COGNITION SUPPORT TOOLS FOR KNOWLEDGE ACQUISITION

J. Brian Woodward & Mildred L G Shaw

Knowledge Science Institute
University of Calgary
Calgary, Alberta, Canada T2N 1N4
woodward@cpsc.ucalgary.ca, mildred@cpsc.ucalgary.ca

Abstract: The role of the knowledge engineer is cognitively demanding in that the knowledge engineer is expected to comprehend the epistemological structure of a domain sufficiently to build a model of that domain. This paper develops a theory of cognitive modeling for knowledge engineering, and discusses the cognitive sites in knowledge engineering. It describes a methodology of cognitive mapping based on this theory, and a knowledge acquisition tool designed to support the methodology. Case Map is a tool which identifies and supports specific cognitive processes in knowledge acquisition, and manages the knowledge structures being developed by the knowledge engineers. Case Map comprises two main components. The first takes a list of questions provided by the knowledge engineer and inserts them in a display sequence shown to the expert. This component stores the information in list form to be used by the mapping component. The second component, ConMap, is a concept mapping tool which graphs the resulting questions and their responses in an interactive graphing environment. The two components are integrated so that any changes in one component is automatically displayed in the other.

1 INTRODUCTION

The current work is motivated by the problems observed by Woodward (1992a) who studied a knowledge engineer re-engineering Oncocin (Musen, Fagan, Combs and Shortliffe, 1987) into K-Oncocin (Linster and Musen, 1992) using the KADS methodology. The knowledge engineer had difficulty mapping the elements of the problem onto the predefined conceptual tools. A number of knowledge engineering issues became apparent from this study:

- The knowledge engineer had to invent his own intermediate steps in order to make a successful mapping and even then the final mapping required some redefinition of the predefined concepts.
- The knowledge engineer often made choices for abstractions but was uncertain of his consistency or correctness.
- Cognitive biases were evident throughout the knowledge engineering process.

The objectives of the research reported in this paper have been:

1) to develop a theoretical framework for the cognitive processes involved in complex knowledge engineering tasks;
2) to use this framework to develop a methodology for managing these processes; and
3) to develop tools that support the methodology.

The tools described are those for general concept maps specialized to different cognitive activities by the use of node types deriving from the methodology. The node types are developed as part of the knowledge engineering process and posed as questions to provide a well-defined elicitation methodology specific to a particular knowledge engineering activity. The result is a structured concept map with heterogeneous but linked regions each corresponding to different cognitive activities involved in knowledge engineering. Links may also be maintained to knowledge sources and to computational structures in order to make a complete record of the knowledge engineering process.

This paper examines the background theory of cognitive modeling for knowledge engineering, in particular looking at the cognitive sites in knowledge engineering. It then goes on to describe the generally applicable but structured informal methodology of cognitive mapping which supports this theory. One example of such a tool, Case Map, is described which was designed specifically with cognitive processes in mind, and an example of its use in knowledge acquisition is given.

2 COGNITIVE PROCESSES IN KNOWLEDGE ENGINEERING

The role of the knowledge engineer is cognitively demanding in that the knowledge engineer is expected to enter into the epistemological structure of a domain far enough to build a model of that domain. Even if the domain is restricted in scope, the knowledge engineer must become familiar with the problem-solving structures in the domain and the underlying assumptions in the domain of interest. When a knowledge engineer enters a new, complex, epistemological domain, he/she uses a variety of cognitive processes first to understand and comprehend the basic structures and assumptions of the domain and second, to model this understanding. Woodward, Shaw and Gaines (1992) have identified the need to engineer cognitive science models and techniques into useful tasks, techniques, and procedures for knowledge engineering. It is these cognitive processes which require identification and description so that tools, techniques, and procedures may be developed as aids and supports to the knowledge engineer alone or in interaction with the expert.

The Bias Toward Complexity Reduction in Knowledge Engineering

Most of the research study of cognitive processes in knowledge engineering has focused on the expert; for example, research on bias in knowledge acquisition (Meyer and Booker, 1991). Shadbolt (1992) comments on the lack of attention placed on the cognitive issues of knowledge engineering especially after the work on expert bias has demonstrated severe difficulties in identifying and controlling for bias. Knowledge engineers, like experts and other people, are also subject to patterns of error even though the operations may be simple. Knowledge engineers deal with complexity and uncertainty in their tasks. The complexity inherent in a new domain results in biases to reduce complexity (reductive bias) and, indeed, the knowledge acquisition process itself is an exercise in reduction and is consequently biased in terms of selection and representation.

Pfeifer, Rothenfluh, Stolze and Steiner (1992) conclude that there are major gaps in our understanding when we try to re-construct the cognitive processes of knowledge engineers. They state the difficulty they had with understanding how a knowledge engineer used MOLE (Eshelman, 1988) in that the process of how the knowledge engineer mapped a diagnostic task onto the 'cover-and-differentiate' problem-solving method was unclear. The descriptions of knowledge engineering tools in the literature focus mostly on the tool ,with little description of the cognitive demands the tool places on the user or the cognitive context in which the tool is used.

This is not an uncommon occurrence in the field of knowledge acquisition. There has been great difficulty in sharing knowledge acquisition tools within the community, not because of a lack of motivation but due largely to the difficulty in using the tool in other than the 'home site'. The reasons for this may be:

- The knowledge acquisition tool is part of a larger, more complex set of cognitive processes. The tool developer has constructed other knowledge-support procedures as well as cognition-support procedures to complement the knowledge acquisition tool. These other procedures are seldom mentioned in the literature which describes the tool or tool use.
- Knowledge engineers try to bend or adjust the tools to meet their own needs or uses. Consequently, the tool(s) used by knowledge engineers in other than their 'home' site are often inadequate to meet local needs.

Much knowledge engineering activity reflects the transformation of information from one framework to another. Pfeiffer et al (1992) suggest the need to study this mapping process as one important line of research in knowledge acquisition. They suggest that the application of cognitive models are best assessed and applied by studying how experienced knowledge engineers apply a knowledge acquisition technique or framework to a particular problem. These authors state that the goal of this research is not to develop an automated system to replace it but to support the process with the appropriate computer tools.

Knowledge engineering also reflects a reductive quality focusing attention on elements of a problem or on a sequence of events which need to be modeled. Research on how medical students and interns come to develop expertise in a complex epistemological domain parallels the cognitive demands placed on the knowledge engineer when he/she enters a new domain. Evans (1989) stresses that addressing the complex domain of medical problem-solving reflects four critical issues:

"1) the relation between domain knowledge and problem-solving behavior;
2) the sources of bias and misconception in problem representation and decision-making;
3) the role of discourse in structuring the acquisition of data...., and
4) the identification of formal methods for the evaluation of performance" (p. 2).

These same issues are of importance to the knowledge engineer and to knowledge engineering activity. The point here is that the knowledge engineer is required to learn something about the domain in order to make knowledge engineering decisions. The cognitive demands placed on a knowledge engineer when he/she enters a new domain are similar to novices in that domain who are in the process of developing a greater understanding of the domain. The rest of this section presents research support for a number of cognitive processes involved when an individual is developing understanding in a complex domain.

Patel, Evans and Groen (1989) identified a number of cognitive processes used to incorporate models of biomedical knowledge in diagnosing clinical cases. These processes included comprehension of text-based propositions, construction of consistent and coherent explanations, and selection of relevant and irrelevant facts based on clear discrimination principles. These findings were based on novice-expert interactions and they represented points of distinction between novice and experts in the domain.

Complex concepts make unusual cognitive demands on the novice. Feltovich, Spiro and Coulson (1989) describe four main demands on:

1) Working memory because of the large number of nested or looped steps or goals, or because of the large number of variables to be considered, processed, and/or reconciled;
2) formal representation in the sense that the degree of abstraction necessary for understanding is often onerous (e.g.. representing the concept of rate or acceleration);
3) intuition or prior knowledge in the sense that new concepts can conflict with prior meanings and/or relationships among concepts; and
4) notions of regularity because many concepts can be ill-structured and highly variable in how they are used or they may be very strongly dependent on other concepts for their meaning.

These same demands are placed on the knowledge engineer. These cognitive demands affect knowledge engineering activity by leading to misconceptions which may be perpetuated into the final system if not 'caught' by the expert (assuming there exist specific procedures for the expert to use to 'catch' the misconception). Also, when complex conceptual domains are addressed, complexity reduction occurs. Many researchers have studied the processes whereby humans use particular procedures to reduce the complexity of conceptual information (Kahneman and Tversky, 1973; Kahneman, Slovic and Tversky, 1982; Coulson, Feltovich, and Spiro, 1986; Spiro, 1980).

