ISSN 0834-1648

Knowledge acquisition as knowledge explication by conceptual analysis

Stephen Regoczei
and
Graeme Hirst

Technical Report CSRI-205
January 1988

COMPUTER SYSTEMS RESEARCH INSTITUTE
UNIVERSITY OF TORONTO

VELUT ARBOR ÆVO

Knowledge acquisition as knowledge explication by conceptual analysis

Stephen Regoczei
and
Graeme Hirst

Technical Report CSRI-205
January 1988

Computer Systems Research Institute
University of Toronto
Toronto, Canada
M5S 1A1

The Computer Systems Research Institute (CSRI) is an interdisciplinary group formed to conduct research and development relevant to computer systems and their application. It is jointly administered by the Department of Electrical Engineering and the Department of Computer Science of the University of Toronto, and is supported in part by the Natural Sciences and Engineering Research Council of Canada.

Knowledge acquisition as knowledge explication by conceptual analysis[1]

Stephen Regoczei

Computer Studies
Trent University
Peterborough, Ontario
CANADA K9J 7B8

Graeme Hirst

Department of Computer Science
University of Toronto
Toronto, Ontario
CANADA M5S 1A4

January 1988

[1]To appear in: Sowa, John F; Foo, Norman; and Rao, Anand (editors), *Conceptual graphs for knowledge systems, Volume I: Knowledge representation,* to be published by Addison-Wesley, 1988.

Contents

1 Introduction

In the context of expert systems, Waterman defines *knowledge acquisition* as "the process of extracting, structuring, and organizing knowledge from some source, usually human experts, so it can be used in a program" [Waterman 1986, pg. 392]. Although this definition seems general enough, in practice knowledge acquisition for expert systems reduces to the search for production rules codifying rather superficial problem-solving knowledge. The deeper conceptual knowledge is often left unexplicated. It is the goal of this chapter to outline techniques for acquiring this deeper conceptual knowledge.

Knowledge acquisition in general has a greater significance than that suggested by the expert systems literature. As Brooks has pointed out:

> The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements. ...No other part of the work so cripples the resulting system if done wrong[ly]. No other part is more difficult to rectify later. Therefore, the most important function that the software builder performs for the client is the iterative extraction and refinement of the product specifications. [Brooks 1987, pg. 17]

In emphasizing "the iterative extraction and refinement of the product specifications", Brooks presents the software builder as playing the role of a knowledge acquisition analyst.

In a sense, *all* application software contains some world knowledge. As the requirements increase to build more and more world knowledge into application software, more people in systems analysis, requirements specifications, and conceptual database design have to perfect their knowledge-acquisition skills. At one time, being a good listener was sufficient for knowledge acquisition work. However, 'doing' knowledge acquisition systematically, developing methodologies for knowledge acquisition, or fitting a knowledge acquisition stage into the life cycle of established methodologies requires more than ad hoc approaches. In particular, superior skills in the techniques of conceptual analysis are required.

This chapter is addressed to the analyst whose job it is to get world knowledge into an application system—including systems analysts, feasibility and requirements analysts, data administrators, database designers working at the conceptual schema level, and knowledge engineers who create expert systems. Thus we interpret the work of the analyst in a wide sense. The purpose of the chapter is the following:

- To give practical advice to the analyst or knowledge engineer who is doing knowledge acquisition work.
- To describe the application of conceptual graphs to knowledge acquisition.
- To give practical advice on conceptual analysis in the context of informant-analyst interaction.
- To describe the informant-analyst interaction–based, natural language–mediated knowledge acquisition process, as distinguished from other types of knowledge acquisition.

The chapter does not describe a single, monolithic knowledge acquisition methodology. Rather it provides some building blocks out of which knowledge acquisition components could be built to fit into existing systems development methodologies. The core of the chapter consists of the following material:

- A description of the informant-analyst–based, natural language–mediated knowledge acquisition process.
- A description of three types of conceptual analysis.
- A description of the Sowa-Sloman algorithm for the analysis of named concepts.

A detailed example involving the OCRA car registration system [Exercise XXX, page XXX of this book] appears as an appendix to the chapter.

2 Concepts and knowledge acquisition

2.1 Knowledge acquisition = elicitation + explication + formalization

In general terms, knowledge acquisition as a process works in the following way. An informant, who knows a lot about some domain, explains to an analyst some of what she[1] knows. Naturally, this report is phrased in her own terminology, liberally peppered with technical terms and even jargon. The analyst tries to clarify the concepts that the informant might have in mind, and represents them using conceptual graphs. The graphs are shown to the informant and are revised until both informant and analyst agree that an adequate recording of the domain of knowledge has been achieved.

The analyst does not merely 'acquire' something that already exists. Rather, he collaborates with the informant in *shaping* and *clarifying* the knowledge. The conversation between the informant and the analyst yields text, and the meaning of this text is negotiated between the two of them. The structures, both concepts and schemata, that constitute the meaning of the text are then recorded—first in the form of some informal notation, and then in terms of conceptual graphs. The resulting structured set of graphs constitutes the conceptual framework for the design and implementation of a knowledge base.

The purpose of this chapter is to take this general description and make it more precise. In other words, we are building a more formal model of the knowledge acquisition process.

We break knowledge acquisition into three stages: *elicitation, explication,* and *formalization* (see Figure 1).

At the *knowledge elicitation* stage, the analyst and informant engage in a dialogue, the context of which is set by:

- The system to be built;
- General guidelines to the domain knowledge that is required;
- Goals and objectives;
- Preliminary specification documents, and other documentation available about the problem domain.

The analyst sits down with the informant to hear about the knowledge and record it. He starts asking questions, basically prompting the informant to talk about what she knows. This is the point where one encounters the basic knowledge acquisition paradox: The analyst is after knowledge, but all he gets is words. Having heard what the informant has to say, the analyst has to re-create in his own mind a model of her knowledge. That is, having heard the words, the analyst has to synthesize the concepts. The bulk of the knowledge is conceptual knowledge about abstract entities. He describes the concepts, asks questions, and elicits further comments.

Further *explication* of the knowledge will be required. One must elaborate on the structure of the concepts, the structure of the knowledge, and how concepts relate to one another; more

[1] Our informant will be female and our analyst male, thereby making pronominal references to each unambiguous.

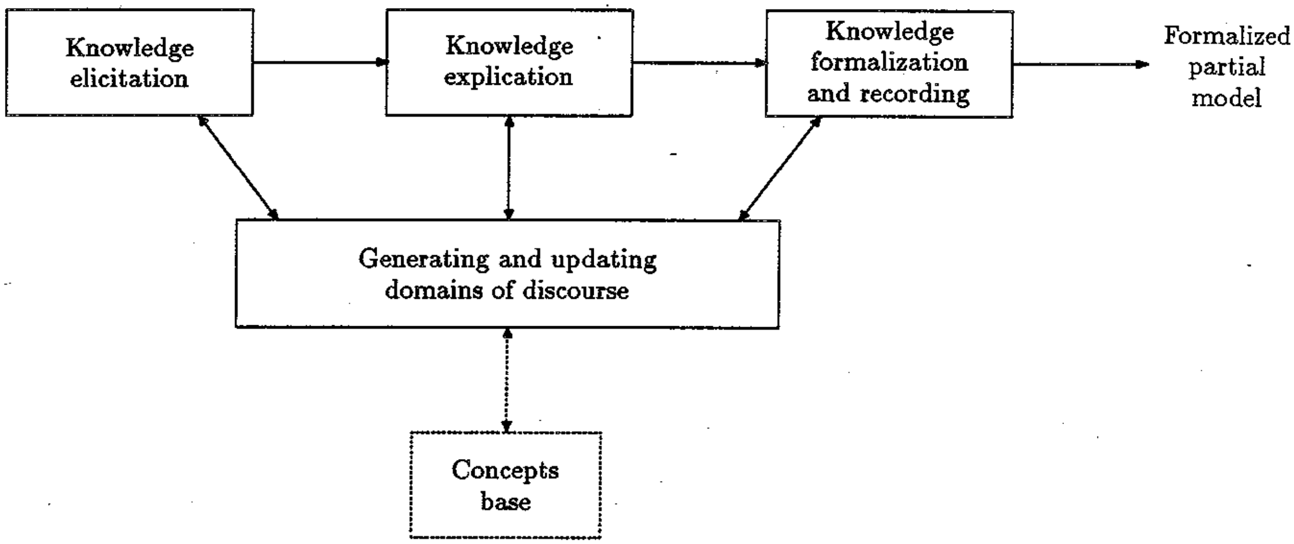

Figure 1: Processes in knowledge acquisition. The input to all processes is the mental models of the analyst and the informant. The output is a formalized *partial* model of domains of discourse (as defined by an ontology), described in conceptual graphs.

concepts may have to be introduced and connected to the ones already explored. The task includes organizing and systematizing the knowledge, separating important from less important concepts, checking the knowledge for completeness of coverage, and above all making sure that both informant and analyst are talking about and thinking about the 'same thing'.

At times, the explication task is interwoven with the elicitation dialogue. The analyst asks questions like:

> What do you mean?
> Why did you say X rather than Y or Z?
> I don't understand this.
> Can you say more?
> How does X relate to Y?
> Who does it?

