Int. J. Man–Machine Studies (1987) **27**, 235–250

Creating the domain of discourse: ontology and inventory

STEPHEN REGOCZEI

Computer Studies, Trent University, Peterborough, Ontario, K9J 7B8, Canada

EDWIN P. O. PLANTINGA

Department of Computer Science, Redeemer College, Ancaster, Ontario, L9G 3N6, Canada

Ontology recapitulates philology.
(James Grier Miller as quoted by W. V. O. Quine)

If you can't build a model of it, it isn't true.
(Buckminster Fuller)

The science of personal relations is not assisted by the fact that only a few psychologists are concerned to discover valid personal ways in which persons, and relations between persons, can be studied by persons.
(R. D. Laing)

The paper describes the foundations of a methodology for natural language-based knowledge acquisition. It concentrates on a special type of context: the case in which an analyst interviews an informant who is a domain expert and the text of the discussion is carefully recorded. In this context the following paradox arises: the analyst is after knowledge, but all he gets are words. Matching concepts to the words—or, more precisely, constructing conceptual structures which model the mental models of the informant—is the task of the analyst. The conceptual structures are to be specified as sets of conceptual graphs.

To carry out this task, the clear specification of the domain of discourse in terms of an ontology and an inventory becomes necessary. The discourse is considered to include not only the text of the discussion between the analyst and the informant, but also the ever-changing mental models of both parties. The mental models are construed as modelling some object domain "out there", but the domain of discourse is created through discourse.

A step-by-step technique is given for specifying the domain of discourse with careful attention paid to version control. It is noted that different interviews about the "same" object domain may give rise to several different domains of discourse.

1. Introduction

Systems analysis, software requirements definition, conceptual database design and knowledge engineering for expert systems can all be viewed as activities which have essentially the same structure. The goal of all these processes is to acquire the world knowledge possessed by an informant who, depending on the situation, may be called an expert or a user. Through the mediation of at least one other person, generally called the analyst, programmer, or knowledge engineer, the process culminates in a "smartened-up machine".

0020-7373/87/090235 + 16$03.00/0

The bottleneck in developing knowledge-based software is the knowledge acquisition (KA) phase. Given the importance of this problem, why is the literature in this area so small, especially when compared with the literature addressing the knowledge representation (KR) issues?† We suggest three reasons for this phenomenon.

The first reason is obvious. The problems associated with the representation of knowledge are easier to deal with than the problems of "acquiring" knowledge. Representational issues are also easier to write about and discuss since they ignore the informant–analyst relation. This is the first reason why the literature on knowledge representation is so much more extensive than the knowledge acquisition literature.

The second reason is slightly more complex. KA is a real dollar-and-cents issue in industry where software must be pushed through the door within budget and on time. In most industrial settings, the analyst needs those crucial bits of knowledge which the informant possesses. In order to construct a working and workable knowledge-based system, this knowledge must be acquired.

Contrast this with a typical situation in the academic world. An academic computer scientist, engaged in research, is not producing software for some user community. The academic already knows what he wants to knows so there is no need for a KA phase. In the academic world, the analyst and informant are one and the same. The key issue from this point of view is representation. The academic wonders how he should encode or represent the knowledge in his head in such a way that it is useable by a machine.

When academics publish in the domain of knowledge-based systems, they, in clear conscience, address those issues which they consider to be crucial. Although we agree with academics that there are important representational issues, we also side with those in industry who stress that the acquisition of knowledge is the crucial problem at present. However, people in industry do not publish their results for fear of losing their competitive advantage in the marketplace. This, we submit, is the second reason for the imbalance in the favour of representation in the knowledge engineering literature.

But there is a third reason. The view of the human mind, which is dominant in our culture, is simply too static to explain the difficulties analysts encounter in the KA process. Systems analysts frequently complain that "the user keeps changing his mind". We believe that this is to be expected, that the user "changing his mind" is the rule rather then the exception. Daniel McCracken and Michael Jackson have argued that

> Systems requirements cannot ever be stated fully in advance, not even in principle, because the user doesn't *know* them in advance—not even in principle. To assert otherwise is to ignore the fact that the development process itself changes the user's perceptions of what is possible, increases his or her insights into the applications environment, and indeed often changes that environment itself. We suggest an analogy with the Heisenberg Uncertainty Principle: any system development activity inevitably changes the environment out of which the need for the system arose (McCracken & Jackson, 1982, p. 31).

† A good recent summary of KR is Ronald Brachman's and Hector Levesque's *Readings in Knowledge Representation.*

We take these comments by McCracken and Jackson very seriously. How can someone construct a system for a user if the user keeps changing his or her mind? We take this question as a challenge, and in response to this challenge we have developed our KA methodology.

