, & Citera, 1993).

COGENT originated within the AKADAM (McNeese et al., 1990; Zaff, McNeese, & Snyder, 1993) and TAKE (Gomes, Lind, & Snyder, 1993) programs at the Human Engineering Division of the Armstrong Laboratory at Wright-Patterson Air Force Base; it refers to the collection

of tools, methods, and software developed to elicit experts' knowledge. COGENT software is Macintosh based, and it has supported participatory approaches to various real-world problems (for an overview see McNeese, Zaff, Citera, Brown, & Whitaker, in press; Young & McNeese, 1993). The COGENT procedure, with associated supporting functions, is shown in Figure 1. In the following we describe how COGENT works, using the g-p-e-i framework introduced above.

**Generate.** COGENT emphasizes "generative learning," in which knowledge records are created when a knowledge elicitor interacts with domain experts to elicit their knowledge of the objects, relationships, constraints, properties, and procedures of a system. In this process, the domain expert plays the role of mentor, and the knowledge elicitor plays the role of apprentice. Working together, they establish an understanding that the knowledge elicitor, as a domain apprentice, could not accomplish alone. This cooperative apprenticeship (Collins, Brown, & Newman, 1989) works through the interplay of reciprocal zones of proximal development (Vygotsky, 1978). A generative learning environment is established, in which the expert mentor may learn more about his/her own knowledge through the process of explaining it to the elicitor apprentice, and as the elicitor apprentice attains new levels of understanding, he/she will generate new questions for the expert mentor. The COGENT procedure

"bootstraps" this process by offering a general node-link architecture for the representation of knowledge and relations, allowing the elicitor apprentice and expert mentor to build concept maps.

**Preserve.** The "preserve" activity within COGENT refers to the social construction (Bereiter & Scardamalia, 1989) of a permanent knowledge record: the concept map. The expert mentor and elicitor apprentice work together for an hour or more to build a map of the expert's knowledge, typically on a white board. The expert learns how to make his/her thoughts explicit through concept mapping, and expresses more knowledge than would be possible alone. During the session, the concept map literally develops in front of the eyes of the expert and the knowledge elicitor. The elicitor "facilitates" this process, and ensures that the map appearing on the white board accurately represents the expert's knowledge, but the expert is nonetheless a full participant in the physical creation of the map. At the same time, the map may be entered into a computer.<sup>2</sup> The COGENT software provides a flexible, direct manipulation interface for creating concept maps, allowing concepts and links to be labeled with text or graphics and to be easily edited and rearranged. If the concept-mapping session is videotaped, it can subsequently be reviewed with the help of MacSHAPA.