Thus the informant is prompted to provide further information in the form of natural language text.

Formalization and recording of knowledge in a publicly examinable form is often referred to as knowledge representation. This can be done in a natural language like English, but it is more desirable that a formal language be used. The language we use in this chapter is conceptual graphs, for two reasons:

- Conceptual graphs stay close to the structure of natural language used by both informant and analyst.

- Conceptual graphs are a clear notation in which to build models for public examination.

The formalization of the knowledge involves a complex series of processes. In broad terms, it includes definition of the *domain of discourse, i.e.,* establishing an entities list that contains the names of all the basic entities that enter into the creation of an *ontology.* In addition, *inventory* is taken to determine what concepts in the ontology have actual instances occurring in the *object domain.* We discuss these ideas in detail in section 2.4. The ontology list and the inventory list are important intermediate products of the knowledge acquisition process.

The final product of the knowledge acquisition process can only ever be a *partial* model of the domain of discourse. The analyst cannot formalize and record all of the knowledge at once. He has to proceed in stages. To begin with, since he receives his information through natural language text or natural language speech, he has nothing but initial fragments to start his work. Partial formalization of informal knowledge is very much like the construction of Hintikka surface models as described by Sowa [Sowa 1984, pp. 179, 184]. Because the formal model of knowledge is built in stages, strict version control is required.

2.2 Concepts

This chapter is mainly about concepts. We look upon concepts as the fundamental building blocks out of which larger conceptual structures are to be built. For example, we can consider conceptual knowledge as consisting of concepts linked together to form more complex structures.

The term *concept* is commonly used to refer to at least three different things:

- A semantic object (cf [Hirst 1987]) in the mind of an agent—an *agent-based concept;*
- An abstract entity in the external world that exists as a socially constructed public concept, of which the agent-based concept may be a model;
- A Platonic entity having existence independent of time, space, people, and social constructs.

In this chapter, we are using the term the first way. As Sowa says,

> Concepts are inventions of the human mind used to construct a model of the world. [Sowa 1984, pg. 344]

We can make a couple of observations that are direct corollaries of this statement.

- There are no concepts without people.

That is, there are no concepts independent of human minds. They are data structures for the software that runs on people. The metaphysical position that attributes abstract, independent, disembodied existence to concepts is not very useful in knowledge acquisition work.

- The world includes not only the physical world, but also other people's concepts and mental models (see section 2.4).

This observation is a form of the relativistic principle. To each of two observers A and B, the contents of the other's minds are in the 'outside world'.

2.2.1 Public concepts

We must distinguish between private concepts and *public*, or *shared*, concepts. Although concepts are created or invented or generated by individuals, some of these concepts are expressible in language. Such concepts can be shared or shaped or refined through social interaction within a group. Although each individual member possesses within his mental models a private copy or copies of the shared concept, and although each one of the private copies is slightly different from the others, the members of the group, and even outside observers, will perceive a commonality. It is this commonality that characterizes shared, or public, concepts.

2.2.2 The naming of concepts

Many concepts, in particular many public concepts, have names. But many do not. Sowa [Sowa 1984, pg. 37] cites examples of experiments by Willwoll that demonstrate that people can possess a concept, and even be able to use it, without necessarily being able to attach a name to it. Concepts without names are necessarily private concepts, particular to the mental models of an individual. These concepts can be put in the public domain only by associating them with language, that is, by attaching publicly recognizable names to them.

In conceptual analysis, we often have to give names to nameless concepts. The name of the concept is not the concept itself. The concept is a mental entity; the name is usually a character string. The naming conventions for concepts are given by the notational standards for conceptual graphs described in Chapter 1.

To make the notation 'user-friendly', the type label in the name of a concept corresponds to one of the natural language terms commonly used for that concept. So, for example, the concept of dog may be denoted by the name [CANINE], as opposed to a less mnemonic designation such as [G203] or [XXYYZZ]. The character string [CANINE] is a name of the concept of dog. In ordinary English prose, however, we seldom use the full syntactic construction such as "consider the concept named [CANINE]". We say, more informally and naturally, "consider the concept [CANINE]".

2.3 The meaning triangle

The *meaning triangle* [Ogden and Richards 1923,Sowa 1984, pg. 11], illustrated in Figure 2, is a helpful tool for the analyst to keep in mind as he does knowledge acquisition work. Its main function is to help draw a distinction between words, concepts, and referents.

Words are concrete linguistic entities embodied in speech or in written text. *Concepts*, as we have seen, are abstract, intangible entities, usually associated with human thinking. *Referents* are usually construed as 'objects out there' with actual existences, although it should be noted that concepts and words can also be referents without their necessarily being physical objects.

Conceptual analysis associates concepts with words and with referents. For example, the words *bird* and *fly* are associated with [BIRD] and [FLY]. It is commonly agreed that there certainly are referents 'out there' of [BIRD: ∗], and that [FLY: ∗] is an observable activity. In the statement *Unicorns cannot fly*, the word *unicorn* can be associated with the concept [UNICORN], but we

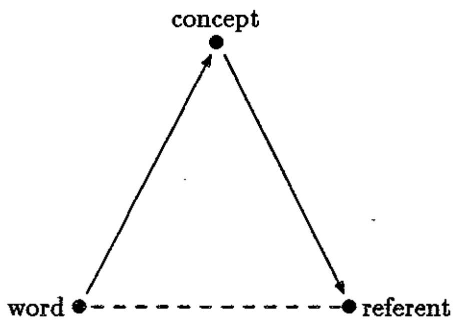

Figure 2: The meaning triangle.

Table 1: Classes of concepts, categorized according to their kind of referents.

Lexical concepts	Referents are words, or verbal entities such as names, text fragments, diagrams, etc.
Abstract concepts	Referents are abstract entities, which can be considered to be social constructs.
Concrete concepts	Referents are perceivable, physical objects, actions, events, or processes; there are percepts associated with the objects.

will not find a referent for the concept in the real world. Nevertheless, it would be a mistake to say that unicorns don't exist, for they do have ontological status as concepts, mythological figures, images in heraldry, and in pictures on Canadian banknotes. They just don't exist as biological entities with an actual, physical presence. Many of the entities that an analyst encounters in his regular work are unicorn-like, especially in the domains of law, management, corporations, organizational structure, and abstract sciences.

The meaning triangle summarizes the three categories of existence: the verbal, the conceptual, and the external (both physical and abstract). Some schools of thought connect words directly with referents, denying the existence of the conceptual domain.[2] It is important to note that in the knowledge explication methodology that we describe in this chapter, concepts play the central role.

We have noted that words and concepts may also be referents, and that concepts of words and concepts of concepts are concepts to be placed on the top vertex of the meaning triangle. We may distinguish classes of concepts according to the referent they are associated with. This is summarized in Table 1.

The simple form of the meaning triangle can be replaced by more extended and elaborate versions as the circumstances warrant. For example, following Martin [Martin 1975], we may

[2] A critique of such objectivist approaches, *e.g.*, those of Kripke and others, is given in [Johnson 1987, pg. 200ff].

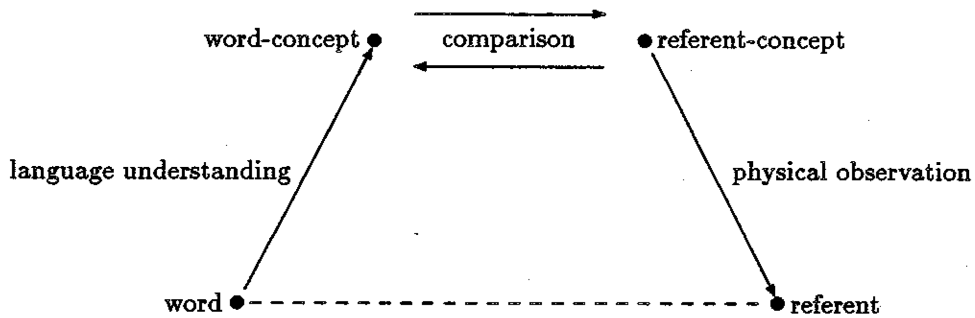

Figure 3: Martin's extended meaning triangle.

distinguish the concept associated with the word, that is, the *word-concept*, from the concept associated with the referent, that is, the *referent-concept* (Figure 3). Mapping the referent to the referent-concept is very much like physical observation; it is the matching used in inventory taking (see section 2.4). Mapping the word to the word-concept is like natural language understanding, and is a key operation at the elicitation and explication stages.

Mapping words to word-concepts takes place in the confines of a dialogue between the informant and the analyst. For the mapping of referents to referent-concepts, one actually has to go out and examine the world. Some authors interpret knowledge acquisition as the referent–to–referent-concept mapping; as, for example, pattern recognition, image recognition, scene recognition, process monitoring, etc. That is definitely not what we are talking about. We are talking about the word–to–word-concept mapping, but at the same time we have to make the point that inventory taking is necessary in order to anchor the knowledge in the external world. All comparisons of word-concepts (generated after hearing text) with referent-concepts (generated after observing the external world) take place at the conceptual level.