Feltovich, Spiro and Coulson (1989) distinguish conceptual bias from general biases in judgment and decision-making discussed by Kahneman, Slovic and Tversky (1982). They define a family of 'reductive biases' which are used solely to reduce complexity rather than reflect judgment processes. Six reductive biases were identified. 'Static' bias occurs when a dynamic system is viewed as a static model. 'Step-wise' bias describes the situation where a continuous process is broken down into discrete, identifiable steps resulting in a loss of meaning. 'External agent' bias is the attribution of essential intrinsic characteristics of entities or processes to external influences. 'Prior analogy' bias occurs when new concepts are given meaning based on already held concepts or simple models within the domain or from other domains. 'Common connotation' bias occurs when technical terms are reduced in meaning based on the day-to-day use of the term or concept. Finally, 'restriction of scope' bias refers to the belief that general principles are taken to apply only under certain circumstances based on a repeated co-incident relationship.

Cognition-Assisting Devices

The field of cognitive science supports different theories of cognition and a wide range of contentious issues (Harman, 1990). However, the domain of knowledge engineering requires a more pragmatic view of the cognitive processes involved in building knowledge-based systems. For this paper, cognitive processes are defined as those basic psychological operations which appear during the completion of complex tasks in knowledge engineering. Of primary interest are the sequences of cognitive/ perceptual/ motor operations involved in meeting the demands of various tasks, collectively known as knowledge engineering. The following list is by no means exhaustive and the examples may not be homogeneous in complexity, but the list provides examples of such cognitive processes:

- interpreting
- searching
- distinguishing
- reminding
- indexing
- listing
- explaining
- demonstrating
- constructing
- selecting
- judging
- matching

Cognition-assisting devices are those tools, techniques, and procedures which either support, enhance, or replace cognitive activity. These provide a mechanism, based on an understanding of a defined cognitive activity, which in some form acts to assist or help in the completion of the activity while at the same time reducing the degrees of freedom of that activity. These cognition-activity devices may take many forms and many general tools already exist although not specifically for knowledge engineering:

- memory devices such as books, notes, etc.
- measurement instruments
- tables of arithmetic conversions
- dictionaries, thesauruses, maps
- word processor programs
- spreadsheet programs
- graphic display programs

Because the field reflects knowledge processes, the cognitive activities of interest are those processes which involve the representation, manipulation, and transformation of knowledge. When knowledge is first elicited, it is a model or re-representation, or transformation of the original activity. This process of transformation requires a number of steps: identification of wholes and components; selection of components; identification of final structures; alteration of components from one to another representation; verification. A transformation tool lessens or shifts the load of the knowledge engineer. The tool may take over one or more steps completely or provide means to reduce the complexity of the step to a series of small judgments and decisions rather than a complex, multi-step cognitive activity.

3 A THEORETICAL FRAMEWORK FOR COGNITIVE PROCESSES IN KNOWLEDGE ENGINEERING

What is needed to bring together the diversity of cognitive processes involved in knowledge engineering and the variety of cognition-assisting devices is a theoretical framework that encompasses the full scope of knowledge engineering activity and supports its management. This framework would help to solve the problem of the 'patchwork' nature (Shaw and Woodward, 1990) of the field of knowledge engineering. This section presents such a framework in terms of different sites of knowledge engineering activity, and different levels of cognitive processes involved.

Sites of Cognitive Activity in Knowledge Engineering

Figure 1 schematically depicts seven distinct sites (A to G) of cognitive activity in knowledge engineering.

a) The knowledge engineering context (A) is that site of cognitive activity which reflects the various goals, techniques, procedures, and tools which are common to all knowledge engineering activity. Whether the knowledge engineer (site C) uses these devices or not, they are available and may have influenced him/her in the past. This site becomes more important for identifying cognitive processes as more knowledge engineers are involved;

b) The domain context (B) is that site of cognitive activity that generates the problems and situations which require attention by the other systems. This is commonly characterized as a domain which can contain multiple humans, multiple human systems and procedures, and multiple problems. The first requirement in knowledge engineering is to define this domain in such a way as identify the portion or sub-domain of interest for knowledge engineering. This is often done from a knowledge perspective but not usually from a cognitive one where the domain may be partitioned and defined in a different manner;

c) Site C represents the cognitive processes used by the knowledge engineer during the knowledge engineering activity. These processes are likely to change as the different stages of knowledge engineering progress, so consequently the cognition-support devices will change in nature and function;

d) The user site (D) reflects those cognitive processes used to solve problems from site B with the help of site F. The user has often been included in knowledge engineering activity but seldom from a cognitive perspective. Identifying the cognitive processes used by the users in solving problems within a domain or sub-domain would help the knowledge engineer to identify attributes of the artificial system which may complement, support, assist, or replace those of the user while problem solving in the domain;

e) The expert site (E) reflects those cognitive processes used to solve problems from site A or from site D. The domain as sub-defined presents the problems for solution and a cognitive environmental history to the problems. Site E identifies and defines only those processes which are necessary and critical to the solution of the domain as defined;

f) The site (F) is that of the final knowledge-based system and reflects cognition-like properties which assist, support, or replace those of the user. Initially the system at site F is non-existent but it begins to build from the outset and acts as a focal point for discussion and interaction; and

g) The knowledge engineering activity (G) refers to the overall cognitive activity which is operative throughout the knowledge engineering process. This site is the most complex in that it represents not only the various sites of cognitive activity and their interactions but also the changing cognitive activities over time. A portion of the cognitive activities attributed to the knowledge engineer, the expert, the domain, the user, and the final knowledge based system are contained within the knowledge engineering context. Interactions between and among the various sites are included, and also time is a factor in this defined context. The cognitive processes of the user should be taken into account before, during, and after the artificial system is completed.