The human mind is not well understood. When confronted with a poorly understood phenomenon, people invariably search for metaphors or analogies whereby they can think of the abstract entity in terms of a more concrete entity. We think that it is a distinct possibility that many of those currently involved in knowledge engineering view the human mind as an expert system. Using this metaphor, the KA problem becomes one of acquiring the production ruleset of the informant. Presumably this ruleset is fairly static. An analyst who works with the *expert system metaphor* becomes upset when the informant "changes his mind". We believe that it is time for a new metaphor: mind as an anthill in which the ants are constantly in motion. We develop this metaphor in Section 3 of the paper under the heading, "The mental models hypothesis".

To give a comprehensive account of KA, we have to depart from the standard descriptions of KA practices as published in the academic computer science literature. We believe that it is essential that the "architecture" of the informant and the acquirer be considered. While studying the "architecture" of these persons could be considered to be outside the bounds of computer science, it is very much in the domain of cognitive science and, therefore, not at all out of place in a paper dealing with issues of artificial intelligence and expert systems.

2. Overview of the methodology

This paper presents the foundations for a natural language-based knowledge acquisition methodology. Our methodology seeks to provide the best possible system description within the constraints imposed on the project, while taking the words of the informant very seriously.

The methodology relies on interviewing, diagramming, and conceptual analysis. An analyst interviews an informant and then diagrams the sentences produced in the interview in a graphic notation. When the analyst and informant have agreed that the text that forms the corpus for the domain of discourse is complete enough, the analyst performs conceptual analysis on the text (Sowa, 1984; Sloman, 1978). This leads to an ontology (a list of entity types) and an inventory (a list of instances of these entities).

The purpose of the ontology is to reduce the inherent fluidity of the KA process which McCracken and Jackson (1982) have described. The ontology list, the list of entities that "exist", defines the entity population of the domain of discourse. The document, in a sense, is a contract between the informant and the analyst that these are the only entities that will be talked about in the future and that these are the only entities the machine is expected to be able to recognize. Without such a contract, the expectations placed upon the machine are infinite. This phenomenon is well known to constructors of expert systems. For example after having constructed a well-functioning system in a limited problem domain such as medical diagnosis, users of the system may be shocked that the system cannot, for example, answer questions about baseball.

The ontology gives to both constructors and evaluators of expert systems a reasonable framework that defines what "exists". An ontology is, in fact, a definition of what constitutes the domain of discourse.

Trying to construct an expert system on the basis of collecting rules or propositions is only possible in clearly defined knowledge domains such as medicine. If the knowledge domain is not clearly defined and codified, so that it is possible to state clearly what is in, and what is not in, the domain, then project management, budgeting, time schedules, and evaluation criteria become uncontrollable. Not only do users change their minds, so do managers. As a matter of sound, practical project management practice, the ontology defines the problem domain in a field not previously codified.

3. The mental models hypothesis

In Fig. 1, we show the essential components of the person-to-person KA process based on the mental models hypothesis. The mental models hypothesis states that an individual understands the world by forming a mental model, that a cogniting† agent understands the world by forming a model of the world in his or her head.

Suppose you want to form a mental model of Alberta. You cannot look at Alberta because your eyes are not big enough to take in all of Alberta, so what you do is consult a map of Alberta. A map of Alberta is a physical model of Alberta. You fold the map up and put it away. Whatever fragmentary information remains in your head is your mental model of Alberta based on looking at the map. It is not your mental model of Alberta based on "reality"; you have not seen Alberta—you were looking at a map. But if you do want to to look at Alberta, you can get in a car and go for a ride. After the trip, your head will contain fragments of information about Alberta based on the trip; after the trip you will have a mental model of Alberta based on "reality".

How do different people's mental models get harmonized? Harmonization cannot be done directly—we cannot rub mental models together. What actually transpires is that the informant and the analyst are constantly revising their mental models (MMs) using the technology of natural language (Hagman, 1982). This is the complaint of the systems analyst: the ants keep moving. This revision of MMs takes place under the influence of input such as text. The stochastic process which connects text input with mental model revision is just as little understood as the "reminding" (Shank, 1982) process discussed in Section 8 of this paper.

MMs are rarely harmonized. But in order to construct knowledge-based software, the analyst and informant must come to an "understanding". In the interests of attaining "understanding", the knowledge must become accessible. Although the knowledge in people's heads is not publicly examinable, the KA process requires that the knowledge be represented in a fixed, publicly examinable form. We believe that the publicly examinable representation should be pictorial, or graphical, or diagrammatic. Words are a refuge which either analyst or informant might hide

† We will make use of the word "cogniting" which is a gerund formed from the verb "to cognit". We believe (tongue in cheek) that this is what the French philosopher René Descartes had in mind when he claimed *Cognito, ergo sum.* Through a typographical error, this has become *Cogito, ergo sum.* Agents who cognit are studied by cognitive science.