The main use of the simple meaning triangle is in the compiling of an entities list. The three corners of the meaning triangle represent the three classes of entities that will be included in the list: words, concepts, and referents. Each entity in the list must be carefully labeled as to its ontological status. This is because in knowledge acquisition work, it is important to distinguish between, say, the word *automobile*, the various concepts [AUTOMOBILE$_i$], and an actual four-wheeled motor vehicle in the real world.

Since we cannot put non-lexical entities down on the page, we write only the lexical labels of concepts or referents. In ordinary language use, there is unfortunately no special notation to distinguish between words serving as lexical labels of concepts and those serving as lexical labels of referents. There is, however, a convention to denote words qua words; for example, "fish" or *fish* is the lexical label used to talk about the word:

fish

In conceptual graphs notation, [FISH] is the lexical label of the concept fish, and [FISH: #] is the lexical label for the referent fish [Sowa 1984, pg. 84]. [FISH: #23456] may be the surrogate for a referent fish in a computer. The surrogate 'stands in' for the referent fish, because one may not be able to pack the actual referent fish into a computer (or into a brain, for that matter).

The extended meaning triangle allows us to go further. It allows us to distinguish concepts anchored in the physical world from those anchored in language.

2.4 Mental models, ontologies, and inventories

The *mental-models hypothesis* is a useful device to help the analyst in his thinking. This hypothesis states that people understand the world by forming mental models; so when the informant is talking about the world 'out there', she is really only talking about—*can* only talk about—her own mental models of that world. We must distinguish between the *object domain* (the external world), the *domain of discourse*, the mental models of the informant and the analyst, and the text generated between the two of them.

We note that several different domains of discourse may be constructed for what is ostensibly the 'same' object domain. The phenomenon of multiple discourses is an unavoidable feature of knowledge acquisition. The main concern is that if the domains of discourse exist only within mental models and are merely artifacts created by the analyst and informant, then how do these constructs get matched to the physical objective world outside the closed system of analyst and informant? We need a mechanism whereby the types of entities that might exist in the domain of discourse are matched up with the actual physical, social, and abstract entities in the external world. This is *taking inventory.* We try to establish what is *really* out there.

The distinction between the two domains is best illustrated by a simple example. Let us suppose that we have an informant who talks about a bank account 'out there' with a million dollars in it. The concepts that we could call by the names [BANK-ACCOUNT], [ACCOUNT-BALANCE], and [ONE-MILLION-DOLLARS] are concepts in the domain of discourse. However, further investigation reveals that although there is a real bank account with account number 26-2085, the balance of the account falls considerably short of one million dollars. Thus we may say that inventory-taking revealed that the referent [BANK-ACCOUNT: #26-2085] does indeed exist in the object domain, but that the existence of the referent

[ACCOUNT-BALANCE] → (QUANTITY) → [ONE-MILLION-DOLLARS]

is somewhat fictitious.

2.4.1 Harmonizing mental models

As noted above, multiple domains of discourse are normally created. Both observers A and B have their own mental models of the world. For A and B to be able to communicate and refer to the same entities in the outside world, their mental models must be *harmonized* with one another. How can this be achieved? Both A and B interact with the world at the physical level and that, presumably, imposes some coherence upon the mental models. But language, especially natural language, is another way to match up the concepts contained in A and B's mental models. As Roy Hagman [Hagman 1982] points out, language is technology that lets people use air-vibration artifacts to rearrange other people's mental models. This natural language–mediated interaction is what enables the informant and the analyst to reach consensus in negotiating the meaning of the text generated during knowledge acquisition. It is also through language that public concepts, as described in section 2.2.1, come about.

2.4.2 Ontologies

The end product of the knowledge acquisition process is a formalized partial model of the domain of discourse, as defined by an *ontology* and described by conceptual graphs. What is an ontology? Sowa states:

> An ontology [is] a catalog of modes of existence. [Sowa 1984, pg. 361]
>
> The ultimate goal [of conceptual analysis] is a precise, formalizable catalog of concepts, relations, facts, and principles. With conceptual graphs, the goal is to determine type labels, canonical graphs, schemata, and laws of the world that define some body of knowledge or domain of discourse. ... The result of the analysis is an *ontology* for a possible world—a catalog of everything that makes up that world, how it's put together, and how it works. [Sowa 1984, pg. 294]

This passage clearly defines an ontology as a catalog, or list, with a structure. It is a structured set of conceptual graphs that forms the conceptual model for a domain of discourse and, in the knowledge acquisition context, serves as the blueprint for the design of a knowledge base.

There are several different ways of structuring an ontology. One of the most useful distinctions is to separate out the basic building blocks in a list that we have been calling the entities list from the catalog of more complex structures such as canonical graphs, schemata, rules, statements, laws, and propositions. We may distinguish several levels in the ontology. The basic level is the entities list, a list of names with type designators. This contains names of concepts, *i.e.,* names of conceptual primitives, conceptual composites, and surrogates for referent entities. The formalization stage of *concept-oriented knowledge acquisition* starts with the compilation of this basic entities list. This phase concentrates on the basic building blocks. For example, consider the list in Table 2. The character strings [FISH], [FISH: *], and "fish" are names of entities. The type of the entities is given by the type designator from the meaning triangle, which is also a character string.

Higher levels of the ontology contain the canonical graphs for concepts, schemata associated with concepts, and various rules, statements, laws, and constraints in conceptual graphs form. Compiling the higher level sections of the ontology is the task of *proposition-oriented knowledge acquisition.*[3]

An ontology is not the same as an inventory. An ontology, describing a domain of discourse, contains what exists in the domain of discourse and *might* exist in the object domain. An inventory, dependent on time, place, situation, and the actual state of affairs of the world, lists what actually

[3] This may be contrasted with the more simplistic rule-oriented knowledge acquisition emphasized by, for example, Waterman [Waterman 1986] for production-rule expert systems.

Table 2: Example of entities and their types.

Names of entities	Type designator
[FISH]	concept
[CAR]	concept
[FISH: *]	referent
[CAR: #Mazda2358]	referent
"fish"	word
"car"	word

does exist in the object domain. We can look upon the inventory list as an instantiation of (some of) the basic entities of the ontology in a specific object domain.

2.4.3 Distinguishing terms from concepts

Terminology is a major stumbling block in the harmonization of mental models. Apart from any difficulties associated with the use of ordinary, common natural language as the medium of communication, the expert informant will necessarily use technical terms to express her interpretation of the domain knowledge. These terms are to be attributed the special meaning that the informant has in mind. The analyst should refrain from the temptation to decide what these terms 'should' mean or to reject the terminology altogether; that is, he must not impose his own normative standards. How closely the informant's mental models correspond to the object domain, including abstract entities, is not for the analyst to decide. The analyst should not think that he 'knows better'. Presumably, the informant is designated as an expert because, through years of practice, she has proved that there is a close correspondence between her mental models and the abstract body of knowledge. In a career of work and social interaction in her discipline, she has demonstrated her ability to harmonize mental models in her domain of expertise. The analyst has to create mental models of the informant's mental models that accurately reflect the semantic objects [Hirst 1987] associated with the terms that she is using.

Terms must be distinguished from concepts. There can be several terms for the same concept, and many concepts referred to by the same term. It is dangerous to assume that because two agents use the same terminology, they have the same concepts in mind. The connection between terms and concepts is best illustrated with an *agent-centered meaning triangle* (Figure 4). The informant-centered triangle and the analyst-centered triangle need to have the same vocabulary on the word vertex. But even if the vocabulary matches, the job of the analyst remains to ensure that the concept vertices also correspond closely.

2.5 Conceptual analysis defined

The term *conceptual analysis* has been used in the literature in widely differing senses. For our purposes, conceptual analysis is the forming of concepts and the associating of these concepts with words and referents, and studying the internal structure of a set of concepts with the objective of

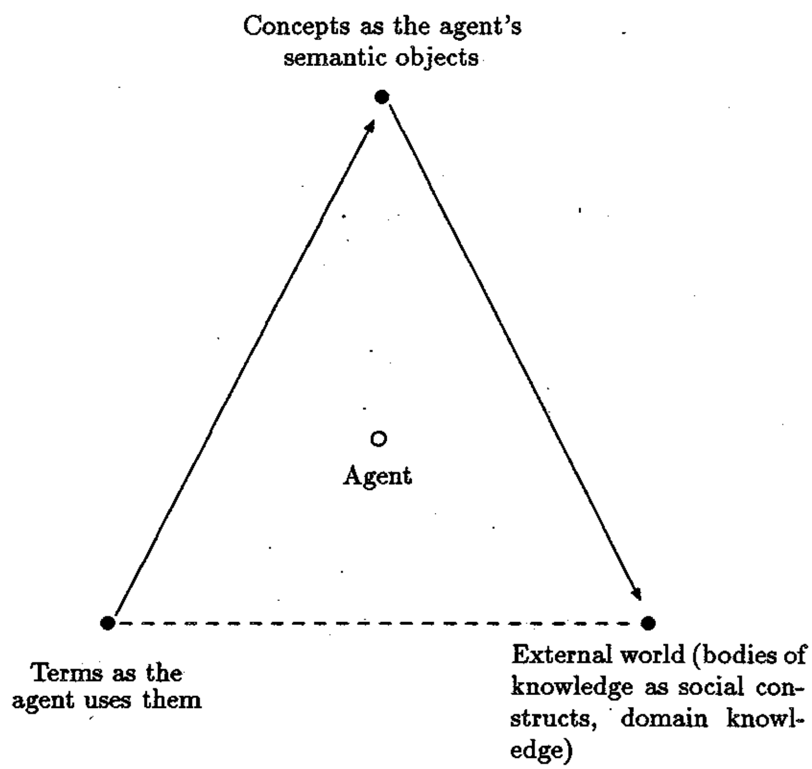

Figure 4: An agent-centered meaning triangle.