**Explore.** At the end of the first session, the concept map in the computer may be printed out for subsequent

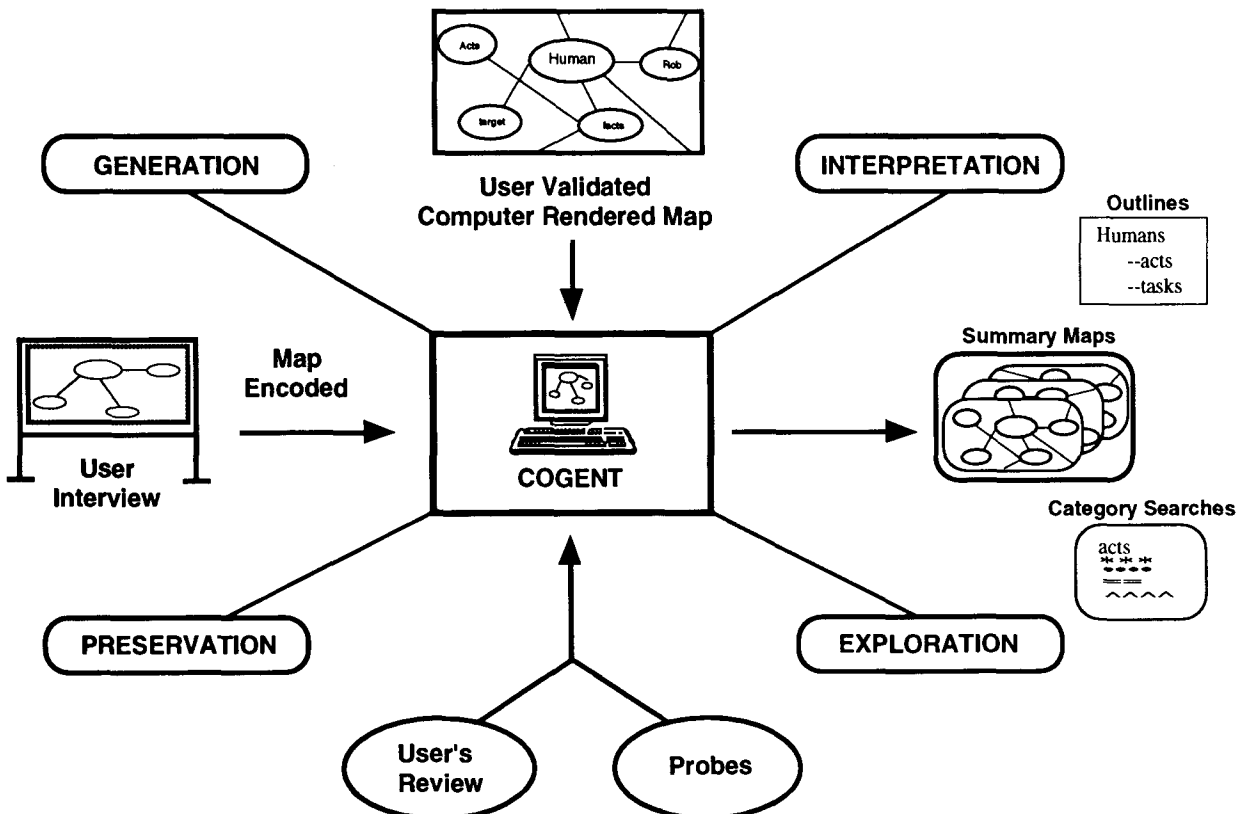


Figure 1. COGENT procedure.

exploration and review by the expert(s) or knowledge elicitor(s). As the experts explore the map they have produced, they can make additions, deletions, or changes to any part of it. Moreover, because each concept map is saved as a series of triplets ( $\langle$ concept $\rangle$ ,  $\langle$ relation $\rangle$ ,  $\langle$ concept $\rangle$ ) in a relational database format, it is possible to combine maps from different experts to form a summary map for further exploration. COGENT thus offers quick and powerful ways to manipulate and query data either from a single session or from many sessions collapsed together.

**Interpret.** Finally, an individual map or a collapsed set of maps may be chosen for interpretation by COGENT. Keywords for certain types of argumentation (e.g., *because* or *reason*) are resident within the software, and the analyst can use them to search the whole data set for the contexts in which they occur. Researchers can either extend this resident classification scheme or create new ones. For example, if we wished to analyze a set of maps to find concepts related to human factors problems, we would establish the category keyword HUMAN FACTORS. We would then identify all keywords associated with this category to access the appropriate triplets of the concept map. So, for HUMAN FACTORS, keywords might be *human factors*, *ergonomics*, *human-machine systems*, and/or

*cognitive performance*. The software provides feedback about the number of concepts elicited for each category that is searched (see Figure 2). The number of elicited concepts can also be presented as a percentage of the total number of concepts in the map. Moreover, concepts matched in the HUMAN FACTORS search can be highlighted in the map in a certain color, such as blue, whereas concepts associated with the category VISION might be highlighted in green.

Concept maps can be reexpressed as text-based hierarchical outlines that can stimulate further thought and help the expert or the knowledge elicitor to generate new concepts. New text inserted into the outline results in a corresponding change in the graphical concept map, and vice versa.

When new systems are in development, concept mapping is a powerful technique for bootstrapping the thinking of experts, designers, and software engineers. Concept mapping is a reflective technique that helps people make their tacitly held knowledge and assumptions explicit and connect their point of view with others'. However, throughout the concept-mapping process, COGENT may interact with other cognitive engineering tools. For example, in order to understand how a system works, the cognitive engineer often must step into the real situation and make observations. In the following, we concentrate