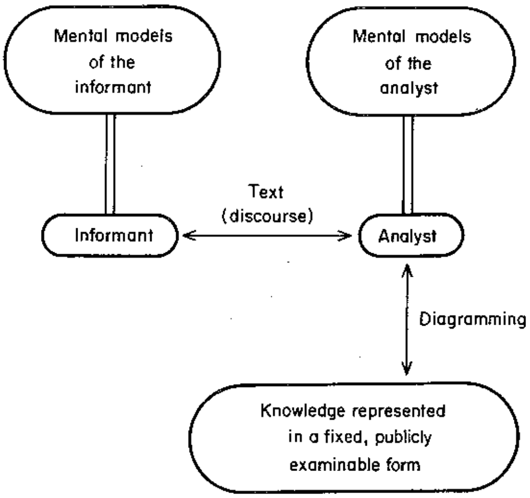

FIG. 1. A model of the person-to-person KA process.

behind to avoid making explicit the contents of the mental models. Our methodology requires that the analyst diagram his "interpretation" of what the informant said. If the informant does not "agree with the interpretation", the process is repeated until both analyst and informant converge on a set of diagrams. We expect the informant to keep "changing his mind"; we expect the ants to keep moving. We believe, however, that confronting the informant with the "meaning" of his own words (as interpreted by the analyst) will help him to "make up his mind". Is it not striking that before the recent invention of tape recorders, there did not even exist a fixed, publicly examinable way of recording what actually was said?

There are a number of candidates for representation languages. Our choice of John Sowa's conceptual graphs (Clancy, 1985; Fargues *et al.*, 1986; Sowa, 1984, 1986) as our fixed, publicly examinable form of knowledge representation was made on the basis of the following considerations:

1. Conceptual graphs stay close to the structure of natural language used by both informant and analyst.
2. Conceptual graphs are a clear notation in which to build models of MMs for public examination.

Other obvious choices are KL-ONE graphs (Brachman & Levesque, 1985) and the object-role information model of ENALIM (Evolving Natural Language Information Model) diagrams used in Control Data's Information Analysis methodology (Olle, Sol & Verrijn-Stuart, 1982).

4. Providing machines with mental models

One of the advantages of Sowa's conceptual graph notation is that the graphs (in their linear form) are directly machine representable. If the analyst can diagram the text which he receives from the informant, and if the informant can agree with the diagrammed representation, and if the diagram can be programmed into the machine, then we can provide the machine with a mental model. This process is

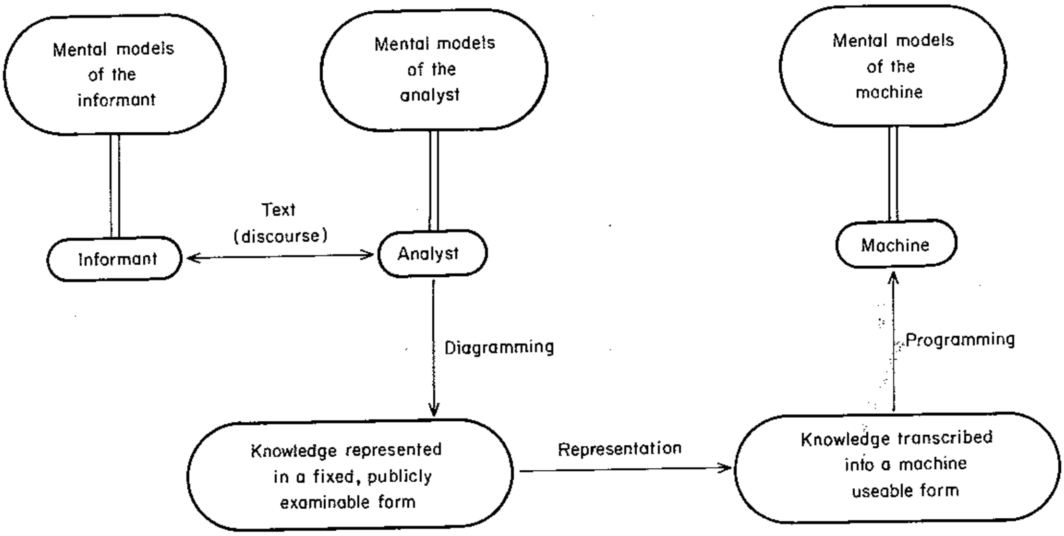

FIG. 2. A model of the knowledge acquisition process.

shown in Fig. 2 and we believe that this figure captures the essence of what any KA methodology must provide.

Sowa's notation is preferable over other notations for the following reasons:

1. The translation of a conceptual graph from its pictorial form to its linear form (the representation portion of Fig. 2) is a trivial operation.
2. The linear forms of the graph can be directly represented *without programming.*

Notice (from Fig. 2) that an automated version of this KA methodology would coalesce the role of the analyst and the machine.