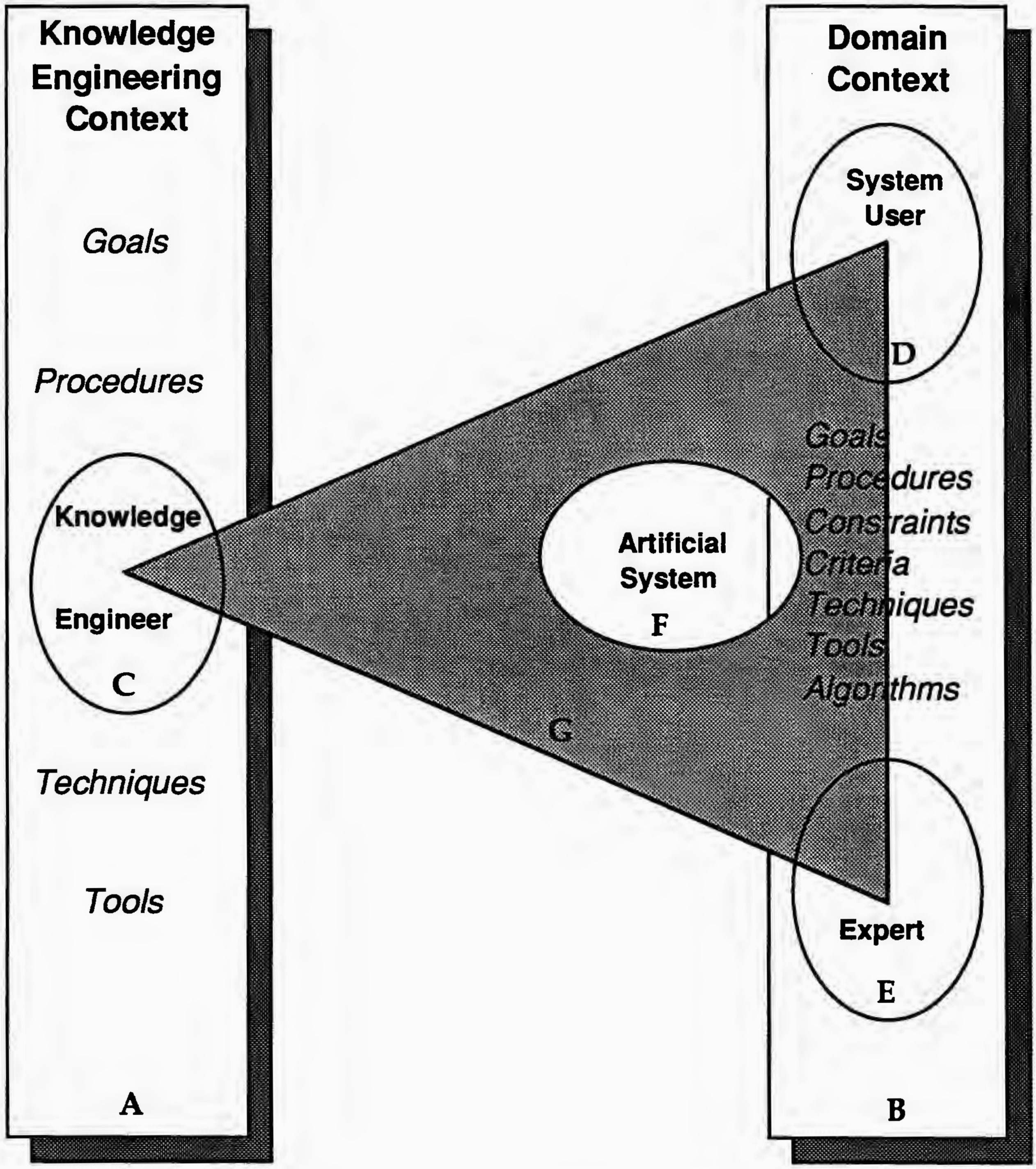

Figure 1. Sites of Cognitive Activity in Knowledge Engineering

Levels of Cognitive Activity in Knowledge Engineering

Activities of interest are those processes which reflect how representations are processed to create meaning. For example, the cognitive processes used by a knowledge engineer to create useful content categories from materials from the domain are identified at site C. The same could be said for both the expert and the user sites E and D. Once the 'system' site is developed, the cognition-like activity at this site is also of interest from a cognitive environment perspective.

The cognitive processes which characterize *intra-site* activity are considered Level I processes. These processes are often responsible for the initial generation and production of representations but they also include cognitive processes which handle representations from other sites. Meaning is accomplished through the development of mental representations or processes which help to structure and put to use the representations available. Other examples of outcome are inductive models, partial representations, and intermediate representations. Meaning is the goal which is accomplished through the use of those cognitive processes at the intra-site level.

Level II processes are concerned with *intrr-site* activity which creates meaning through communication processes between and among the sites. Level II activity is concerned with the cognitive processes involved when sites interact and exchange representations. At this level of activity, representations are re-created so those processes which are involved in re-creation through communication are of interest. Human-to-human communication may be characteristically different from human-machine communication but the final goal is still to create meaning through communication and socio-cognitive processes. Meaning is represented at this level by the development of shared, validated, and verified models. The complete development of site F (the system) may be considered the demonstrable goal of Level II activity.

Defining very precisely the cognitive context of knowledge engineering helps to identify or control intentions and to establish meaning and clarity in interpersonal discussions by severely restricting or constraining the cognitive activity of the sites. By restricting and limiting the role played by an individual, that person's cognitive processes are likewise constrained. Expectations are established, conversational rules outlined, specific goals identified and rules for interaction are used to guide conversation. Ensuring conditions for relevancy and/or restricting cognitive processes through constricting task expectations and activities accomplish the same thing: a better understanding and control of processes which reduce the ability to obtain the best information, the important knowledge without too much misinterpretation and misunderstanding.

The Methodological Framework

This model gives a strong methodological framework in which to place a knowledge acquisition, knowledge structuring or knowledge representation tool. For example, induction can be viewed as an interaction between the expert (E) collecting mass data sets from the system user (D) in the domain context (B) in order to derive rules for the system (F). Flexible knowledge engineering tools are needed to manage this context for Level II tasks from the perspectives of all involved in the system including the knowledge engineer, the expert, the software engineer, and the user.

This framework subsumes previous knowledge-oriented classifications that identify four levels of knowledge (Gaines and Shaw, 1992):

- informal knowledge including statements, behaviors, notes, sketches and photos;
- structured knowledge including verbal protocols, texts, diagrams, observations and arguments;
- formal knowledge including repertory grids, entity-relationship diagrams, conceptual graphs, state transition diagrams and derivations; and
- computational knowledge including classes, objects, relations, properties, values, inference schema, computations and interfaces.

The cognition-oriented analysis provides a more principled framework for the knowledge flows involved, and motivates the linkages between different forms of knowledge in terms of management of the cognitive processes involved in creating these linkages.

We see the theory as leading not to a single methodology of knowledge engineering, but rather as guiding the generation of families of methodologies that are specific to particular knowledge engineering tasks, problem domains, computational tools, and so on. What motivates and guides the formation of particular methodologies is the consideration of particular cases of knowledge engineering in terms of the cognitive activities involved. This case-based approach is discussed in the next section.

4 THE 'CASE': AN INTERSECTION OF COGNITIVE SCIENCE AND KNOWLEDGE ENGINEERING

At the first Joint Workshop on Contemporary Knowledge Engineering and Cognition, a number of papers identified the 'case' as a structural framework of interest to both knowledge engineering and cognitive psychology (Schmalhofer, Strube, and Wetter, 1992). Case-based approaches for developing expert systems are numerous (see Neale, 1988) and generally take the form of "examples which have occurred in reality and consist of a problem description, a solution, and the underlying justification (derivation) for that solution" (Althoff and Wess 1992, p. 146). Specifically, machine-learning, cognitive psychology, and knowledge engineering are fields of research interested in the acquisition of cases and case-based reasoning. In the first field, the approach is exemplified by inductive learning methods and hypothesis generation. The second field views cased-based reasoning as a model of problem-solving. For the final field, case-based reasoning is used in the form of knowledge modeling.

From a psychological point of view, cases are abstractions of events or processes with temporal and spatial tabs. As a consequence of abstraction, cases are models of the original experience. For Althoff and Wess (1992) cases must be described at three different levels of abstraction.: the cognitive level reflects the observable or behavioral outcomes of cognitive processes in context; the representational level reflects the structures used to distinguish the pieces of the original case for abstraction (e.g. attribute-value pairs); and finally the implementational level reflects how the case is implemented and used (e.g. lists, node and arc diagrams). Case-based reasoning is also a research paradigm in cognitive science which addresses the issues of experience and its impact on reasoning (Slade, 1991; Riesbeck and Schank, 1989). In this domain case-base reasoning means solving novel or present problems based on the application of information from already known similar problems.

Compared to computers, a human is much more capable of accumulating experience and making that experience available for subsequent use in problem-solving (Kolodner, 1984). The knowledge used in case-based reasoning is usually associated with specific events with clear temporal and spatial coordinates. This type of 'episodic' knowledge has been compared to 'special purpose knowledge' (Shavlik, Dejong, and Ross, 1987) but both tend to refer to the application of knowledge in a problem-solving process.

Janetzko and Strube (1992) comment on the basic requirements of a case-based model in cognitive science. They state:

> "There is general agreement among researchers in case-based reasoning that the feasibility of this kind of knowledge-based problem-solving is tied to a number of computational steps.....:
> - How are cases represented (i.e. how to find adequate representations of episodic knowledge and interfaces with general knowledge)?
> - How are cased retrieved from memory (e.g. from a case library)?
> - What measure of similarity is appropriate to select a suitable case in memory (=source case) given a current case (=target case)?

• How is (partial) carry-over of the solution of a source case done to the target case?
• How may (parts of) the solution of a source case have to be adapted to the constraints of the target case?
• How can general knowledge be automatically acquired from episodic knowledge
• How should a case library be organized (and re-organized)?" (p. 104)

One of the goals for knowledge engineering outlined by Strube (1992) is to "enrich present-day knowledge engineering techniques with methods specifically devised or adapted for episodic knowledge, thereby integrating case-based approaches with more traditional lines of expert systems design" (p. 173).

Case-based reasoning appears to offer a framework for unifying theories of problem-solving learning, and knowledge representation (Janetzko and Strube, 1992). These authors as well as Barstsch-Sporl (1991) have developed KADS inference structures based on case-based reasoning. Others have developed machine-based learning algorithms which used a case-based approach (e.g. Bareiss and King, 1989; Raaijmakers and Schiffrin, 1981).

Schmalhofer, Globig and Thoben (1992) and Bergamann and Schmalhofer (1991) have developed a knowledge acquisition approach based on the application of concrete past experiences in order to develop an abstract hierarchy of problem cases. The past experiences are used to develop model-based abstractions used for further knowledge acquisition from text (Schmidt and Schmalhofer, 1990) and for the development of skeletal plans (Bergmann, 1992).

Representing Cases

Attributes of the event(s) must be abstracted and represented when using the 'case' as a knowledge structure in knowledge engineering. The engineering perspective focuses on the representation of the case whereas the psychological perspective focuses on the selection-abstraction-representation process.

A template for the cases is always chosen based on the problem characteristics, the knowledge engineer's preferences and experience, and the knowledge support tools required. The knowledge engineer normally assesses the problem and selects a method of case representation. Then a series of steps are necessary to transform the informally defined case into a more formally defined one for use in a knowledge-support tool.