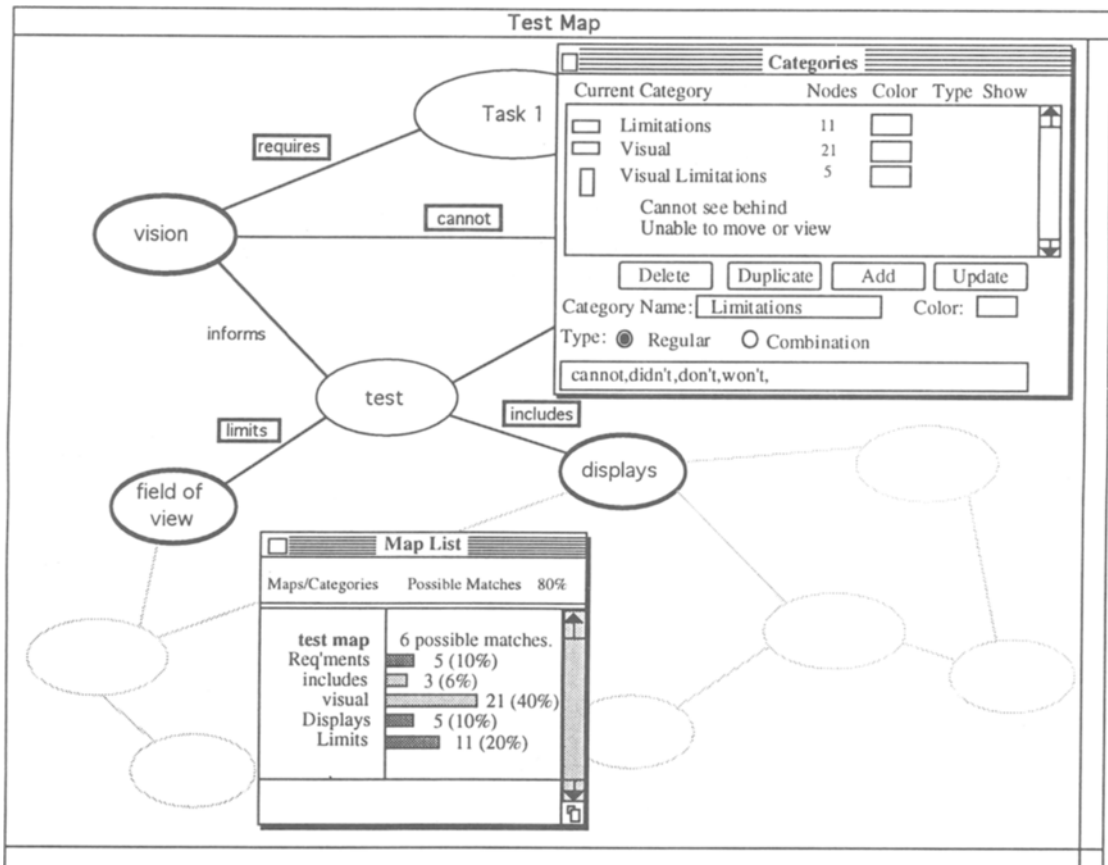


Figure 2. Using COGENT functions to search a concept map.

on how this is achieved with MacSHAPA. Finally, we outline how a fruitful symbiotic relationship can develop between the two tools.

**MacSHAPA: Software Environment for Observational Data Analysis**

Observation is used in several ways in cognitive engineering: (1) to analyze how existing systems work, (2) to identify problems or strengths of existing systems, (3) to examine sequential and temporal dependencies in user-system interaction, (4) to test prototype designs, and (5) to evaluate new designs once they are in production (Woods, 1993). Cognitive engineers rely on various types of observational records, which include video recordings, eye-movement data, system state information, data logs of human activity, transcripts of verbalizations, and field notes. Because observational data are rich and heterogeneous, it is often difficult to know how to analyze them. Moreover, analysis always seems prohibitively time-consuming. For discussions of these issues, see MacKay (1989), Woods (1993), Sanderson and C. Fisher (in press), and Jordan and Henderson (in press).

MacSHAPA is a software tool designed to help researchers overcome some of these problems (Sanderson, 1993; Sanderson, James, Watanabe, & Holden, 1990;

Sanderson et al., in press). MacSHAPA allows researchers to (1) enter data and view it in different ways, (2) edit, manipulate, and query data, and (3) analyze data with various statistical reports. Although MacSHAPA supports the “g-p-e-i” functions outlined previously, it does so in a way that is rather different from COGENT, so these functions will be examined only at the end of this section.

With MacSHAPA, all the data in a session can be collected into one database and displayed in a spreadsheet-like format (see center of Figure 3). Time runs downward, and the columns, or *spreadsheet variables*, represent different streams of data. The rectangles inside each spreadsheet variable, or *spreadsheet cells*, hold atoms of information that can be joined together or split even more finely. Very often, the onset and offset times of events in different spreadsheet variables are different; MacSHAPA handles this by using a weak temporal ordering logic (see Sanderson et al., in press, for more details). The data can also be displayed in a “Timeline” format, in which strong temporal ordering is preserved (see lower right section of Figure 3).