The knowledge acquisition methodology which we present is based on the following hypotheses:

1. Cogniting agents, including computers, understand the world by forming MMs.
2. MMs have a structure.
3. The structures of MMs can be modelled with conceptual graphs.
4. The operations on MMs can be modelled using operations on conceptual graphs.

5. Knowledge acquisition and terminological confusion

There is a good deal of terminological confusion surrounding many KA kinds of activities. Let us consider the word "knowledge" which, in English, is used as a mass noun, indicating something of substance and bulk. "Acquiring" knowledge suggests that it is a concrete entity that could be "acquired". Reality does not support the metaphor.

Contrary to the way that English speakers use the word "knowledge", it is obvious that the concept is as intuitively elusive and as difficult to define precisely as the concepts of information and entropy. Yet the successful handling of entropy in thermodynamics and the precise and fertile definition of the Shannon–Weaver concept of information should give us hope that the concept of knowledge will also yield to a fruitful and operationally meaningful definition. Unfortunately, that time has not yet arrived.

As a substitute, and following suggestions made in the literature (Sowa, 1984; Johnson-Laird, 1983), for the purposes of this paper, we define knowledge as the semantic content of mental models. Tying knowledge to a mental model implies that there is no knowledge without a knower. This is our intent. Knowledge is possessed by a knower only in so far as the MMs "contain" this knowledge.

We assert that:

1. There is no knowledge without a knower.
2. There is no acquisition without an acquirer.
3. There is no acquisition without a source and, for us, the source is a cogniting agent.

Now let us consider systems analysis as a typical KA process. The role of analyst is played by the "systems analyst" and the role of informant is played by the "user".

Let us consider the problems in this scenario:

1. The "user" may in fact have nothing to do with the use of the system under construction. The "user" may only act as a source of expertise in creating specifications for the future system. In what sense is this kind of a "user" a user?

2. The systems analyst thinks that he is analyzing a system, but this system does not exist, except perhaps in the mind of the analyst. In trying to analyze a non-existent system, certain distortions and inaccuracies are created as a side effect of the informant–analyst knowledge transfer.

3. The systems analyst would normally use the systems approach: he would interpret information received in terms of his view of the system which is usually an input–process–output model. At times, the crucial issue is the structure of the information which goes through the input and output processes and not so much the nature of the processes or the nature of the transformation which the information undergoes. While information flows are analyzed and carefully diagrammed on data flow diagrams, the structure of the information that flows is ignored. It is pushed over to database design.

Actually, a process of fact-finding does take place. Knowledge is being acquired by the analyst and this knowledge will be built into the computer system. If the systems analyst is analyzing anything, he is analyzing chaos. His job is create order out of chaos. Attempts to describe exactly what goes on during the process of systems analysis, conceptual database design, and requirements specification writing illustrate the confused state of the discourse about KA issues.

6. Providing machines with ontologies and inventories

How does a machine know what is "out there in the world"? Write a program in the Fortran programming language which accepts as input two integers, and returns, as output, their sum. Run the program and enter a real number and the letter *k*. What happens? The program aborts. Somewhere between the programmer and the user, somebody lied to the machine. The machine, when executing this program, "thinks" that there are only integer numbers "out there in the real world".

Does the computer, when executing this program have an ontology? Only implicitly. We define an *explicit ontology* to be a publicly examinable list of entity types. In a similar vein, we define an explicit inventory to be a publicly examinable list of instantiations of concept types listed in the explicit ontology.

Each cogniting agent has, as a minimum, an implicit ontology: man, animal, and machines all have implicit ontologies. Very few cogniting agents have explicit ontologies. The clearest example of an explicit ontology is a database schema which can be printed and is comprehensible to those who understand the data definition language. Databases also have explicit inventories in the form of population tables.

Fire insurance companies sometimes request that their policy holders complete an explicit ontology and an explicit inventory. Many accounting procedures (such as recording an inventory) can be viewed from the point of view of ontology and inventory. Philosophers are also interested in compiling ontologies although inventories are a foreign notion to them. Anthropologists, it could be argued, are also in the ontology and inventory business. They visit foreign cultures and, as part of their field work, attempt to compile an explicit ontology. We believe that KA for systems development is fundamentally an anthropological activity.

How does a machine know what is "out there in the world"? Let us compare a computer to an anthropologist. The anthropologist, who wants to find out, initiates an inquiry and so discovers what is "out there in the world". The machine has no way of finding out and must rely on humans to tell it which entities are "out there". The analyst or programmer ultimately decides which entities in the world will be known to the machine. If the analyst or programmer misses a few, the system might fail at some point. This makes a complete ontology list imperative. In fact, the entities defined by the analyst-programmer team constitute the domain of discourse.

What should be the basis for the ontology constructed for the machine? The obvious choice would be to use "things in the world", things like employees, managers, employee numbers, departments, and department budgets.