One main issue in representing cases is transforming examples into cases (Reimann and Schult, 1992): " If [the case] structure is simple - for example, flat attribute-value lists - then case acquisition is almost trivial. If the structure is more complex - for example - if cases contain descriptions of causal relations - then case acquisition is more demanding..." (p. 139). These authors have developed a method called 'active elaboration strategy' as a 'plan-recognition' task in which the knowledge engineer is required to "encode a given example solution in terms of problem-solving goals, operators, and relations to domain concepts" (p. 141). Solution traces from the expert(s) are used as the input for the strategy. The objective of the strategy is to build a case representation depicted by nodes and labeled arcs in a program called AXE.

Characteristics of Cognition Support Tools

Any program which acts as an aid to the knowledge engineer should also take into account the processes by which the expert or knowledge source provides the material for a case. The process of eliciting cases is one which requires attention to the cognitive activities inherent in 'case production'.

Cognition support tools are defined as devices which are developed to assist in some manner the cognitive processes apparent during cases of knowledge engineering. More specifically, these devices are instruments or implements designed to accomplish the goal of assisting a specific cognitive process or small set of cognitive processes during the various stages of knowledge engineering. *Tools* are defined as those devices which are automated to a large extent whereas *techniques* are defined as a method or sequence of short steps which are minimally automated. A device which uses, manipulates, organizes, and presents information in a manual, or mostly manual fashion is considered a technique. A spell-checker is a tool but scanning a document for consistent verb tense is a technique. A procedure can be defined as the execution or performance of a patterned or organized series of steps. A procedure may use a series of techniques and/or tools. Converting written prose into a semantic net or using protocol analysis are examples of procedures in which many techniques and tools may be used in an ordered sequence.

There are a number of methods for classifying or typing the various examples of tools, techniques, and procedures based on function, purpose, or structure. For example, they may be developed:
1) to support, enhance, or replace cognitive processes;
2) to assist specific site-based cognitive activity (user, knowledge engineer, expert);
3) to assist inter-site activities (interpersonal cognition, human-computer communication); or
4) to assist those cognitive processes which generate usable knowledge structures vs. those cognitive processes which assist in problem-solving, understanding complexity, or generating meaning.

Due to the cognitive load (that is, the degree of cognitive processing required to complete a step or process and reflecting the number of simultaneous and sequential tasks required by the knowledge engineer) placed on the knowledge engineer during some stages of knowledge acquisition, tools specifically designed to reduce the cognitive load would be useful. Pfeifer et al (1992) suggest that such support tools should allow for:

"• easy modification (including consistency tracking) of task structure
• ways of graphically representing the results to provide the knowledge engineer with an overview of the task structure
• ways of browsing through the taxonomies (tasks and problem solving methods) ....
• ways of manipulating the content itself (marking, extracting, connecting, editing, etc.)" (p 71)

The following section introduces concept maps as a general cognition support tool satisfying these requirements.

5 CONCEPT MAPS

Concept maps have long been used as a method to structure argument forms, and as a visual language for expressing relationships between events (Gaines and Shaw, 1993). Concept maps have been used as knowledge acquisition tools (McNeese, Zaff, Peio, Snyder, Duncan and McFarren, 1990) and as tools to support the interviewing process in knowledge acquisition from experts, for example in the Wright-Patterson development of the pilot's associate (McNeese et al, 1990). In management, Axelrod proposed cognitive maps as a means of representing the conceptual structures underlying decision making (Axelrod, 1976), and these have been used empirically to analyze organizational decision making (Eden, Jones and Sims, 1979), social systems (Banathy, 1991) and the policies of political leaders (Hart, 1977). In linguistics, Graesser and Clark have developed an analysis of argument forms in text in terms of structured concept maps with eight node types and four link types (Graesser and Clark, 1985), and Woodward has developed tools to extract such maps from text (Woodward, 1990). They have also been used extensively in education, mainly to investigate a student's understanding of some topic such as concepts in science (Novak and Gowin, 1984), and there are many different forms that have been applied in this field (Lambiotte, Dansereau, Cross and Reynolds, 1989). Figure 2 (from Novak and Gowin, 1984) shows such a concept map for water, showing some related concepts and propositions which can then be discussed by a teacher and students.

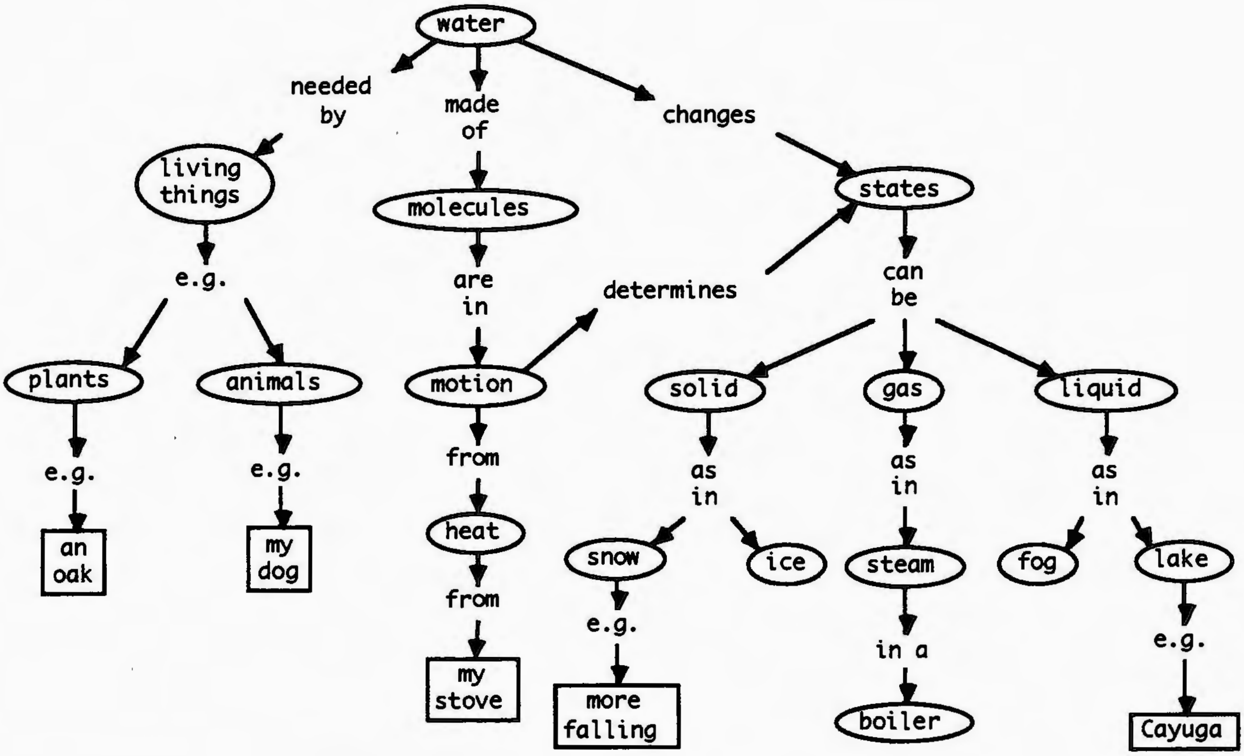

Figure 2. Concept Map from Novak and Gowin

In the history of science, the dynamics of concept maps have been used to represent the processes of conceptual change in scientific revolutions (Nersessian, 1989; Thadgard, 1992). In the philosophy of science, Toulmin developed a theory of scientific argument based on typed concept maps (Toulmin, 1958) that is regarded as one of the major models of the rhetoric of western thought (Golden, Berquist and Coleman, 1976). An example is shown in Figure 3.