establishing an ontology. It is a technique used in both the elicitation and explication stages of the knowledge acquisition process. Sowa says:

> Conceptual analysis is the work of philosophers, lawyers, lexicographers, systems analysts, and database administrators. Philosophers have been doing conceptual analysis ever since Socrates taught Plato how to analyze JUSTICE; lawyers do it whenever they draw fine distinctions in arguing a point of law; lexicographers do it in bulk quantities when they compile dictionaries; and systems analysts and database administrators do it when they translate English specifications into a system design. Conceptual analysis is essential for giving content to the empty boxes and circles of conceptual graphs.
>
> Every discipline that uses conceptual analysis gives it a different name. In the computer field, the most common names are *systems analysis, enterprise analysis,* and *knowledge engineering.* [Sowa 1984, pg. 294]

In the knowledge acquisition process that we are discussing in this chapter, we are mainly dealing with conceptual knowledge about abstract entities. Because abstract entities are social constructs, one finds out about them by talking to people. At this elicitation stage, the task is to match concepts to lexemes. First attempts at matching are usually discouraging; informant and analyst will have different concepts in mind. But during the discussion, they will usually achieve greater harmony.

At the knowledge explication stage, we further examine the concepts that have been singled out for special attention. Examining a concept is not like looking at a chair or a lamp. We cannot pick up a concept in our hands to examine it. If we want information about a concept, we have to assume that it is in somebody's mental model, or it is a social construct, and talk to somebody about it. We can also engage in introspection, examining our own mental models.

How exactly does one examine a concept? Essentially, by thinking and talking about it. We unpack the 'internal structure' of the concept and look at how it is related to other concepts. The natural language dialogue between analyst and informant serves as a probe to determine what concepts exist, and what these concepts are like.

Conceptual analysis falls into several different types, which we discuss in section 4.

2.6 The knowledge acquisition process

We are now in a position to describe knowledge acquisition more formally. As we said above, the knowledge acquisition process that we are dealing with in this chapter is based upon interaction between the informant and the analyst. It is mediated by natural language, because the analyst and the informant conduct a dialogue in a natural language such as English or French, and the transcript of this dialogue, and some possible additional documents, are written in the same natural language. But an important component of the discourse consists of conceptual graphs, which make the formalized record of the knowledge that is being acquired during this process. This body of formalized knowledge is built up in stages. At each stage it forms a version of the partial model of the domain of discourse.

The fine structure of the knowledge acquisition process can become quite complex. The analyst and informant talk with one another; the text of their conversation is tape-recorded, transcribed, analyzed, and diagrammed. The analyst and informant change each other's mind. They influence each other's thinking by trying to understand what the other is saying. The analyst, in particular, builds mental models of what he construes to be the analyst's special knowledge that is relevant to the system to be built. The knowledge acquisition is taking place within a complex; it is not performed merely by the analyst. Symbolically, we can say that knowledge acquisition is based upon the triple

$\langle$informant, text, analyst$\rangle$

(see Figure 5). It is the whole complex, symbolized by the triple, that is doing the knowledge acquisition and recording. The conceptual graphs are part of the common text that is being built up. They form the output of the process. The input is the two agents, together with the information they possess, plus any necessary documents which become part of the common text. The interaction, however, goes beyond textual communication; it includes the generating and revising of mental models.

The eventual goal is the conceptual design of a knowledge base. The process occurs in stages. At each stage k, there exists version k of the text, including version k of a partial model in conceptual graphs form. The final, 'releasable' version of the partial model of the domain of discourse contains knowledge 'recorded' with conceptual graphs.

To summarize, the knowledge acquisition situation that we are examining is an ***informant-analyst interaction–based, natural language–mediated, knowledge acquisition process.***

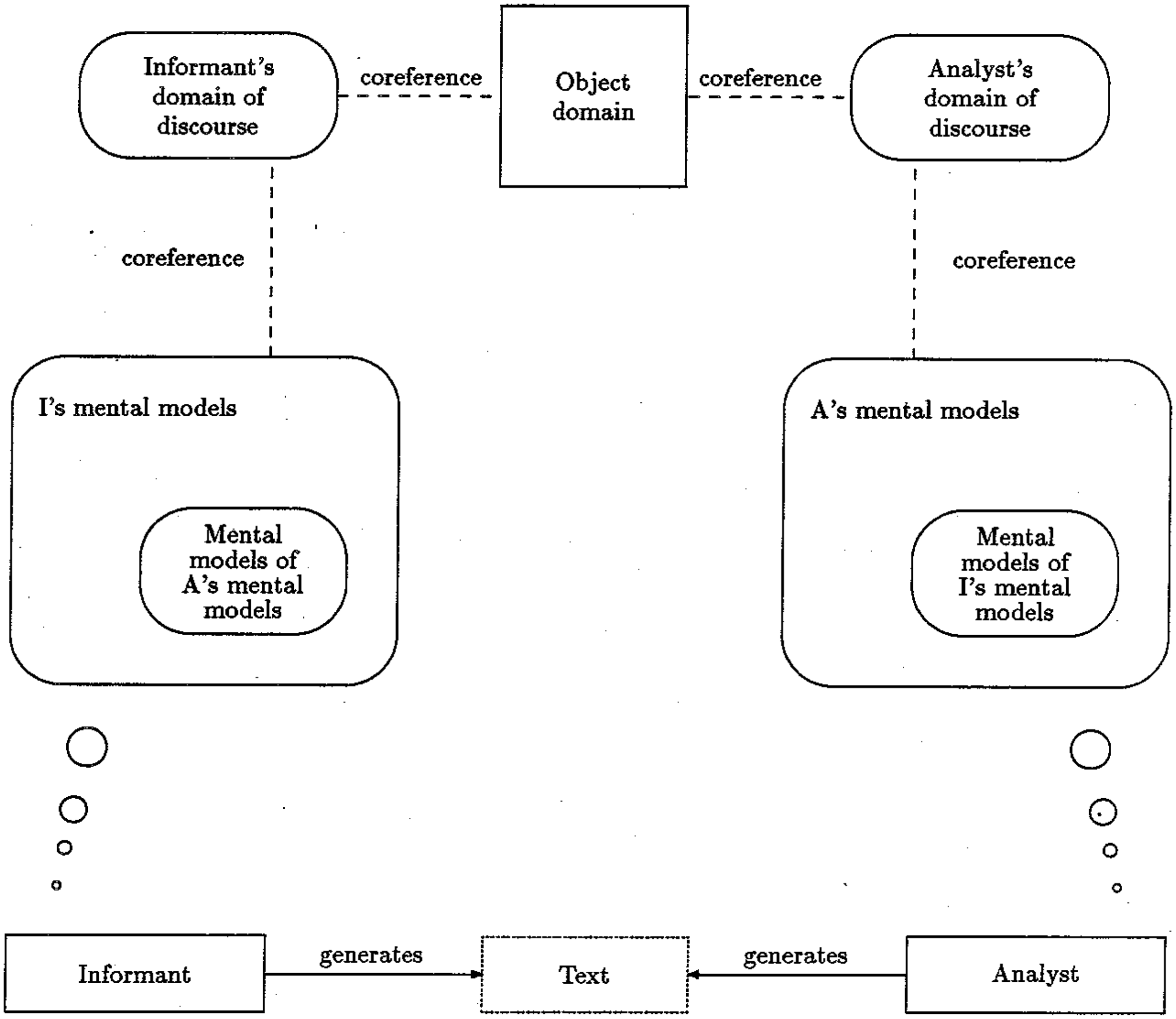

Figure 5: Entities in informant-analyst interaction–based natural language–mediated knowledge acquisition.

3 Difficulties in knowledge elicitation

Let us look at some examples to illustrate the process of conceptual analysis and to show the sources of some of the difficulties that we might encounter in actual knowledge acquisition work.

Suppose we want to build a knowledge-based system that is capable of classifying animals into various categories. We as analysts can ask an expert in animal taxonomy the question: "Please, madam, how does one tell birds apart from the other animals?" Let us suppose that the expert responds, "Birds fly", and suddenly vanishes, so that she cannot answer any more questions.

We are left to ponder this cryptic remark, *Birds fly*, which we can diagram as a graph with two concepts and one conceptual relation:

[BIRD] ← (AGENT) ← [FLY]

This is some knowledge, but not much, because we would like to have a bit more information about this activity of flying that birds do. We can ask ourselves, or perhaps a friend, some commonsense questions such as these:

Us: "Do birds fly?"

Informant: "Yes."

Us: "Always?"