Figure 4 is a diagram of MacSHAPA’s capabilities. At the center are spreadsheet cells, contained within spreadsheet variables, and around the perimeter are MacSHAPA’s functions. *Video* refers to the fact that MacSHAPA re-

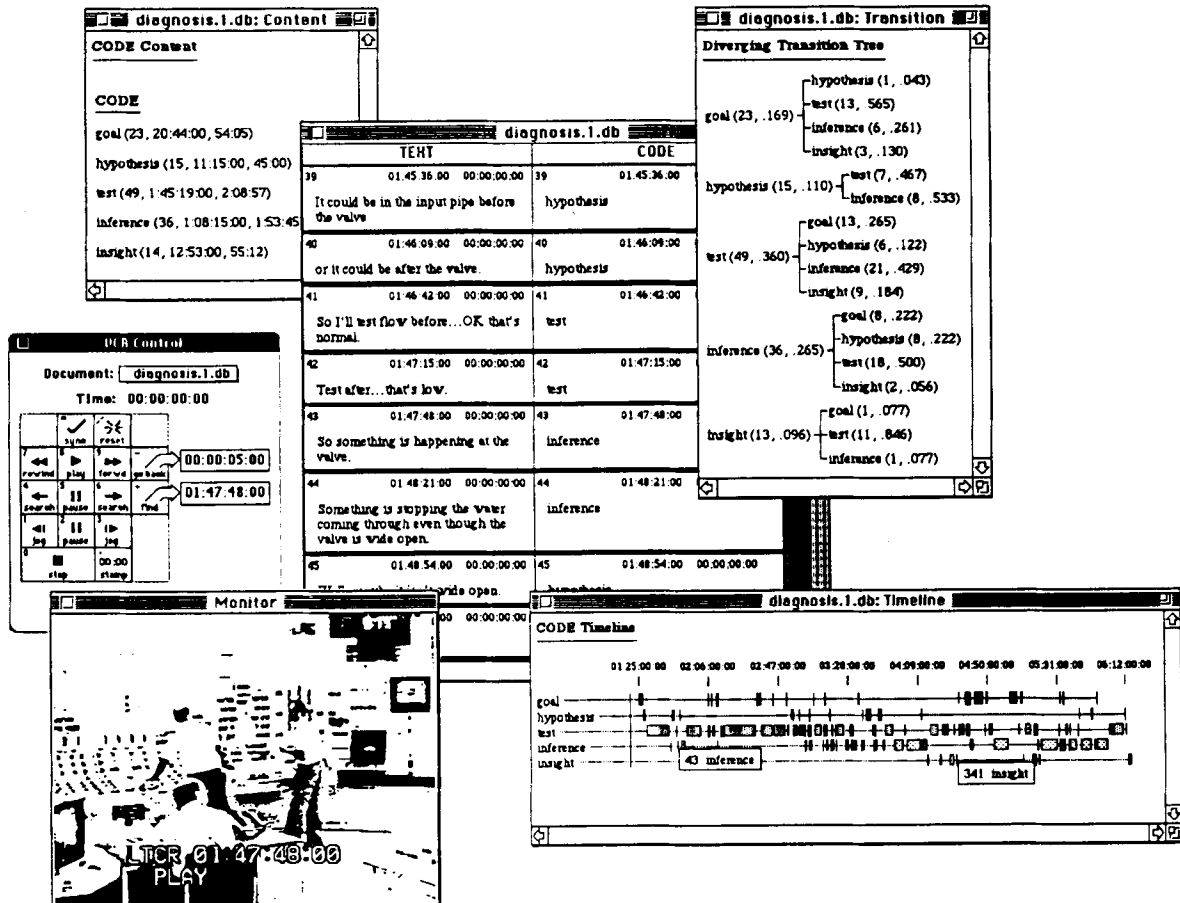


Figure 3. MacSHAPA: windows showing different expressions of data.