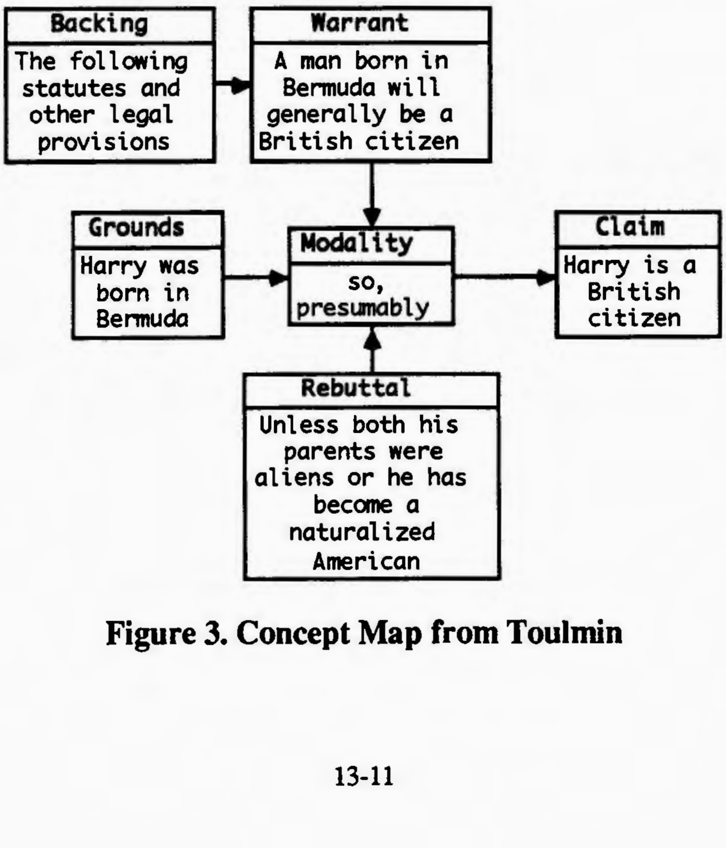

Figure 3. Concept Map from Toulmin

Bernstein, Smolensky and Bell (1989) have developed a similar hypertext tool, EUCLID, in which users can develop arguments. In a study of how graduate students use such tools to investigate their own research ideas, eight students created concept maps showing how their research related to existing literature and tools in the field (Gaines and Shaw, 1993). One student produced maps which he then went on to develop into an expert system on the protection of computer software, one of which is shown in Figure 4 (Madan, in preparation).

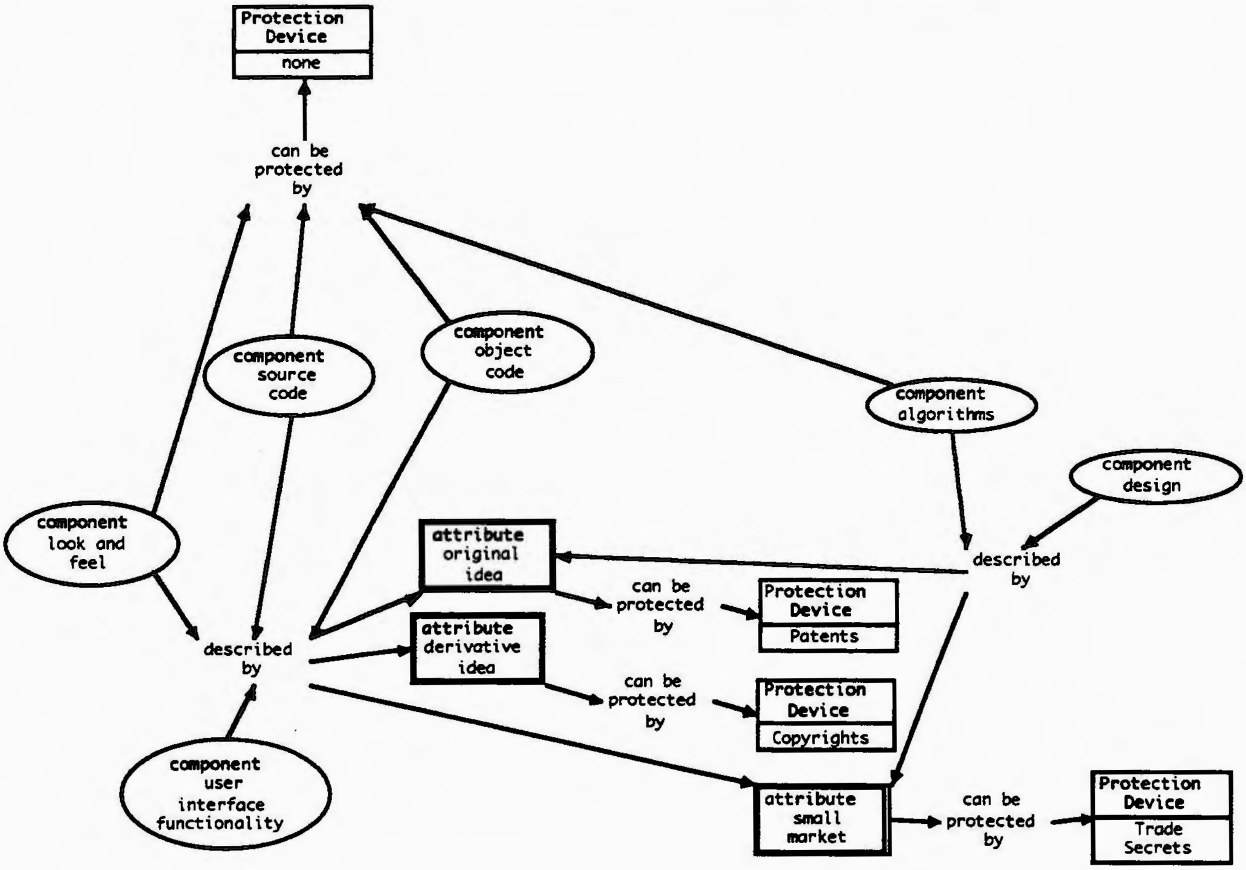

Figure 4. Top Level Concept Map on The Protection of Computer Software

ConMap (Gaines and Shaw, 1993), which was used to produce Figures 2 to 4, is a generalized visual language tool that provides a user interface and data structures for creating, editing, analyzing and exporting typed graphs with directed and undirected links. The user interface allows ConMap to be tailored to particular requirements by specifying the node types required, and associating with visual interface characteristics such as shape, color and text attributes. ConMap recognizes a variety of user interface items, such as menus and buttons, that need not be specified in advance, and this enables its interface to be tailored by specifying different interface resources rather than modifying code. The specified types and interface then allow graphs to be generated using these node types together with lines and arrows as links. Links may be labeled using specified link label node types.

ConMap supports all the interactive graphical capabilities for editing, printing and so on. It supports an inter-application communication protocol enabling other programs to make use of the graphs, and to interact with the users through the graphs. Hence it may be used as a front end to many other systems such as conceptual graph and KL-ONE deductive systems, qualitative simulation systems, and Petrinet and category-theoretic proof systems. ConMap may also be used more informally to specify concept mapping environments, and to develop concept maps within them. It is this type of use that is incorporated in the Case Map system.

6 CASE MAP

Whereas Novak and Gowin use concept maps in a declarative manner to explore a domain which is being taught to students, Toulmin uses them in a procedural way to develop a structured analysis of argument forms. Case Map also uses them in a procedural manner to enable the knowledge engineer to devise a set of questions to elicit the significant events in the process of constructing knowledge structures.

Case Map was designed to elicit and display information about actual occurrences or episodes. The elicited information is stored in a manner which allows further manipulation. For the knowledge engineer, Case Map provides a means of systematically questioning the expert about an episode (Case) and a means of mapping the results in a way which is easy to discuss (Map). In doing so, Case Map supports some of the cognitive processes involved in reduction by providing a means for selecting the needed information categories and for systematically eliciting this information. The categorized information is then displayed in a structured form for clarification, elaboration, and confirmation.

Some of the cognitive processes involved in transformation are also supported. The process of categorical elicitation of an episode or case provides a means of transforming available expert information into related, meaningful packages. The information comes 'tagged' with its category to help provide meaning, and these structures are available for further transformation and manipulation desired by the knowledge engineer.

Case Map is comprised of two main components. The first component takes a list of questions provided by the knowledge engineer and inserts them in a display sequence shown to the expert. This component stores the information in list form to be used by the mapping component and can also be used by other automated knowledge engineering tools. The second component, ConMap, graphs the resulting questions and their responses in an interactive graphing environment. The two components are fully integrated so that any changes in one component is automatically displayed in the other.

A 'case' is defined as an example or episode of interest which has occurred. It is composed of a sequence of activities generally called 'events'. The events may be steps in a process, a series of decisions, a series of tasks, or a set of components in an episode or case. The knowledge engineer uses this component of Case Map to insert the list of questions she wishes to ask of the expert. These questions are developed from protocols and interviews with the people involved, covering the things they find appropriate for a particular expert in a particular context. The questions take many forms depending on the approach used by the knowledge engineer and on the stage of knowledge engineering. For example, the knowledge engineer may wish to address the task components of a case so her events are tasks or steps in a task. For example, her questions then may be:

> What preparation was required?
> How did you complete this step?
> Why was this step involved?
> What was the result of completing this step?
> What tools were used in this step?
> What information was necessary in this step?

If the knowledge engineer wished to elicit information about the decision points in an episode or case then the events would be defined a decisions and the following questions might be asked:

> What information was required to make this decision?

What related information was required but not used directly?
What did you hope this decision would accomplish?
What tools or procedures did you use in making this decision?
What was the outcome of this decision?

These questions are asked in the same order for each event. The elicited information is stored and available to the mapping component for display.