Informant: "No. In general, birds fly; or perhaps typically, birds fly; or perhaps, some birds fly most of the time; or birds, in general, have the capacity to fly, unless they have broken wings."

Us: "Do two-day-old chicks fly?"

Informant: "No."

Us: "Do dogs fly?"

Informant: "No."

Us: "Do dogs in airplanes fly?"

Informant: "Yes."

Us: "Do I fly?"

Informant: "No."

Us: "Do I fly when I am flying to Montreal?"

Informant: "Yes."

Us: "Am I sitting when I am flying to Montreal?"

Informant: "Yes." [So apparently it is possible to sit and fly at the same time.]

Us: "Does the pilot fly?"

Informant: "Yes." [But the pilot is flying in a very different sense from that in which the passenger is flying.]

Us: "Do flying squirrels fly?"

Informant: "Hard to say." [Flying squirrels don't fly; they parachute. But then, how does one tell the difference between the flying of a glider and the flying squirrel's non-flying?]

And so on. It is clear that knowledge elicitation on the basis of this kind of question-and-answer session is quite difficult. Where exactly is the difficulty?

We could try a few guesses. We could say that the difficulty concerning the expressions *always*, *in general*, and *typically* has to do with the fact that these are qualifiers. In logic we have a good theory of quantifiers, but no usable theory of qualifiers. Alternatively, we could say that the cryptic remark *Birds fly* suffers from vagueness. In reference to young chicks and adult birds there is no account given within the text concerning the age structure of birds and the time dimension of their capabilities. Going further, we could claim (and this is a much-used cliché about natural language) that the word *fly* is ambiguous, having several different senses in which it can be used. If it is an ambiguous word, that is it 'has' many meanings, or we can 'attribute' many meanings to it, then we can blame our problems on an inadequate dictionary.

In conceptual analysis we cannot take the easy way out. Let us conclude, as indicated by the above analysis, that the word *fly* maps into elements of the set $\{[\mathrm{FLY}_k], k \in I\}$, and try to construct a set of schemata for each element of this set. So, for example, for the use of *fly* in the sentence *The pilot flies* we would map *fly* into $[\mathrm{FLY}_1]$ and the concept $[\mathrm{FLY}_1]$ would contain subgraphs regarding the guiding of a transport vehicle. For the passenger, the concept would be $[\mathrm{FLY}_2]$ and would contain a subgraph describing passive sitting in a vehicle moving through the air. The process of generating conceptual graphs for the various concepts 'behind' the different uses of the word *fly* relevant to our knowledge domain is what we mean by conceptual analysis.

Let us now look at a different, and somewhat more difficult, example. Suppose we are asked to explain to a computer-based knowledge system the meaning of the sentence

> The sky is blue.

On the basis of grammatical and some semantic analysis the machine draws some conclusions and answers

> You mean to say that there is an entity called *sky*, and it has the attribute color, and the color happens to be blue?

And we answer "Yes, that is what we mean." After a little while, the machine starts asking questions:

Machine: "Where is the sky?"

Us: "Up there." [*We point.*]

Machine: "How far up there?"

Us: "Quite far."

Machine: "What is the sky for?"

Us: "Well, birds fly in it."

Machine: "I already know that birds fly through the air. I hypothesize that *sky* means *air*. So air is blue, and one can breathe sky, right?"

Well, we got caught in a conceptual trap. Although we talk about *the sky*, reify it as an entity, and regard it as 'something out there', actually a different mental model is being activated. The mental model is a piece of cultural heritage that pictures the sky as a fairly thick belljar of blue, evanescent substance that covers the surface of a basically flat-looking earth. Our talk about the blue sky refers to this mental model, not to the item in the inventory of the object domain of the real world.

We have and talk about mental models that have very little to do with physical reality, but they work nevertheless. The mental model we have that incorporates the concept [SKY] is quite explicit and widely known. It is based on an older cosmology. As the next example will show, concepts developed as part of an older worldview are sometimes implicitly packaged into the language like fossils. Consider the sentence:

The wind is blowing.[4]

We may now ask:

What does the wind do when it is not blowing?

This is a difficult question. Is it just sitting there, or is there a wind when it is not blowing? We could say that the wind is 'really' air current, but we don't say that

The air current is blowing.

We also do not say that

The air current is currenting.

This linguistic fossil, embedded in English syntax, harkens back to the older worldview of the four winds blowing from the four corners of the earth.

The lesson: It becomes clear that conceptual analysis is like archeology—through patient digging we uncover artifacts. These artifacts are concepts, and these concepts may be components of some very strange-looking mental models. The job of the analyst is to uncover the mental models of the informant, and to separate the concepts within the mental models from the referents 'out there' in the world that they may or may not be referring to. The main tool for drawing distinctions of this type is the meaning triangle.

[4] We are indebted to Francis Horvath for this example.

4 Conceptual analysis

Let us now briefly summarize our discussion up to this point about conceptual analysis. Conceptual analysis is the process of generating a set of conceptual graphs that model mental models, using the information given about a knowledge domain. This information usually comes through natural language text, either oral or written. In a slightly more expanded version, we can describe conceptual analysis as a process for generating sets of conceptual graphs for text, named concepts, and mental models. The set of conceptual graphs is to be itself regarded as a partial model for the domain of discourse, or for the meaning of a piece of text.

At one level, we can regard conceptual analysis as a technique for formalizing the different senses in which a word may be used. This is a common and useful lexicographic approach, and one which tries to cope with one of the central problems of natural language–mediated knowledge acquisition work. A formal approach—one that we are not adopting here—would take a sentence like *Birds fly* and would put down logical formulas such as:

$$\forall x(B(x) \Rightarrow F(x))$$
$$\exists x(B(x) \wedge F(x))$$

assuming in the latter case that x is a 'typical' x, that is, the kind of x that we would 'normally' encounter. The issue, however, is not so much the shape of these formulas, but rather the choice of the various F_i in the set $\{F_i, i \in I\}$ and the relationship of this choice to the shape of the formulas. For example, consider the sentence:

> Some birds fly$_1$, but not a single cow is capable of flight$_1$ although all of them can fly$_2$ according to some general notion of flight$_2$, as in flying$_3$ by airplane.

The various subscripts on the forms of the word *fly* correspond to the subscripts on the F_i's.

All F_i's may have some commonality. If so, this should be shown by conceptual graphs. Or only some subset of the F_i's may have commonalities. Commonality may be defined in conceptual graphs by subgraphs. If graphs G_1 and G_2 both contain the subgraph G', then G' may be the commonality of G_1 and G_2.

Is conceptual analysis a simple translation problem? There is certainly a temptation to look upon it as such. We can identify the task as translating the text of statements into conceptual graphs or logical statements. But this is misleading. Conceptual graphs should be looked upon as models of concepts—models of mental models. Also we should note that *no* translation is "simple", least of all the translation that takes natural language expressions and maps them into logical expressions; it requires a great deal of analysis.

Conceptual analysis is not so much a task in translation as a task in meaning generation or meaning representation. *Birds fly* can certainly be represented in one of these forms:

$$\forall x(B(x) \Rightarrow F(x))$$

[BIRD] ← (AGENT) ← [FLY]

but the main issue is one of understanding and meaning generation. Let us define understanding.

$$S \rightarrow \boxed{\text{Analyst}} \rightarrow \langle S, \textit{meaning} \rangle$$

$$\langle S, \textit{meaning} \rangle \rightarrow \boxed{\text{Informant}} \rightarrow S_1$$

(a)

$$S_1 \rightarrow \boxed{\text{Analyst}} \rightarrow \langle S_1, \textit{meaning} \rangle$$

(b)

Figure 6: Understanding as an iterative process.

Definition. An expression S is ***understood*** if a conceptual graph or logical formula or other representation of meaning can be formulated for S.

We can picture this as a process, with S as the input, the ***interpreter*** as the process, and the ordered pair $\langle S, G \rangle$ as the output, where G is a set of conceptual graphs:

$S \rightarrow$ Interpreter $\rightarrow \langle S, G \rangle$

Definition. S is understood ***unambiguously*** if G contains only one graph.

A paradoxical consequence of this perspective is that interpreters with little imagination and restricted powers of meaning determination have a better chance of coming up with unambiguous meanings. A good interpreter does not find too many meanings, only the ones required. The 'correctness' of the meaning generated is judged in a dialogue between informant and analyst. In the interaction between them, conceptual graphs can be regarded as expressions in a language. The exchange can be diagrammed as in Figure 6a. From the vantage point of the informant, the input $\langle S, \textit{meaning} \rangle$ is a character string. The output S_1 can be construed as a paraphrase of S, or a comment, or a report on the state of mind of the informant such as "I am reasonably satisfied that $\langle S, \textit{meaning} \rangle$ is correct". Then we have the situation of Figure 6b, repeating the process on S_1.

4.1 Different types of conceptual analysis

There are several different types of conceptual analysis, depending on whether one is building conceptual models of mental models, generating conceptual models for the concepts 'behind' a text, or analyzing the schemata associated with a well-known, public concept. For conceptual analysis to be useful in practical knowledge explication, it is important to recognize this diversity of techniques. For our purposes we distinguish three:

- Conceptual analysis of text.
- Conceptual analysis of named concepts.
- Conceptual analysis of mental models.