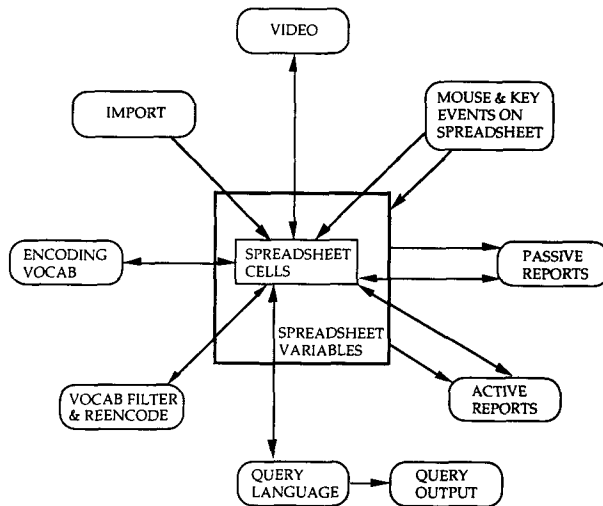


Figure 4. Structure of MacSHAPA.

motely controls all the basic transport functions of a VCR (see "VCR Control," left side of Figure 3). The video image itself can be directly displayed on the computer screen (as in the lower left section of Figure 3) or sent to an external monitor. Researchers can click on a cell in the spreadsheet or Timeline display and ask MacSHAPA to find and play the corresponding part of a time-coded videotape. Moreover, SMPTE timecodes can be passed from the VCR to MacSHAPA. This means that, while playing a video, researchers can (1) create timestamped cells in real time and type comments into them, or (2) ask MacSHAPA to highlight all the cells whose timestamps correspond to the part of the video currently being viewed.

*Mouse & Key Events on Spreadsheet* and *Import* indicate that data can be directly entered into MacSHAPA or imported from other applications, such as word processors or spreadsheet programs. Researchers can type comments or codes into the spreadsheet; they can resize, move or hide spreadsheet columns, and cut, copy, and paste cells from one place to another.

*Encoding Vocab* indicates that researchers can establish customized coding schemes if they wish, although this is not necessary when using MacSHAPA. Figure 3 shows a simple keyword coding scheme (*goal*, *hypothesis*, etc.), but more complex multidimensional schemes can be created. *Vocab Filter & Reencode* allows the researcher to manipulate and edit any encoding performed with customized coding schemes. The *Vocab Filter* allows the researcher to select which part of the data should go forward to analysis, and *Reencode* allows codes to be replaced with others in one-to-one or many-to-one mappings.

The two right-hand nodes on the perimeter of Figure 4 refer to ways of reexpressing, analyzing, and reporting the data. *Passive Reports* includes the Timeline analysis (previously discussed) and Content analysis (see upper left section of Figure 3), which provides basic statistics on the frequency and duration of coded events. *Active Reports* includes Transition analyses with some Markov statistics (see upper right side of Figure 3), Cycles reports

(C. Fisher, 1991; James & Sanderson, 1991), Lag Sequential Analysis (Faraone & Dorfman, 1987), and a set of Comparisons reports. The Comparisons report provides different ways of aligning two timestamped sequences and of determining the degree of match between them by using various algorithms.

Finally, MacSHAPA's Prolog-based *Query Language* can be used to enter new data, manipulate the database, or ask further questions of the data (Charniak & McDermott, 1986). The query language includes a wide array of relational, arithmetic, string, and temporal operators for specifying data patterns. MacSHAPA's *Query Output* actions include selecting and inserting cells, as well as modifying, counting, summing, or printing cell contents.

**Generate, preserve, explore, and interpret.** MacSHAPA can be used for exploratory or confirmatory data analysis, but most relevant for cognitive engineering design is its exploratory use, helping to generate hypotheses as much as to test them. MacSHAPA's most fundamental uses are preserving observational data—and thereby some of the original context—and integrating different data sources. This type of preservation allows the data to be explored more effectively. When data are visualized and reexperienced in different ways, they can also be more reliably and validly interpreted. Generative learning typically arises from a researcher's intense familiarity with a data set, which comes from exploring it, talking about it with colleagues and participants, and attempting different interpretations of it to see which is best supported by the data.

## COMPLEMENTARITY OF COGENT AND MacSHAPA

As outlined earlier, both COGENT and MacSHAPA support requirements gathering for design—MacSHAPA by supporting observational data analysis, and COGENT by supporting participatory knowledge elicitation. In this section we outline the principal similarities and differences between COGENT and MacSHAPA as software tools. We then point to ways in which the tools are complementary, and sketch out the type of cognitive engineering research that can be performed with the two tools working in concert.