The mapping component of Case Map provides a structure for the elicited material. This map consists of an overall display structure, boxes to contain the elicited information, and labels for each box containing the full question or its abbreviation. The display structure has three options. The vertical graph places the events at the top of the graph and the question responses descend from the events. The horizontal graph places the events along the left side of the graph with the elicited information flowing to the right. The diamond graph places the events in the middle of the graph and the elicited information is placed to the left and right of the events. The graphs are interactive so that boxes and labels can be added, moved, edited, and deleted. The changes are stored as they are made. The graph is used for discussions between the knowledge engineer and the expert for clarification, validation, and correction.

Case Map has been used extensively in counselling situations where someone presenting a problem would analyse it according to the questions elicited by the counsellor in an interview. For example, when a father reported a problem with disciplining his son for bad behaviour, the questions related to each event were:

Who did this event or action?
How did this person do this?
What happened when this person did this?
Describe how you felt when this person did this?
What did you think when this person did this?

The resulting graph is shown in Figure 5.

An Example of Case Map in Knowledge Engineering

One of the key areas of project management expertise is the establishment of project authority. Considered a 'softer' area, it is also identified as essential to the success of any engineering project. Successful project managers invest a great amount of time and effort in the process of project authority. Authority is generally delegated to the project manager but power and influence is granted from staff. Unlike a functional or line management position where power and influence come from position, the project manager establishes these attributes from personal credibility, expertise, decision-making, planning, controlling, and organizing.

The establishment, negotiation, and acceptance of project authority provides the project manager with the means to identify her lines of responsibility and authority as well as that of others key to the project. Gaining the acceptance and support from the functional hierarchy for the authority structure requires organizational, interpersonal and political skill and expertise. The most common problems of project authority stem from a variety of issues, some common to all organizations, some idiosyncratic to a single organization.

The concept map shown in Figure 6 is derived from the case description from an experienced project manager who has developed a set of procedures for establishing project management authority structures. The project manager was asked to respond to the general questions in Case Map, give the sequence and a brief description of each of the procedures, and respond to the question set developed by the knowledge engineer. This question set represented a 'first cut' at the area:

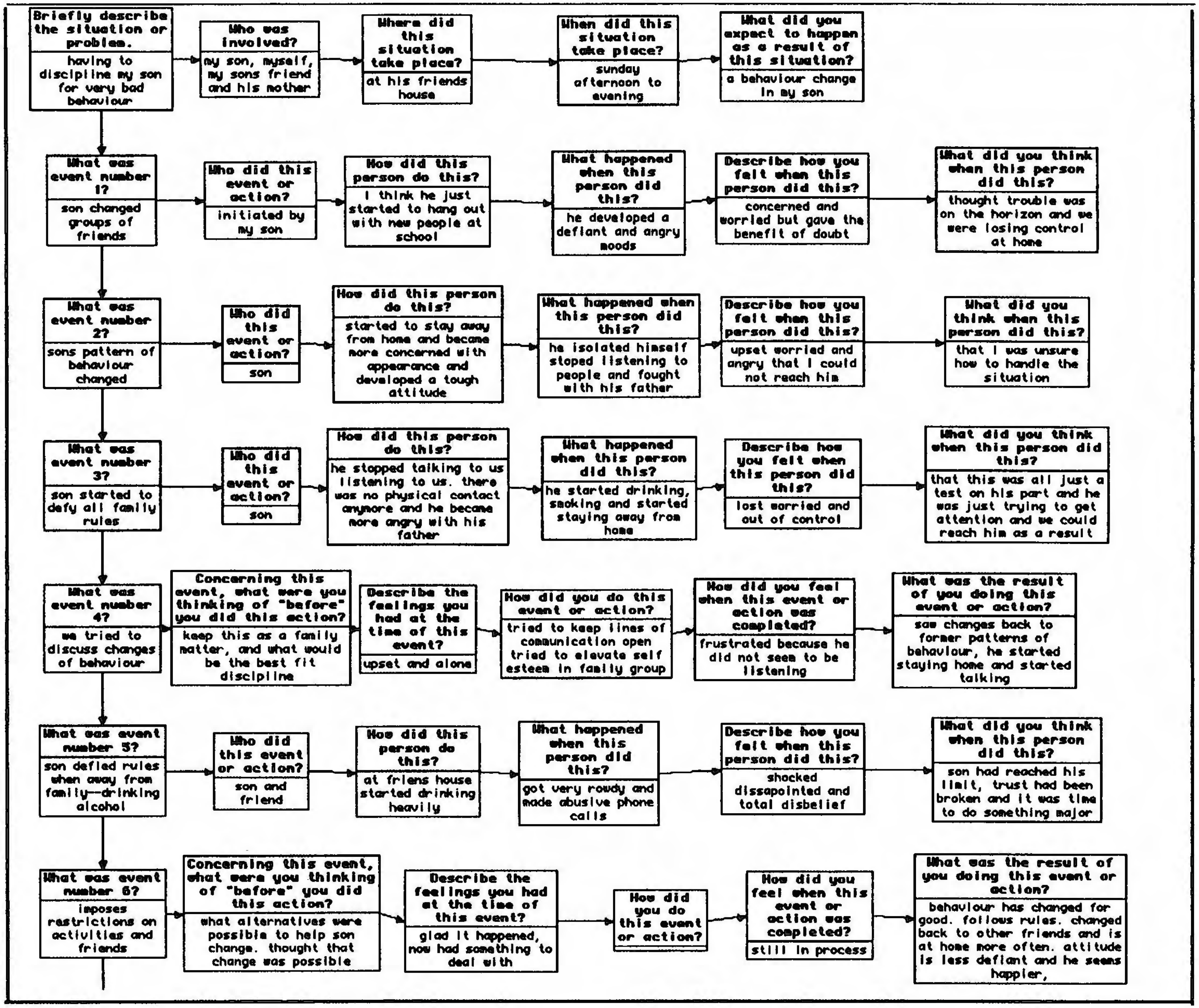

Figure 5. Concept Map used in Counselling

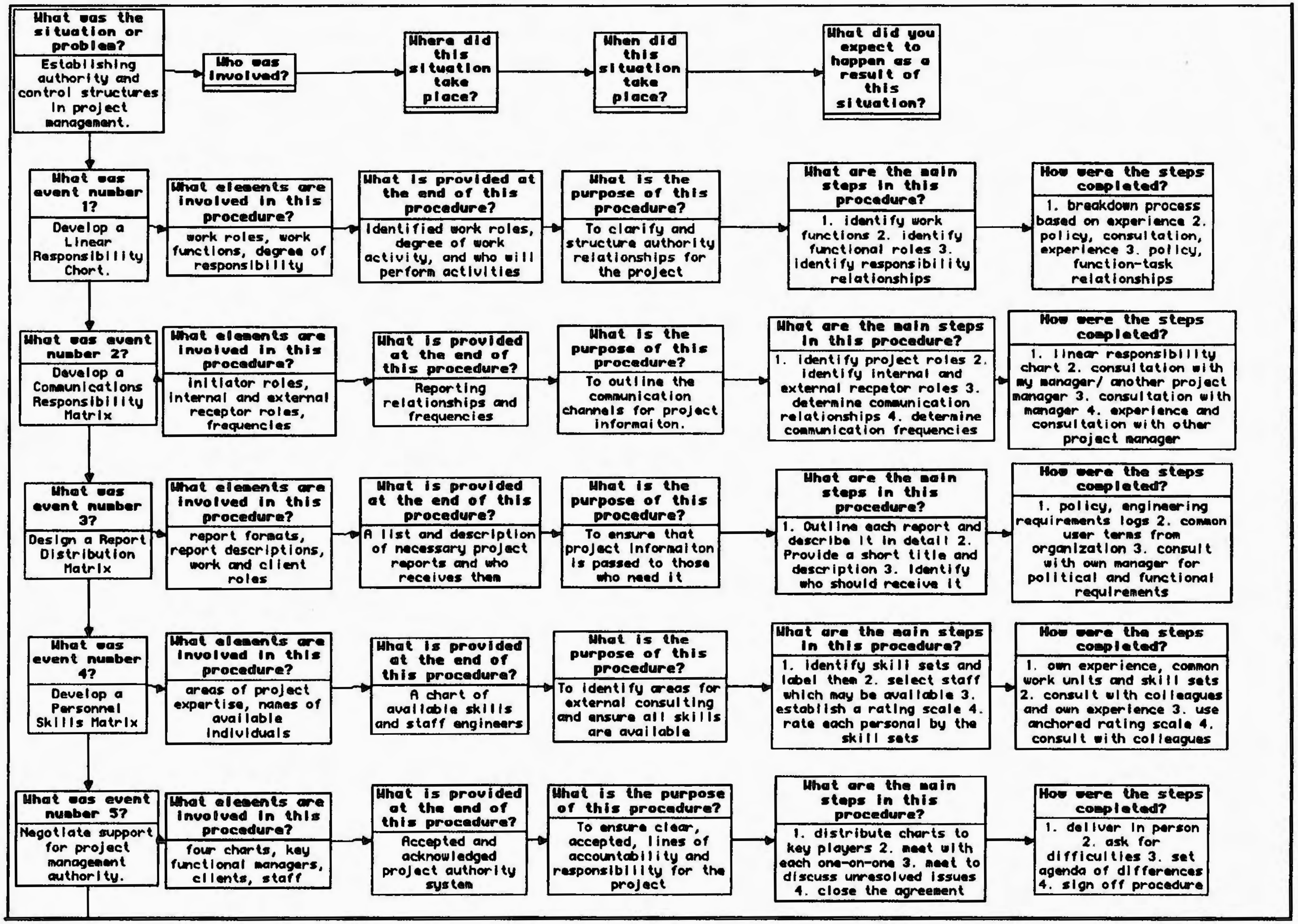

Figure 6. Concept Map from the Project Manager

• What are the key elements involved in this procedure?
• What is the main result provided at the end of this procedure?
• What is the purpose of this procedure?
• What are the main steps in this procedure?
• How were the steps completed?

The project manager then responds to each of these questions in turn for each event pertaining to the situation under discussion. The initial questions shown on the top line of Figure 6 are intended to establish the context of the inquiry. These, and in fact all the questions, are optional. If a respondent chooses to omit one or more questions, that in itself may prove to be a valuable discussion point. Where questions have been omitted, they show up as just headers as shown at the top of Figure 6.

Figure7 shows an alternative layout of the map with abbreviated headings to act as a reminder of the questions.

Now the situation under consideration has been made overt and explicit, the user can see if it makes sense in terms of the whole problem. The map can then be edited or amended to take account of any new aspect which becomes apparent.

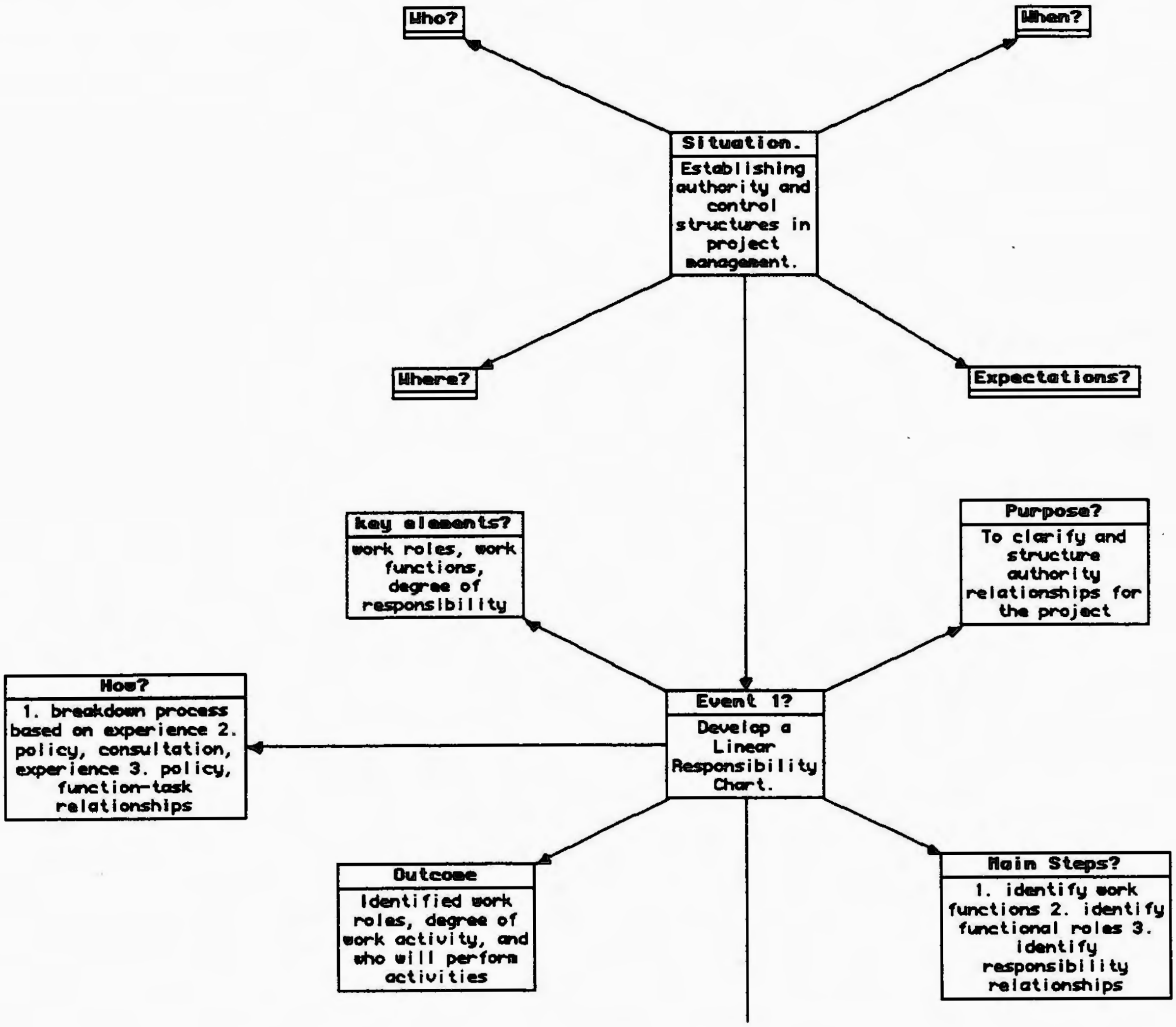

Figure 7. Alternative Layout of the Concept Map from the Project Manager

Future Developments

Maps such as the ones above can be used for feedback to the expert in order to check procedures, find holes in processes, and identify patterns of questions or responses. This is an iterative process where the map can be continually edited until the expert feels that it captures the situation as well as it possibly can. Case Map is a general tool which can be used at different stages in knowledge engineering, with different experts, and at various levels of detail. One direction for future work is to investigate ways of comparing and combining maps from different parts of the system and from different sources. Another future direction is to examine in greater detail other cognitive processes which could benefit from such support in elicitation, display of results and the use of the results for further input into the knowledge-based system, and to carry out a full evaluation of the methodology and the tool.

5 CONCLUSIONS

The objectives of the research reported in this paper have been to develop a theoretical framework for the cognitive processes involved in complex knowledge engineering tasks; to use this framework to develop a methodology for managing these processes; and to develop tools that support the methodology. The tools described are basically those for general concept maps specialized to different cognitive activities by the use of node types deriving from the methodology. The node types are developed as part of the knowledge engineering process and posed as questions to provide a well-defined elicitation methodology specific to a particular knowledge engineering activity. It does this by breaking the tasks identified by the knowledge engineer into a sequence of actual episodes or situations. Each situation is then broken down again into questions such as *Who was involved? Where did this situation take place? When did this situation take place? What did you expect to happen as a result of this situation?* The results are then plotted out as a structured concept map in a form convenient to the user. The result is a structured concept map with heterogeneous but linked regions each corresponding to different cognitive activities involved in knowledge engineering. Further work remains to develop a complete methodology for transferring such data into a form suitable for priming an expert system shell.

ACKNOWLEDGEMENTS

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada, and in part by the Canadian Guidance and Conselling Foundation. We are grateful to Brian Gaines for the use of ConMap, and for his helpful comments on the paper. We are also grateful to Craig Jackson for the supply and interpretation of some of the data.

REFERENCES

Althoff, K.D. and Wess, S. (1992) Case-based reasoning and expert system development. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition.** Springer-Verlag: Heidelberg, 146-160.

Axelrod, R. (1976). **Structure of Decision.** Princeton, New Jersey, Princeton University Press.

Banathy, B.H. (1991). Cognitive mapping of educational systems for future generations. **World Futures 30**(1) 5-17.

Bareiss, R. and King, J.A. (1989) Similarity assessment in case-based reasoning. In DARPA (ed.) **Proceedings of a Workshop on Case-Base Reasoning.** Morgan Kaufmann: San Mateo, 67-71.

Bartsch-Sporl, B. (1991). KADS for (all) cases. In B. Ueberreitter and A. Voss (eds.) **Materialien KADS Benutzer Treffen.** Munich: Siemens.