The difference between the three types is best appreciated by relating them to the knowledge acquisition process that we described above in Section 2.6. In the conceptual analysis of text, the informant is not present. In the conceptual analysis of public, named concepts such as [MARRIAGE], [CORPORATION], or [RIEMANN-INTEGRAL], the starting point of the analysis is the name of the concept. In the conceptual analysis of mental models, we are actually 'debriefing' the informant, as well as introspectively analyzing our own mental models.

4.2 The conceptual analysis of text

In the conceptual analysis of text, the analyst is given a piece of text, and perhaps some other aids, and his job is to generate conceptual structures based on the text. In this stark formulation, the task may yield some highly idiosyncratic results.

In Schankian terms, Birnbaum and Selfridge define conceptual analysis—by which they really mean conceptual analysis of text—as

> the process of mapping natural language text into the memory structures that represent the meaning of that text. [Birnbaum and Selfridge 1981, pg. 318]

This view requires some modification. The view is that text has meaning and that meaning is extracted. In fact, text is received by someone, such as the knowledge acquisition worker, and it is this person that *attributes or ascribes* meaning to the text. This attribution is not a totally arbitrary process. In fact, in the case of some texts, there will be so little latitude in the range of meanings that may be attributed that one might as well say that the text 'has' meaning. But we want to be open to the attributive role of the knowledge acquisition analyst as he receives the elicited text from the informant. The knowledge is not *in* the text. It is in the mental models of the informant, and is recreated by the analyst.

According to the mental models hypothesis, there is no knowledge without a knower [Regoczei]. The analyst, upon reading the text, may or may not generate mental models. If he does, the set of conceptual structures so generated is the 'meaning' of the text for the analyst. The conceptual structures can be diagrammed with conceptual graphs. These graphs are publicly examinable, and should be examined by the informant or others involved (see below) to guard against undesirable idiosyncratic interpretations by the analyst.

Even on the basis of this simple, schematic look, we can see an intricate complexity of interactions and intermediate stages. First, we have a complex cast of characters:

- The author of the text (the informant).
- The text itself.
- The reader that the author had in mind.
- The editor of the text (if any).
- The reader of the text.
- The author that the reader has in mind.

- The person who examines the reader/analyst's conceptual analysis and provides feedback.

The analyst's procedure is the following:

1. Break the text into pieces (without losing meaning).
2. Associate concepts with the fragments of text (and hope they are suitable).
3. Analyze the concepts, explicating connections, and evaluating the suitability of the text fragments and concepts.
4. Repeat as necessary.

In other words, the analyst reads the text, tries to form mental models, does a conceptual analysis of his own mental models (introspection), and perhaps some conceptual analysis of named concepts (as when one says, "I wonder what the author means by this term"), somehow gets his jumbled thoughts organized perhaps with the aid of some pre-diagramming techniques, and finally produces neatly-drawn conceptual graphs for public examination. This all seems a bit too difficult. There has to be some way of catching the mistakes that the analyst will inevitably make. To keep things manageable, and to keep sight of the pragmatic objectives, he should strive to establish the ontology and inventory of concepts in a dialogue with an informant, and not merely rely upon text [Regoczei and Plantinga 1988]. This involves the conceptual analysis of the mental models of an informant (who may or may not be the same person as the author of the text), the generation of additional text through dialogue, and careful version control of the new text.

To illustrate these points, let us look at some examples from the OCRA car registration problem [Appendix XXXX]. Let us suppose that there is an analyst who is asked to do conceptual analysis of text upon the car registration description. He comes to the sentence [pg. XXXX]:

Each car is of a particular model.

He marks up the text and assembles a list of content words and phrases together with their associated concepts:

Contentive	**Concept**
car	[CAR]
each car	[CAR: *]
model	[MODEL]
particular model	[MODEL: *]

So the conceptual graph for the sentence would be:

[CAR: *] → (ATTR) → [MODEL: *]

Could the author of the text have meant that each car is of a particular *unique* model, wonders the analyst. If such thoughts do not occur to him, if the analyst is unimaginative, few problems will be noticed. It is easy enough as long as the analyst can avoid asking questions such as "What is a car?" The analyst does not know, nor does he need to know. Otherwise, the questions are endless.

"Is a taxi a car? Is a schoolbus a car? If not, why not?" For pragmatic reasons, [CAR] can be treated as an undefined semantic primitive relative to a specific domain of discourse. But problems arise when terms such as *car* and *automobile* occur in the same piece of text. Then the analyst has to switch from conceptual analysis of text to conceptual analysis of named concepts, and try to find further knowledge about the interconnection of the concepts [CAR] and [AUTOMOBILE]. Perhaps in this case one would like to make the 'helpful' suggestion to the analyst that he should use his own 'common sense' and jump to the conclusion that the two concept types are defined by essentially identical conceptual graphs. One should resist this temptation. In both jurisprudence and systems analysis, two concepts 'obviously' identical to one person may be associated with very different conceptual structures in the mental models of another person.

To cite an even stronger example from the car registration problem, which, after all, is about rules and regulations and, hence, would have a substantial amount of legal verbiage behind it, let us look at two terms occurring in the text: *Oz* and *the Kingdom of Oz*. How many analysts of a legal bent of mind would argue that these two terms refer to the same concept? In this case, a simple conceptual analysis of text is not enough. Both conceptual analysis of named concepts, and conceptual analysis of mental models is called for, and preferably in a group dialogue, because public concepts are *social* constructs.

To summarize the most important points: The objective of conceptual analysis of text is to associate the text with a set of conceptual graphs, which could be thought of as the 'meaning' of the text or of a particular sentence. Therefore, conceptual analysis of text, sentence by sentence, is a mapping that takes the character string S into a conceptual graph $[S_k]$, where the index k indicates that several different interpretations are possible. Thus S is mapped into the ordered pair $\langle S, [S_k] \rangle$. The output of the conceptual analysis of the text is a set of conceptual graphs that are subsequently stored in a knowledge base.

The final point to be made is that since concepts ultimately reside in the mental models of individuals, the conceptual analysis of text should, where possible, be replaced by an informant-analyst discourse, with further explanatory text being generated as a result of this discourse. Such a procedure is preferable to the analyst working in an introspective manner, because, as we said, public concepts are social constructs.

4.3 Conceptual analysis of named concepts

Named concepts such as those of marriage, algorithm, and integer are the objects of study when philosophers talk about conceptual analysis.[5] The philosophers' techniques for conceptual analysis of named concepts have mostly been ineffective; their approach is too unsystematic and the results are expressed in prose that compounds rather than reduces the problem.

A notable exception to this practice was Aaron Sloman's attempt [Sloman 1978] to set out an operationally effective method for conceptual analysis. Although he did not realize it at the time, he was restricting his attention to the conceptual analysis of named concepts, following the established philosophical tradition. His approach was not well suited for knowledge explication

[5] Words do not necessarily have concepts behind them. The names often are just empty labels. In this case, conceptual analysis is simply pushing empty tokens around, uncovering nothing of substance—just rhetoric.

and for the kind of methodologies that we are attempting to construct here; the way Sloman phrased the technique made it difficult to apply in practice. Some of the difficulties were these:

- Sloman did not draw a sharp distinction between a concept and its name, thus introducing further problems into the analysis.
- The results of the analysis were presented in natural language prose. Thus the problem of conceptual analysis of named concepts burdens us, as a result, with the task of conceptual analysis of text.
- Sloman was not aware of the problem that since named concepts have names, this name is a lexeme that can have many different senses that may have nothing to do with the original concept. The problem is not so much one of combinatorial explosion, but the danger of going on a wild goose chase by selecting the wrong sense of the lexeme.
- Sloman has no criteria for terminating the analysis.

However, a major improvement was effected in Sloman's approach by Sowa [Sowa 1984, pp. 297–298], who condensed and sharpened some of the techniques that Sloman was suggesting, and also divided the method into a clearly distinguishable analysis phase and testing phase. Difficulties that are found in the testing phase cause the process to cycle back to the analysis phase. In this form, the Sowa-Sloman algorithm is a powerful technique for the conceptual analysis of named concepts and should be an integral part of any knowledge acquisition project. We now briefly summarize the algorithm.

4.3.1 The Sowa-Sloman algorithm

The Sowa-Sloman algorithm has the following analysis components [Sowa 1984, pg. 297]:

1. *Instances:* Collect evidence of instances of the use of the concept.
2. *Type hierarchy:* Classify the concept according to a taxonomy.
3. *Canonical graphs:* Construct the appropriate graphs.
4. *Definitions:* Collect dictionary definitions, other definitions, type definitions, schemata definitions, prototypes.
5. *Schemata:* Collect examples of families of uses.

These five components result in a body of text and a set of conceptual graphs. To test the adequacy of the coverage, the analyst carries out a number of thought-experiments. These experiments are usually thought-experiments only for reasons of constraints that the analyst works under, such as time, money, and resources. The tests are these:

1. *Teaching:* How would the concept be taught to a child, or someone not very familiar with the subject?

2. *Operational tests:* What operational tests would be required to test the truth or falsity of statements about the concept?