### Similarities

COGENT and MacSHAPA were both designed to help researchers and designers make inferences about the needs of new systems, and they are therefore useful for needs gathering, requirements analysis and system specification. Both are based on naturalistic methods; the researcher lets what is happening unfold with minimal direction or interruption, whether it be the concept-mapping process or the normal unfolding of work activities. Both tools promote a flexible yet systematic approach to novel research or design problems by allowing researchers to explore relevant data through visualizations, reexpressions, and querying. The two tools integrate data that would otherwise be handled in a piecemeal fashion: COGENT coordinates results from multiple maps, and MacSHAPA allows multiple data streams to be viewed alongside each other and

multiple interpretations to be sustained. Both tools support qualitative analyses, but they also offer quantitative tools when precision is necessary. Finally, both tools support participant involvement in analysis. With COGENT, domain experts and users become mentors, and with MacSHAPA, the people being observed can take part in the review and interpretation of their own data.

### Differences

COGENT and MacSHAPA are used at different points in cognitive engineering and support qualitatively different methods of structuring data. COGENT is usually used during meetings, interviews, and consultations, and the concept-mapping procedure is designed to help participants consciously formalize their knowledge. In contrast, MacSHAPA usually focuses on real-time task performance; cognitive engineers form conclusions about system needs when participants are simply acting, rather than reflecting on their actions. Through its use of concept maps, COGENT produces nonlinear representations of knowledge, whereas MacSHAPA preserves and analyzes temporal aspects of performance that are more linear in nature. Finally, COGENT allows queries and integrates findings over multiple maps, whereas MacSHAPA supports the detailed analysis of one document (or observational session) at a time.

### Complementarity

Because of the above similarities and differences, COGENT and MacSHAPA can be used in a complementary fashion. In cognitive engineering—whether focused on research or system design—it is important to continually review the raw data upon which inferences are formed. Working together, COGENT and MacSHAPA can help to achieve this. They can be used either at the same time (e.g., using Multifinder in the Macintosh environment) or sequentially, with the data from one being passed to the other. Complementary use of the tools means that data can be accessed either as events in time (MacSHAPA) or instances of concepts (COGENT). For example, both tools can be used for observational data analysis: MacSHAPA helps the researcher to understand the temporal relations between events, and COGENT allows the researcher to think about nontemporal themes, such as the structure of argumentation or the logical relations in some domain of discourse. COGENT can also act as a “sketchpad” for understanding the high-level sequential or conceptual structure of the data observed.

MacSHAPA can bring the use of videotape to concept mapping. A concept map may be developed from videotaped data that record either the current system being used, or an earlier concept mapping session about the proposed system. If the concept map stores indices associated with the raw video data, such as timecodes or keywords, then these indices can be passed to MacSHAPA, which can handle search and play functions.

It is also possible to share data across COGENT and MacSHAPA. As outlined above, the basic data structure for concept maps is  $\langle \text{concept} \rangle$ ,  $\langle \text{link} \rangle$ ,  $\langle \text{concept} \rangle$ , which

is a “triple” in COGENT’s world, and a certain kind of “matrix” in MacSHAPA’s world. Observational data that are coded as a three-argument matrix in MacSHAPA can be exported to COGENT as triples and represented as a map. Alternatively, concept-mapping data developed within COGENT can be passed to MacSHAPA for further analyses, such as content analysis and certain types of queries. Most important, MacSHAPA makes it possible to observe the concept-mapping sessions themselves. This is the focus of the following section.

### “Symbiotic” Research Prospects

In this final section we sketch scenarios in which MacSHAPA and COGENT can work together to investigate concept mapping as a research topic in its own right. Zaff, McNeese, and Snyder (1993) and McNeese et al. (in press) have shown that concept mapping can be an effective knowledge-acquisition tool for cognitive engineering problems, but relatively little is known about why it works. What factors make a concept-mapping team work well? Is it the composition of the group, the work tactics chosen, or the presence of certain group dynamics? How does prior exposure to concept mapping in general, and exposure to preexisting maps in particular, affect the effectiveness of a design team? If different teams are introduced to the same concept map, but are given different entry points into that map, does this systematically affect their final maps? How are inferences for design drawn from maps?