There is a problem with this approach.† Both informants and analysts speak of nonexistent entities—things like systems-to-be-built. The system will only exist (hopefully) at some point in the future. And in what sense is an employee number a "thing in the world"? If we base the ontology on "things in the world", we are restricting ourselves unnecessarily.

A second possibility is to use the names of entities as the basis for an ontology. This approach leads to three problems. First, names are also entities. Thus this approach requires distinguishing between lexical and non-lexical entities as is done in ENALIM diagrams with a distinction between LOTs (lexical object types) and NOLOTs (non-lexical object types) (Olle, Sol & Verrijn-Stuart, 1982). Second, the same name, as we all know, can refer to different entities. An ontology based on names is bound to lead to confusion. Third, many entities in the "real world" lack a name although they are part of the implicit ontology of an informant. Let us

† The relationship between "data" and "reality" is discussed very thoroughly in William Kent's *Data and Reality*.

TABLE 1

1	system	subsystem	component
2	set	subset	element
3	domain of discourse	subdomain	entity
4	mental model	model fragment	concept
5	conceptual graph	subgraph	concept-node

consider an example from systems analysis. In a manufacturing environment, some partially assembled structure must be moved from one department to another. The analyst wishes to describe this inter-departmental transfer and asks the informant for the name of the sub-assembly. The informant can provide no name: the sub-assembly is an unnamed object.

Restricting the ontology to "real world" entities will not work. Nor will a name-based ontology work. We believe that the only basis for constructing an ontology is concepts. Concepts are components of MMs and can be connected to entities which are unnamed. Concepts can be named allowing the informant and analyst to discuss entities such as "system" for which no "real world" referent exist. While there are those who wish to keep their ontologies pure of "non-existing entities", we believe that a KA methodology should be more descriptive than normative.

To clarify our terminology, and to allow the reader a glimpse of our mental models, we offer Table 1.

Items in a given column of Table 1 are in the same taxonomic relationship to items in the other columns. Lines 1 and 2 give the hierarchical taxonomy of systems theory and set theory, respectively. The ontology of the computer is restricted to line 5. The inventory of the computer consists of instantiated concept-nodes.

7. Knowledge acquisition through text

If there is no knowledge without a knower, then knowledge is not to be found "in" a text. All the same, discourse, whether oral or written, does deservedly occupy a special, privileged place in KA.

How does a person go from words to knowledge? Words "create" knowledge by causing a cogniting agent to form new MMs or alter existing MMs. During discourse, the MMs of both informant and analyst are harmonized; the MMs are "brought together" using language as a technology of harmonization.†

From our point of view, KA reduces to Natural Language Understanding (NLU). We seek knowledge; we are given words. KA reduces to NLU: generating meaning for words. To generate a meaning for a word means to map a word into a concept where the concept can be represented as a conceptual graph.

Sowa has provided a mechanism for connecting words and concepts. That mechanism is a conceptual lexicon which Sowa (1984), in an example, organizes as a look-up table. For example, the lexeme "occupy" maps into the concepts [OCCUPY-ACT], [OCCUPY-STATE], and [OCCUPY-ATTENTION]. These three

† We owe this key insight to R. Hagman (1984, pp. 104–107).

concepts are illustrated by the following sentences:

- The enemy occupied the island with marines.
- Debbie occupied the office for the afternoon.
- Baird occupied the baby with computer games.

Some would say that the word "occupy" is being used in different senses. The "sense"of a word is a fairly crucial issue which will require further elaboration.

The sense of a word is defined in the *Concise Oxford English Dictionary* as: "meaning, way in which word etc. is to be understood". For the purpose of this paper, we define the "sense" of a lexeme "lex", where "lex" could be—and in fact typically would be—a phrasal lexeme, as the ordered pair

("lex", [LEX n])

where [LEX n] is the *n*th concept associated with the lexeme "lex". The sense predicate

S("lex", [LEX n])

asserts that ("lex", [LEX n]) is a sense of the lexeme "lex".

It is doubtful that lexemes and concepts are associated in human memory as a simple look-up table. The process of "reminding" in the human mind is not very well understood.† Likewise, very little is known about the process that associates a lexeme with a concept.‡ However, from the point of view of KA, only the resulting concept [LEX n] is of significance and not the process that forms the pair

("lex", [LEX n])

in the mind of the informant or analyst.

Having acquired the concept [CONC], the analyst can describe [CONC] to the informant using both natural language and the formalism of conceptual graphs. Feedback from the informant further modifies the MM of the analyst. On the basis of the feedback, the analyst may create a new concept [CONC1]. This process can continue until both the analyst and the informant are satisfied or until the allotted time for the KA phase of the project has expired.

It should be noted that there are more sense relations than are defined in a dictionary. These "extra" senses are particular to individuals and hence idiosyncratic. Dealing with these complexities can be accomplished through a formalization of the meaning triangle (Regoczei, 1987).