3. *Story telling:* Imagine fictional situations that reveal the scope and ramifications of the concept.

4. *Computer simulation:* "For knowledge-based systems, the analysis must enable a computer to converse with people in normal English. Is it missing any aspect that would cause a computer to use the concept incorrectly?" [Sowa 1984, pg. 298]

If these tests uncover shortcomings in the description of the concept in the form of text or conceptual graphs, the analyst cycles back to one of the appropriate analysis components for correction.

Because named concepts are public, the analysis should be conducted in an informant-analyst context. As with the conceptual analysis of text, trying to do it on an introspective basis yields idiosyncratic, defective, and controversial results.

It should also be noted that although the Sowa-Sloman algorithm concerns itself with the conceptual analysis of named concepts, the input to the process comes in the form of text. The name of the named concept is text and so are the synonyms for the name. Any descriptions of the mental models associated with the name of the concept either by the informant or the analyst should be recorded as text. For the sake of completeness, this body of text should as a further control technique be subsequently conceptually analyzed as text. While the analysis of text, of named concepts, and of mental models are interdependent, the analyst should keep in mind which type of analysis he is doing at any particular moment.

Why should we be so thorough in our analysis? Mainly because much expertise and much knowledge is based upon tacit knowledge. There are tacit assumptions behind all domains of discourse. This tacit context is very difficult to reveal. The Sowa-Sloman algorithm provides an effective way of uncovering, of explicating, the underlying conceptual richness of a domain of discourse.

4.4 The conceptual analysis of mental models

The conceptual analysis of mental models seems easy, but is actually quite difficult. It tries to determine what another person has in mind. To look at the concepts that constitute one's own mental models seems easier than trying to 'read another person's mind', or to 'hammer out a group consensus'. In fact, because of the numerous tacit assumptions that we use on a regular, normal basis, the concepts even within our own mental models are available to us for observation only to a limited extent. Much of our thinking is subconscious and non-monitorable. (This is one of the reasons why psychotherapy is so difficult, for example.)

We now present an algorithmic way of analyzing mental models, with natural language as the medium of interaction. This technique, described in greater detail in [Regoczei and Plantinga 1988], is based upon a systematic version-by-version harmonization of the informant's and analyst's mental models. It should be emphasized that version control must be enforced, as we note below at the relevant steps. This technique tries to come to grips with the paradox that although we are after concepts, we have to start with words. This is why we stated above that the knowledge acquisition process is performed by not a single agent but a triple, ⟨informant, text, analyst⟩. Accordingly, we start with whatever preliminary text is available.

1. *Establish a text.* The text should be in a permanent, publicly examinable form such as a written document, voice tape, or video tape. Text is not to be altered without changing the version number.

2. *Select the content words.* One should pay special attention to terms that are phrasal lexemes or fragments of noun phrases.

3. *Produce a lexeme-to-concept mapping.* This is done either through a conceptual lexicon, if one exists, or by free association. Typically, several different but related concepts would be associated with each term, and the analyst's concepts would differ from those of the informant.

4. *Diagram the concepts.* Schemata are drawn to show how concepts under examination are related and linked to other concepts. Semantic primitives (relative to a given domain of discourse) are diagrammed as single-concept nodes. This produces a particular formal version of the partial model of the domain of discourse. Version control is to be exercised on the partial model.

5. *Test the concepts.* Ask the informant to express her opinion about the appropriateness of the graphs.

6. *Model mental models.* Since the analyst is trying to form mental models of the informant's mental models, at this stage a frank discussion is encouraged about what the informant 'really has in mind' and how well the analyst's thinking matches her thinking. This discussion yields text, which should be tape-recorded and cycled back to step 1, perhaps to be part of the next version of the text.

5 Practical difficulties

Associating concepts with words is a difficult activity. Describing concepts in words is even more difficult. Yet associating concepts with words, and describing concepts using a symbol system such as conceptual graphs is precisely what lies at the core of the informant-text-analyst–based, natural language–mediated knowledge acquisition process. In naturally occurring text, and in what informants actually say (as opposed to what they think they are saying, or what they should be saying, or what they would like to be saying), the analyst will encounter a number of practical difficulties. Much of what people utter is ill-formed text—defective both syntactically and semantically.

Natural language, like all other symbol systems, has limitations in its expressive power. Natural language 'in the raw'—that is, what people are actually saying and the unedited text that they actually write down—can be surprisingly poor in grammar and in expressive capability. People are often shocked when their conversation is tape-recorded and played back to them. They may exclaim, "I did not say that! What I really meant was ..." Of course, they really did say what the tape recorder recorded. The effect is even more pronounced when the tapes are transcribed. In written text, we are conditioned to expect highly polished, edited prose. Oral natural language 'in the raw', when transcribed, seems to contradict all theories of grammar and semantics. In fact, most linguists prefer not to deal with natural natural language at all; they prefer to look at competence rather than evidence of actual performance.

In brief, the analyst encounters difficulties in a natural language–mediated interaction. Some of these difficulties are ascribable to the ill-formed nature of the text that is actually used in everyday interaction. The main problem is that the natural language expressions do not give enough clues, or give misleading clues, about the mental models behind them. Almost all of the difficulties are attributable to the fact that the mental models of the informant are not directly accessible to the analyst. The analyst wants knowledge, but all he gets are words. To reconstruct knowledge on the basis of textual input is difficult. Hence the occasional, slightly exasperated request, "Why don't you just say what you mean?!" But it's not so easy for an informant to figure out what she means, and, having done that, express it in words.

The analyst has to be aware of the fact that many of the difficulties cannot be avoided in practice. Non-linguistic difficulties are also common. For example, mental models usually have strong components of the 'folk models' variety. We have already described in Section 3, both *The sky is blue* and *The wind blows* as examples of this type. Likewise, describing 'naïve physics' in formal terms is extremely difficult because we have only the slightest glimpse of the intuitive, naïve, folk models that underlie our everyday actions in interacting with the physical world (see, for example, [Gentner and Stevens 1983]). We should also mention in passing category mistakes, metaphors, figurative language, vagueness, discretization, overprecision, missing Fillmore case-frames, and various kinds of canonical-graph violations.

6 Putting it all together

Let us now return to the theme that started this chapter. Knowledge acquisition is the initial phase of knowledge system development. But knowledge systems are application software, and since all application systems contain some world knowledge, knowledge systems may not differ all that radically from the more traditional kinds of application software.

Characterizing knowledge systems as ones that are programmed in Lisp or Prolog seems to confuse the programming language of implementation with the nature of the system. Characterizing knowledge systems as rule-based systems (cf [Waterman 1986]) also misses the mark. Rule-based systems can be looked upon as ones developed with a particular programming style, or a particular way of approaching software architecture. In the software development lifecycle, these are all decisions that are taken fairly far 'downstream'.

Even characterizing knowledge systems as ones that have a 'knowledge base' is difficult to operationalize. How do we tell a 'knowledge base' from a database that has a particularly rich semantic structure. In fact, there is a confluence of programming, database, and artificial intelligence research under the banner of 'conceptual modeling' that tries to transcend distinctions that are based on no more than implementation details [Brodie *et al* 1984,Brodie and Mylopoulos 1986].

We propose to define a knowledge system as one in which the knowledge acquisition process dominates the system development effort. We mentioned at the beginning of the chapter that knowledge acquisition is the 'front end' of the process. This is correct, but requires further elaboration. Some of this so-called 'front end' work is actually spread over several stages of activity. To illustrate this point, let us look at two traditional models of the system development lifecycle in Table 3.

How can knowledge acquisition and conceptual analysis fit into either of these traditional models of the systems development lifecycle? For example, in activity 3 of model 2, 'define user requirements', we note that there is an overlap with some of the most crucial stages of the lifecycle. User requirements definitions, the result of an activity that is basically knowledge acquisition, in fact continue to be refined while the design and even construction of the new system proceed.

Also, activity 1 of model 1, 'system planning', involves knowledge acquisition and conceptual analysis, but the output is highly edited formal prose, not conceptual graphs. The systems analysis activity would cover the design of the knowledge base. But the design for the knowledge base is the equivalent of a conceptual schema. With some fourth-generation languages and database management systems, specifying a conceptual schema is as good as implementing the database. This database is 'empty'; to complete system implementation, one fills the database with records obtained from the inventory. But finalizing the small details of the knowledge in the knowledge base would be under activity 4, 'system implementation'. It is only at this stage that the analyst can sit down with the informant to 'debug' the models of the informant's mental models, of the domain of discourse, and of the object domain. Therefore, in the case of fourth-generation languages and expert system shells, at times no sharp distinction can be drawn between knowledge acquisition, system design, and software implementation.

This conclusion has some very interesting consequences. Suppose we had an expert system shell that accepted conceptual graphs as input. The graphs produced by knowledge acquisition

Table 3: Two traditional models of the system development lifecycle.

Model 1

1. System planning (output: user requirements).
2. System analysis (output: system specifications).
3. System design (output: technical design specifications).
4. System implementation (output: usable software).
5. System maintenance.