These questions could be answered with appropriate experimental contrasts, by observing concept-mapping sessions, and by analyzing the results with MacSHAPA. In addition, the sequential order in which individuals or teams generate concepts may suggest subjective associative paths. The temporal clustering of concept generation may suggest how strong these paths are; strongly associated concepts may be produced in rapid succession. If concepts are produced in approximately the same order across individuals, and some concepts are always more tightly coupled in time, then this suggests a consensus about associative pathways. MacSHAPA is geared toward finding such temporal patterns: the Cycles report helps the cognitive engineer identify characteristic subsequences of activity, and the Comparisons report gives a figure of merit on the similarities of the selected subsequences. MacSHAPA’s Query language can also help uncover characteristic subsequences and can report the span of time within which such subsequences are produced.

## CONCLUSIONS

In conclusion, COGENT and MacSHAPA both help cognitive engineers determine how sociotechnical systems should be designed. COGENT helps to elicit and represent nonlinear conceptual structures, whereas MacSHAPA helps to capture, review, and analyze observational and systems data. The two tools balance hypothesis-generation and hypothesis-testing needs, bring rigor to open-ended research and design questions, and can be used in parallel.

Although we have concentrated on cognitive engineering in the present article, we believe that COGENT and MacSHAPA can be useful in many other areas of the behavioral, social, and decision sciences.