8. An interviewing procedure for knowledge acquisition

While the interviewing process associated with a knowledge acquisition process may be quite extensive, in this paper we will restrict our attention to the particular task of compiling an ontology and inventory list for a particular domain of discourse. Statements by the informant would be about entities. A complete ontology list

† All this talk of reminding reminds us of Roger Schank's book *Dynamic Memory*.

‡ There is a body of psycholinguistic literature on this subject but there seem to be little agreement in the literature on how the lexeme-concept association process works.

contains all the concept types and a complete inventory list contains all the instances that statements of the informant might refer to. We note that these statements are of no concern to us during the interview, except in so far as they give clues about the "existing" population of entities in the domain of discourse. Descriptive information is of use only in so far as it describes the conceptual components of the mental models of informant and analyst.

With these restrictions, the step-by-step procedure of interviewing is as follows:

Step 1. Establish a text. The text should be in a permanent, publicly referenceable form. In practice, this would mean a written document, a voice tape, or a videotape.

Step 2. Select the contentives. Select all contentive lexemes, paying special attention to all the phrasal lexemes and subfragments of noun phrases.

Step 3. Produce a lexeme-to-concept mapping. The mapping is done either through a conceptual lexicon, if one exists, or through free association by the informant or analyst. This step associates senses with the lexemes. It gives sense to the words.

Step 4. Diagram the concepts. Concepts that are semantic primitives are diagrammed as a single concept node containing the concept itself. For non-primitive concepts, the conceptual graph is as elaborate as necessary.

Step 5. Test the concepts. Elicit further comments from the informant on the appropriateness of the graphs. This step may produce more text which is to be cycled to Step 1 to produce a new version of the text.

Step 6. Model mental models. Model, using conceptual graphs, the relevant mental models of the informant to test the adequacy of the coverage. The main techniques are conceptual analysis and further dialogue with the informant. This step may produce only diagrams, in which case the interviewing procedure is complete, or more text, in which case cycle back to Step 1.

Clearly, the process may not terminate within a preset time limit. Cutoff criteria may have to be established for termination conditions.

The final product of this algorithm consists of two lists:

1. The Ontology List: a list of concepts, each defined by a conceptual graph. A conceptual graph consisting of a single concept node is a semantic primitive relative to this ontology.
2. The Inventory List: a list of instantiations of concept types relative to the particular domain of discourse.

It is assumed that there probably is an object domain "behind" the domain of discourse. The mental models of the informant are assumed to represent this object domain. As a matter of practical fact, the mental models of the informant define the domain of discourse, if it is assumed that knowledge about the domain is only accessible through the informant. The analyst's task is to establish a set of mental models in his own mind which are coherent with the set of mental models of the informant. The analyst makes a publicly examinable record of his own mental models by drawing diagrams. If the diagrams do not adequately model the mental models of the informant, then a process of adjustment will have to take place. If we look upon language as a form of technology, then the process can be described as follows: the analyst and the informant are asked to synchronize mental models using the technology of language (Hagman, 1982), assisted by a diagramming technique such as conceptual graphs.

9. Knowledge acquisition—an example

To illustrate our methodology we will run through a typical knowledge acquisition problem. In our example, the analyst is trying to find his way to a party and the informant provides directions. Admittedly, the matter is not one that requires a high degree of sophisticated expertise, but we have all been confronted by cases where the directions were so ineptly phrased that careful conceptual analysis was required to create order out of chaos.

To make the example more interesting, and to illustrate how two different ontologies and two different domains of discourse can be established for what, at an abstract level, are the "same" instructions, we shall give two versions of the text:

1. An object-oriented version.
2. A procedure-oriented version.

Both sets of instructions are about the "same" object domain, yet they create two different domains of discourse. **The domain of discourse is defined by the discourse.** Giving the full analysis of the two texts would be too lengthy for this paper. Some selected examples will illustrate the techniques.

The following texts are two sets of instructions given by two informants to an analyst on how to get to a party:

1. Text 1—Object-oriented:
   "We live at 251 Elm St.. Elm is a north–south street. The nearest major intersection is George and Hunter. We are in the southwest quadrant."
2. Text 2—Procedure-oriented:
   "Where are you now? OK. Drive down George until you get to Hunter. Turn right. Turn left after the fourth block. That's Elm. Now go down six houses and we are on the right."

We believe that it will become apparent that the ontology and the domains of discourse are quite different for the two examples, in spite of the fact that they are talking about the "same" thing.

Let us step through the procedure described in the previous section. This is artificial because the two cogniting agents, the person who gave the instructions and the person who received the instructions, are not present. We have to invent their responses and we have to make some guesses about the mental models of both analyst and informant.

Let us take the object-oriented version first.

Step 1. Establish the text. This is done above.