Bergmann, R. (1992) Knowledge acquisition by generating skeletal plans from real-world cases. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition.** Springer-Verlag: Heidelberg, 125-133.

Bergmann, R. and Schmalhofer, F. (1991) CECos: A case experience combination system for knowledge acquistion for knowledge-based systems. To appear in: **Behavior Research Methods, Instruments, and Computers.**

Bernstein, B., Smolensky, P. and Bell, B. (1989) Design of a constraint-based hypertext system to augment human reasoning. **Research Report CU-CS-423-89, Department of Computer Science University of Colorado at Boulder.**

Coulson, R.J., Feltovich, P.J., and Spiro, R.J. (1986) Foundations of a Misunderstanding of the Ultrastructural Basis of Myocardial Failure: A Reciprocating Network of Oversimplifications. **Report #1 Conceptual Knowledge Research Project**, Southern Illinois University School of Medicine, Springfield, IL.

Eden, C., Jones, S. and Sims, D. (1979). **Thinking in Organizations**. London, Macmillan.

Eshelman, L. (1988). MOLE: A knowledge acquisition tool for cover-and-differentiate systems. In S. Marcus (ed.), **Automating Knowledge Acquisition for Expert Systems.** Kluwer.

Evans, D.A. (1989) Issues of cognitive science in medicine. In D.A. Evans and V.L. Patel (Eds.), **Cognitive Science on Medicine: Biomedical Modeling** (pp. 1-16). MIT Press: London.

Feltovich, P.J., Spiro, R.J., and Coulson, R.L. (1989). The nature of conceptual understanding in biomedicine: the deep structure of complex ideas and the development of misconceptions. In D.A. Evans and V.L. Patel (Eds.), **Cognitive Science in Medicine: Biomedical Modeling** (pp. 113-172). MIT Press: London.

Gaines, B.R. & Shaw, M.L.G. (1992). Integrated knowledge acquisition architectures. **Journal for Intelligent Information Systems 1,** 9-34.

Gaines, B.R. & Shaw, M.L.G. (1993) Supporting the creativity cycle through visual languages. **AAAI Spring Symposium: AI and Creativity.** pp. 155-162. Stanford (March).

Golden, J.L., Berquist, G.F. and Coleman, W.E. (1976). **The Rhethoric of Western Thought**. Dubuque, Idaho, Kendall/Hunt.

Graesser, A.C. and Clark, L.F. (1985). **Structures and Procedures of Implicit Knowledge**. New Jersey, Ablex.

Harman, G. (1990) Some philosophical issues in cognitive science. In Posner, M.I. (ed.) **Foundations of Cognitive Science**. MIT Press: Cambridge, Mass.

Hart, J.A. (1977). Cognitive maps of three latin american policy makers. **World Politics 30**(1) 115-140.

Janetzco, D. and Strube, G. (1992) Case-based reasoning and model-based knowledge acquisition. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition**. Springer-Verlag: Heidelberg, 99-114.

Kahneman, D.L., Slovic, P. and Tversky, A. (Eds) (1982) **Judgment Under Uncertainty: Heuristics and Biases**. Cambridge University Press: New York.

Kahneman, D.L. and Tversky, A. (1973). On the psychology of prediction. **Psychological Review,** 80, 273-251.

Kolodner, J.L. (1984) **Retrieval and Organizational Strategies in Conceptual memory: A Computer Model.** LEA: Hillsdale, N.J.

Lambiotte, J.G., Dansereau, D.F., Cross, D.R. and Reynolds, S.B. (1989). Multirelational semantic maps. **Educational Psychology Review** 1(4) 331-367.

Linster, M., and Musen, M. (1992) The use of KADS to create a conceptual model of the ONCOCIN task. **Knowledge Acquisition, 4,** 55-88.

Madan, M. (in preparation) Evaluating Alternative Approaches to Knowledge Acquisiton for a Legal Expert System **Masters Thesis, Department of Computer Science,** University of Calgary.

Major, N. and Reichgelt, H. (1990) Alto: An automated laddering tool. In Weilinga, B., Boose, J., Gaines, B., Schreiber, G.,and van Someren, M. (eds.) **Current Trends in Knowledge Acquisition.** IOS Press: Amsterdam.

McNeese, M.D., Zaff, B.S., Peio, K.J., Snyder, D.E., Duncan, J.C. and McFarren, M.R. (1990). An Advanced Knowledge and Design Acquisition Methodology for the Pilot's Associate. Harry G Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Meyer, M.A. and Booker, J.M. (1991). **Eliciting and Analyzing Expert Judgment: A Practical Guide.** Academic Press: New York.

Motta, E., Rajan, T., Domingue, J. and Eisenstad, M. (1991) Methodological foundation of KEATS: the knowledge engineer assistant. **Knowledge Acquisition**, 3(1) 21-48.

Musen, M., Fagan, L., Combs, D., and Shortliffe, E. (1987). Use of a domain-driven model to drive an interactive knowledge editing tool. **International Journal of Man-Machine Studies**, 26, 105-121.

Neale, I.M. (1988) First generation expert systems: a review of knowledge acquisition methodologies. **The Knowledge Engineer Review**, 3, 105-146.

Nersessian, N.J. (1989). Conceptual change in science and in science education. **Synthese 80**(1) 163-184.

Novak, J.D. and Gowin, D.B. (1984). **Learning How To Learn**. New York, Cambridge University Press.

Patel, V.L., Evans, D.A. and Groen, G.J. (1989). Biomedical knowledge and clinical reasoning. In D.A. Evans and V.L. Patel (Eds.), **Cognitive Science on Medicine: Biomedical Modeling** (pp. 53-112). MIT Press: London.

Pfeifer, R., Rothenfluh, T., Stolze, M., and Steiner, F. (1992). Mapping Expert behavior onto task-level frameworks. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition**. Springer-Verlag: Heidelberg.

Raaijmakers, J., and Schiffrin, R. (1981) Search of associative memory. **Psychological Review**, 88, 93-134.

Reimann, P. and Schult, T.J. (1992) transforming examples into cases. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition**. Springer-Verlag: Heidelberg, 139-145.

Riesbeck, C.K. and Shank, R.C. (1989) **Inside Case-Based Reasoning.** Lawrence Erlbaum Associates.

Schmalhofer, F. Globig, Ch., and Thoben, J. (1992) The refitting of plans by a human expert. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition**. Springer-Verlag: Heidelberg, 115-124.

Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) (1992) **Contemporary Knowledge Acquisition and Cognition**. Springer-Verlag: Heidelberg.

Schmidt, G. and Schmalhofer, F. (1990) Case-oriented knowledge acquisition from text. In Weilinga, B., Boose, J., Gaines, B., Schreiber, G.,and van Someren, M. (eds.) **Current Trends in Knowledge Acquisition**. IOS Press: Amsterdam, 302-312.

Shadbolt, N. (1992). Facts, fantasies and frameworks: the design of a knowledge acquisition workbench. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition**. Springer-Verlag: Heidelberg.

Shavlik, J., Dejong, G., and Ross, B. (1987). Acquiring special case schemata in explanation-based learning. In **Proceedings of the Ninth International Conference of the Cognitive Science Society.** Hillsdale, NJ. Erlbaum, 851-860.

Shaw, M.L.G. and Woodward, J.B. (1990) Modelling expert knowledge. **Knowledge Acquisition**, 2(3), 179-206.

Slade, S. (1991) Case-based reasoning: a research paradigm. **AI Magazine**, Spring.

Spiro, R.J. (1980) Constructive processes in prose comprehension and recall. In R.J. Spiro, B.C. Bruce, and W.F. Brewer (Eds.), **Theoretical Issues in Reading Comprehension**. Lawrence Erlbaum Associates: Hillsdale, N.J.

Strube, G. (1992) The role of cognitive science in knowledge engineering. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition**. Springer-Verlag: Heidelberg, 161-174.

Thadgard, P. (1992). **Conceptual Revolutions**. Princeton, New Jersey, Princeton University Press.

Toulmin, S. (1958). **The Uses of Argument**. Cambridge, UK, Cambridge University Press.

Woodward, B. (1990). Knowledge engineering at the front-end: defining the domain. **Knowledge Acquisition 2**(1) 73-94.

Woodward, J. B. (1992a) Developing K-ONCOCIN: a case study in the cognitive processes of a knowledge engineer. **Knowledge Acquisition, 4,** 237-258.

Woodward, J. B., Shaw, M.L.G., and Gaines, B.R. (1992). The cognitive basis of knowledge engineering. In Schmalhofer, F. Strube, G., and Wetter, Th. (eds.) **Contemporary Knowledge Acquisition and Cognition**. Springer-Verlag: Heidelberg.