## REFERENCES

- BEREITER, C., & SCARDAMALIA, M. (1989). Intentional learning as a goal of instruction. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 361-392). Hillsdale, NJ: Erlbaum.
- CHARNIAK, E., & MCDERMOTT, D. (1986). *Introduction to artificial intelligence*. Reading, MA: Addison-Wesley.
- COLLINS, A., BROWN, J. S., & NEWMAN, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Roger Glaser* (pp. 453-494). Hillsdale, NJ: Erlbaum.
- FARAONE, S. V., & DORFMAN, D. D. (1987). Lag sequential analysis: Robust statistical methods. *Psychological Bulletin*, **101**, 312-323.
- FISHER, C. (1991). *Protocol analyst's workbench: Design and evaluation of computer-aided protocol analysis*. Unpublished doctoral dissertation, Carnegie-Mellon University, Pittsburgh, PA.
- FISHER, K. M., FALETTI, J., & QUINN, C. (1990). *Exploring cognitive structure with semantic networks*. Available from K. M. Fisher, CRMSE, 6475 Alvarado Road, San Diego State University, San Diego, CA 92182.
- GOMES, M. E., LIND, S., & SNYDER, D. E. (1993). A human factors evaluation using tools for automated knowledge engineering. In *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON)* (pp. 661-664). Dayton, OH: IEEE, Aerospace & Electronic Systems Society.
- JAMES, J. M., & SANDERSON, P. M. (1991). Heuristic and statistical support for protocol analysis with SHAPA Version 2.01. *Behavior Research Methods, Instruments, & Computers*, **23**, 449-460.
- JORDAN, B., & HENDERSON, A. (in press). Interaction analysis: Foundations and practice. *Journal of the Learning Sciences*.
- MACKEY, W. E. (1989). EVA: An experimental video annotator for symbolic analysis of video data. *SIGCHI Bulletin*, **21**, 68-71.
- McFARREN, M. R. (1987). *Using concept mapping to define problems and identify key kernels during the development of a decision support system*. Master's thesis (AFIT/GST/ENS/87J-12), Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio.
- MCNEESE, M. D., ZAFF, B. S., BROWN, C. E., CITERA, M., & WELLENS, A. R. (1992). The role of a group-centered approach in the development of computer-supported collaborative design technologies. In *Proceedings of the 36th Annual Meeting of the Human Factors Society*, **2**, 867-871. Santa Monica, CA: Human Factors Society.
- MCNEESE, M. D., ZAFF, B. S., CITERA, M., BROWN, C. E., & WHITAKER, R. (in press). AKADAM: Eliciting user knowledge to support participatory ergonomics. *International Journal of Industrial Ergonomics*.
- MCNEESE, M. D., ZAFF, B. S., PEIO, K. J., SNYDER, D. E., DUNCAN, J. C., & McFARREN, M. R. (1990). *An advanced knowledge and design acquisition methodology: Application for the pilot's associate* (Report No. AAMRL-TR-90-060). Wright-Patterson Air Force Base, OH: Armstrong Aerospace Medical Research Laboratory.
- NOVAK, J. D., & GOWIN, D. B. (1984). *Learning how to learn*. Cambridge: Cambridge University Press.
- OLSON, G. M., OLSON, J. S., & KRAUT, R. E. (1992). Introduction to this special issue on computer-supported cooperative work. *Human-Computer Interaction*, **7**, 251-256.
- RASMUSSEN, J., PEJTERSEN, A. M., & GOODSTEIN, L. P. (in press). *Cognitive engineering: Concepts and applications*. New York: Wiley.
- SANDERSON, P. M. (1993). Designing for simplicity of inference in observational studies of process control: ESDA and MacSHAPA. In E. Hollnagel & M. Lind (Eds.), *Proceedings of the Fourth Cognitive Science Approaches to Process Control Conference (CSAPC'93)* (pp. 20-47). Birkerød, Denmark: CRI A/S.
- SANDERSON, P. M., & FISHER, C. (in press). Exploratory sequential data analysis: Foundations. *Human-Computer Interaction*.
- SANDERSON, P. M., JAMES, J. M., WATANABE, L. M., & HOLDEN, J. (1990). Human operator behavior in complex worlds: Rendering sequential records analytically tractable. In P. C. Caccibue & G. Mancini (Eds.), *Proceedings of the Ninth European Annual Conference on Human Decision Making and Manual Control* (pp. 123-132). Ispra, Italy: CRC JRC.
- SANDERSON, P. M., SCOTT, J. J. P., JOHNSTON, T., MAINZER, J., WATANABE, L. M., & JAMES, J. M. (in press). MacSHAPA and the enterprise of exploratory sequential data analysis. *International Journal of Human-Computer Studies*.
- SNYDER, D. E., MCNEESE, M. D., & ZAFF, B. S. (1991). Identifying design requirements using integrated analysis structures. In *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON)* (pp. 786-791). Dayton, OH: IEEE, Aerospace & Electronic Systems Society.
- SUCHMAN, L. (1987). *Plans and situated actions: The problem of human-machine communication*. New York: Cambridge University Press.
- VYGOTSKY, L. S. (1978). *Mind in society*. Cambridge, MA: Harvard University Press.
- WOODS, D. D. (1987). Commentary: Cognitive engineering in complex and dynamic worlds. *International Journal of Man-Machine Studies*, **27**, 571-585.
- WOODS, D. D. (1993). Process tracing methods for the study of cognition outside of the experimental psychology laboratory. In G. Klein, J. Orasanu, R. Calderwood, & C. E. Zsombok (Eds.), *Decision making in action: Models and methods*. Norwood, NJ: Ablex.
- WOODS, D. D., & ROTH, E. M. (1988). Cognitive systems engineering. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 3-43). Amsterdam: Elsevier.
- YOUNG, M. F., & MCNEESE, M. D. (1993). A situated cognition approach to problem solving with implications for computer-based learning and assessment. In G. Salvendy & M. Smith (Eds.), *Human-computer interaction: Software and hardware interfaces* (pp. 825-830). Amsterdam: Elsevier.
- ZAFF, B. S., HUGHES, E. R., MCNEESE, M. D., BROWN, C. E., & CITERA, M. (1993). Diagnosing macro-ergonomic problems: A case study in the use of concept mapping for TQM initiatives. In *Proceedings of the 37th Annual Meeting of the Human Factors and Ergonomics Society* (pp. 873-877). Santa Monica, CA: Human Factors Society.
- ZAFF, B. S., MCNEESE, M. D., BROWN, C. E., CITERA, M., & SELVARAJ, J. A. (1993). Empowering designers with user-centered knowledge: Issues, methods, and solutions. In *Proceedings of the Interface 93* (pp. 231-235). Santa Monica, CA: Human Factors Society.
- ZAFF, B. S., MCNEESE, M. D., & SNYDER, D. E. (1993). Capturing multiple perspectives: A user-centered approach to knowledge acquisition. *Knowledge Acquisition*, **5**, 79-116.

## NOTES

1. The term *expert* is generically used to refer to various users/stakeholders, subject-matter experts, and domain specialists participating in engineering-design and/or work-management projects.
2. Currently, a single analyst/researcher can easily encode the map as it develops on the white board by using the COGENT software. However, the full embodiment of COGENT would allow every member of the cognitive engineering team to enter concepts through pen-based graphic tablets and see the emerging map on an electronic white board in either collocated or distributed computing environments. This is a goal in line with putting tools in the users' hands, and one that is currently under development at the Armstrong Laboratory/Human Engineering Division in the Collaborative Design Technology Program.