Step 2. List the contentives.

Normally, this would be a very long list, but this text is relatively short. We produce this list in Table 2.

Step 3. Produce a lexeme-to-concept mapping.

Looking at this list of contentives for Text 1, the analyst may decide to try to compile the first version of the ontology list. The concepts associated with the

TABLE 2

we	live	251
Elm	St.	Elm St.
251 Elm St.	north	south
north–south	street	north–south street
nearest	major	intersection
major intersection	nearest major intersection	George
Hunter	southwest	quadrant
southwest quadrant		

lexemes may turn out to be semantic primitives with no further definition, or may be defined by conceptual graphs which are constructed out of primitives and other non-primitive concepts. We show the lexeme-to-concept association in Table 3.

Lexemes such as "intersection" and "quadrant" may be difficult to handle at this stage for the analyst. We shall illustrate some possible techniques for dealing with this later. We note that at this stage that the ontology and inventory lists for Text 1 may look as shown in Tables 4 and 5.

We note that conceptual graphs would be required to connect the concepts [COMPASS-DIRECTION] and [STREET-TYPE]. The analyst may also want to connect this object-oriented text to procedural information. In this case he may ask at some future point in the process: "Yes, I hear what you are saying, but how do I actually get to your house"? At this point more text will be generated.

To contrast, let us look at the first two steps for Text 2.

Step 1. Establish the text.

This is done above.

Step 2. List the contentives.

The contentives are shown in Table 6.

We note that Text 2 is not only procedure-oriented, but also more colloquial. Colloquial language is usually ill-formed and idiosyncratic, and also makes references to entities which require a great deal of background knowledge to understand. But the analyst can still try to associate concepts with lexemes, although the process may require further information from the informant.

TABLE 3

Contentive lexeme or phrase	Associated concept
we	[PARTY-HOST]
live	[RESIDENCE]
251 Elm St.	[ADDRESS]
251	[HOUSE-NO]
north	[COMPASS-DIRECTION]
street	[STREET]
north–south street	[STREET-TYPE]
nearest	[NEAR]
George	[STREET:George]
southwest	[COMPASS-DIRECTION]

TABLE 4
Text 1 ontology list: version 1

[PARTY-HOST]
[RESIDENCE]
[ADDRESS]
[HOUSE-NO]
[COMPASS-DIRECTION]
[STREET]
[STREET-TYPE]
[NEAR]

TABLE 5
Text 1 inventory list by concept type: version 1

[ADDRESS: 251 Elm St.]
[HOUSE-NO: 251]
[STREET-TYPE: north–south street]
[COMPASS-DIRECTION: north, south, southwest]
[STREET: Elm, Hunter, George]

TABLE 6

where	you	now
OK	drive	drive down
George	get	get to
Hunter	turn	turn right
turn left	block	fourth
fourth block	after the fourth block	Elm
go	go down	six
houses	six houses	we
right	left	on the right

Now let us return to the processing of Text 1.

Step 3. Produce a lexeme-to-concept mapping (cont.)
The analyst may now want to tackle some of the more difficult concepts. Let us say that the analyst starts with the lexeme "intersection". Relying on his own mental models, and using his own "reminding" processes, the analyst may come up with the following natural language phrasings to capture the various senses of the lexeme "intersection":

1. "a place where one can pass from one street to another"
   (captures the concept [INTERCHANGE])
2. "a place where two streets meet"
   (captures the concepts of [T-JUNCTION], as well as [FORK-IN-THE-ROAD])
3. "a place where two streets meet at right angles creating four quadrants"
   (capturing the concept of [CROSSROADS])

The analyst may not be satisfied with any of these natural language phrasings of the sense for "intersection" and may, if there is no conceptual lexicon available, consult dictionaries and other sources for further information.

Suppose the analyst decides that Sense 2 and Sense 3 are the best candidates. For the sake of version control, he may not add the statements of Sense 2 and Sense 3 to the text as yet.

Step 4. Diagram the concept using a conceptual graph.

The analyst decides initially that for [INTERSECTION2] the concept will be [JUNCTION] and that for [INTERSECTION3] the concept will be [CROSSROADS].

He draws conceptual graphs for [INTERSECTION1] and [INTERSECTION2] and shows them to the informant.

Step 5. Elicit comments from the informant.

The informant expresses his opinion that [CROSSROADS] is the concept that best matches his mental models, i.e. that [CROSSROADS] is what he "had in mind" when he was giving instructions. He volunteers the following extra prose:

> Well, think of analytic geometry. The two axes intersect, dividing the plane into four quadrants. The four city blocks around an intersection, provided it is an intersection formed by two streets intersecting at right angles, are similar to the four quadrants. You remember how they are labelled: the first quadrant, the second quadrant, the third quadrant, and the fourth quadrant. Well, when you look at a compass rose, you see the same thing. Obviously, the southwest quadrant according to the compass rose is the third quadrant of analytic geometry.

This step provided more text, namely the natural language description for the sense predicate S ("intersection", [INTERSECTION3]) above, plus the extra information provided by the informant. This text should be added to the original version to produce Version 2 of the text. Now the analyst can enlarge the list of contentives above to produce Version 2 of the contentive list. Then, he may update the ontology list by adding the conceptual graph for [INTERSECTION3] to the list.

At the end of this iteration from Step 1 to Step 5 we have

- Text—version 2
- Contentives list—version 2
- Ontology list—version 2

Step 6. Model, using diagrams, the mental models of the informant.

Making an appropriate decision on the depth of the analysis, the analyst could proceed to elicit additional information about the mental models of the informant. Conceptual analysis is described by Sloman (1978) and Sowa (1984), as well as by Riesbeck (in Schank, 1975). The process may not coverge, unless decisions are taken by informant and analyst on what does, or does not, belong to the domain of discourse.

The final product is an ontology list which contains all the concepts that are semantic primitives, as well as the concepts that are defined by more complex conceptual graphs. The inventory list is assembled on the basis of the contentives list. Each item in the inventory is an instantiation of a concept type. Matching up items on the contentives list with concept types is based on information from the informant.

For example, the concept [STREET] may be accepted as a semantic primitive and the contentives "Elm", "George", and "Hunter", as instances of [STREET]. This could be recorded as a population table, or in the linear notation of conceptual graphs, as

[STREET: Elm, George, Hunter]

10. Conclusion

The necessity of relying on an outside expert as the source of the knowledge in the creation of knowledge-based systems forces us to state—more clearly than ever before—exactly what steps must be carried out during the KA process. Writing such a "procedures manual" for people engaged in person-to-person KA work is a prerequisite for creating the architecture and specifications for an automated knowledge acquisition methodology. We are still a long way from formulating a methodology which can be automated. We offer our methodology as a workable approach for performing natural language-based, person-to-person knowledge acquisition for knowledge-based systems development in a present-day environment.

References

Brachman, R. & Levesque, H. Eds (1985). *Readings in Knowledge Representation*, Los Altos, CA: Morgan Kaufmann.

Clancey, W. J. (1985). Review of J. F. Sowa's *Conceptual Structures. Artificial Intelligence*, **27**, 113–128.

Fargues, J., Landau, M.-C., Dugound, A. & Catach, L. (1986). Conceptual graphs for semantics and knowledge processing. *IBM Journal of Research and Development*, **30**(1), 70–79.

Hagman, R. S. (1982). *Language, Life, and Human Nature*. Current Inquiry into Language and Human Communication 41. Edmonton: Linguistic Research.

Johnson-Laird, P. (1983). *Mental Models*. Cambridge, MA: Harvard University Press.

Kent, W. (1978). *Data and Reality*: *Basic Assumptions in Data Processing Reconsidered*. New York: Elsevier Science Publishers.

Laing, R. D. (1961). *Self and Others*. Harmondsworth, England: Penguin Books.

McCracken, D. D. & Jackson, M. A. (1982). Life cycle concept considered harmful. *Software Engineering Notes*, **7**(2), 29–32.

Olle, T. W., Sol, H. G. & Verrijn-Stuart, A. A. Eds. (1982). *Information Systems Design Methodologies*: *Proceedings of the IFIP WG* 8.1 *Working Conference, Noordwijkerhourt, The Netherlands*, 10–14 *May* 1982. Amsterdam: North Holland, 1983.

Plantinga, E. P. O. (1986). Who decides what metaphors mean? *Proceedings of the Conference on Computing and the Humanities—Today's Research, Tomorrow's Teaching* Toronto, pp. 194–204.

Quine, W. V. O. (1960). *Word and Object*, Cambridge, MA: MIT Press.

Regoczei, S. Formalizing the meaning triangle. *Trent University Technical Report* (to appear).

Regoczei, S. Can computers—or, for that matter, people—understand natural language, and if not, why not?—the impact of the cognitive revolution (to appear).

Schank, R. C. Ed. (1975). *Conceptual Information Processing*, Amsterdam: North Holland.

Schank, R. C. (1982). *Dynamic Memory*: *A Theory of Reminding and Learning in Computers and People*. Cambridge: Cambridge University Press.

Sloman, A. (1978). *The Computer Revolution in Philosophy*: *Philosophy, Science, and Models of Mind*, Harvester Studies in Cognitive Science, Atlantic Highlands, NJ: Humanities Press.

Sowa, J. F. (1984). *Conceptual Structures*: *Information Processing in Mind and Machine*, Reading, MA: Addison-Wesley.

Sowa, J. F. & Way, E. (1986). Implementing a semantic interpreter using conceptual graphs. *IBM Journal of Research and Development*, **30**(1), 57–69.