Model 2

1. Survey the situation.
2. Study the current system.
3. Define user requirements.
4. Evaluate alternative solutions.
5. Select new computer equipment and software (if necessary).
6. Design the new system.
7. Construct the new system.
8. Deliver the new system.

could be entered directly into the shell. As soon as the knowledge acquisition phase were complete, we would have a specification for the knowledge base, and since the expert system shell accepts conceptual graphs, these specifications become executable. Conceptual analysis and knowledge acquisition based on conceptual graphs would become a way of preparing executable specifications for knowledge systems.

Acknowledgements

We are grateful to John Burbidge, Roy Hagman, and Ed Plantinga for discussions. Preparation of this chapter was supported by a grant to the first author from the Trent University Committee on Research, and to the second author from the Natural Sciences and Engineering Research Council of Canada.

References

[Birnbaum and Selfridge 1981] Birnbaum, Lawrence and Selfridge, Mallory (1981). "Conceptual analysis of natural language." In: Schank, Roger Carl and Riesbeck, Christopher Kevin (editors). *Inside computer understanding: Five programs plus miniatures.* Hillsdale, NJ: Lawrence Erlbaum Associates, 318–353.

[Brodie *et al* 1984] Brodie, Michael L; Mylopoulos, John; and Schmidt, Joachim W (editors) (1984). *On conceptual modelling: Perspectives from artificial intelligence, databases, and programming languages* (Topics in information systems). NY: Springer-Verlag.

[Brodie and Mylopoulos 1986] Brodie, Michael L and Mylopoulos, John (editors) (1986). *On knowledge base management systems: Integrating artificial intelligence, and database technologies* (Topics in information systems). NY: Springer-Verlag.

[Brooks 1987] Brooks, Frederick C (1987). "No silver bullet: Essence and accidents of software engineering." *Computer*, **20**(4), April 1987, 10–19.

[Gentner and Stevens 1983] Gentner, Dedre and Stevens, Albert (editors) (1983). *Mental models.* Hillsdale, NJ: Lawrence Erlbaum Associates.

[Hagman 1982] Hagman, Roy S (1982). *Language, life, and human nature* (Current inquiry into language and human communication 41). Edmonton: Linguistic Research.

[Hirst 1987] Hirst, Graeme (1987). *Semantic interpretation and the resolution of ambiguity* (Studies in natural language processing). Cambridge University Press.

[Johnson 1987] Johnson, Mark (1987). *The body in the mind: The bodily basis of meaning, imagination, and reading.* Chicago: The University of Chicago Press.

[Martin 1975] Martin, Graham Dunstan (1975). *Language, truth, and poetry: Notes towards a philosophy of literature.* Edinburgh University Press.

[McKeown 1985] McKeown, Kathleen Rose (1985). *Text generation: Using discourse strategies and focus constraints to generate natural language text* (Studies in natural language processing). Cambridge University Press.

[Ogden and Richards 1923] Ogden, C.K. and Richards, I.A. (1923). *The meaning of meaning.* NY: Harcourt, Brace, and World. 8th edition, 1946.

[Regoczei and Plantinga 1988] Regoczei, Stephen and Plantinga, Edwin Peter Owen (1988). "Creating the domain of discourse: Ontology and inventory." *International journal of man–machine studies*, ???(???), ???–???.

[Sloman 1978] Sloman, Aaron (1978). *The computer revolution in philosophy.* Atlantic Highlands, NJ: Humanities Press.

[Sowa 1984] Sowa, John F (1984). *Conceptual structures: Information processing in mind and machine.* Reading, MA: Addison-Wesley.

[Waterman 1986] Waterman, Donald A (1986). *A guide to expert systems* (The Teknowledge series in knowledge engineering). Reading, MA: Addison-Wesley.

A The OCRA car registration system

A.1 The context of the OCRA expert system

The acquisition of world knowledge extends to a much larger sphere of activity than traditional rule-based expert system construction would indicate. To illustrate this, let us take the Kingdom of Oz car-registration problem as described in the Appendix to this book [pg. XXXX]. The two????-page document, together with the exercises, can be construed as a requirements specification.

Let's pretend that Oz really exists, that their car industry is not too different from that of North America, and that we are analysts whose job is to examine OCRA, analyze the system, and design a new one. The system will incorporate expertise, the expertise of how to effect valid car registrations. It will also include a database and a natural-language interface component (Figure 10). Our task, as knowledge acquisition analysts, is to establish a particular conceptual model of a particular domain of discourse (one of many possible) by defining an ontology with conceptual graphs.

The work unfolds by stages. At each stage, we have a more and more complete picture of the world of OCRA, as defined by the partial model of the domain of discourse. In this paper, we can, of course, give only a small sampling of the work carried out at each stage, for a fully detailed description would contain hundreds of pages of text and conceptual graphs. Since this is an imaginary situation, we will have to hypothesize what answers OCRA officials might give to our questions.

A.1.1 Looking at the requirements specification document

The requirements specification document itself [pg. XXXX] can be pictured as the product of a knowledge acquisition project. The result of such a project is often presented as a well-crafted, well-edited text, as the Oz specifications are. We can do further conceptual analysis on this text, but first we should imagine how it may have come about. We can imagine OCRA officials engaging an analyst to write the original document; or, perhaps, a staff member, acting as a systems analyst, drafted it. We will illustrate the gradual unfolding through a sequence of progressively more complex diagrams.

The base state, shown in Figure 7, before the specification document existed, consists of the Kingdom of Oz, which contains OCRA, cars as the physical objects, and the persons or institutions that own cars. We then populate the universe of Figure 7 by adding the first analyst, who produced the requirements specification document, obtaining Figure 8.

A.1.2 The goals

The knowledge acquisition work is goal-oriented. The goals are presumably set through a document or verbal instructions from OCRA. We are seeking knowledge that is conceptual in nature, mostly about abstract entities that are social constructs. The knowledge is to be used in the expert system components shown in Figure 10, specifically for the following:

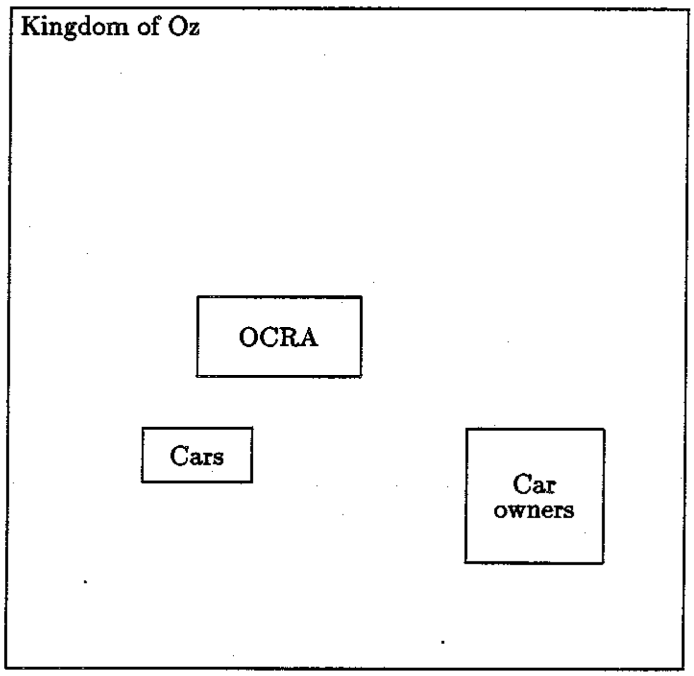

Figure 7: Base state.

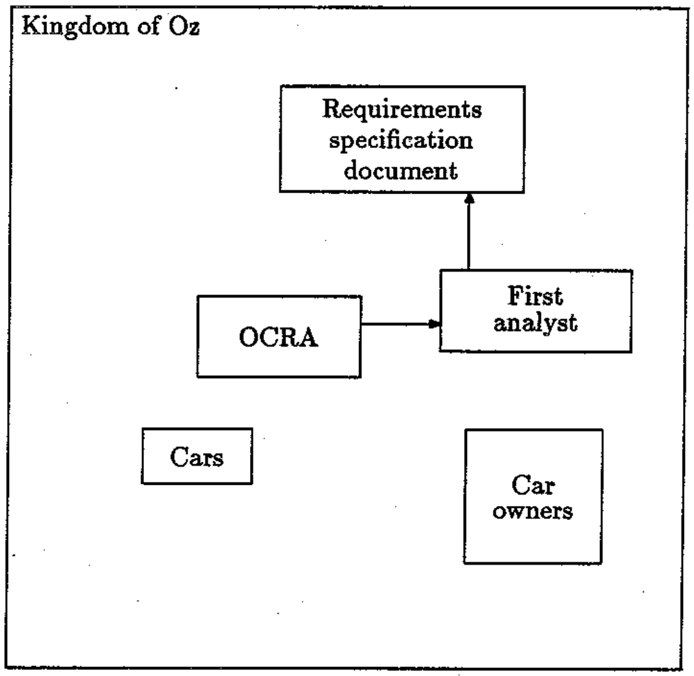

Figure 8: Second stage: We add the first analyst, who produces a specification document for OCRA.

- To build the rule base and constraints checker for an expert system that assists clerks in registering cars and checking the database for possible violations of the rules.
- To design the conceptual schemata for the database, including various user views.
- To design, for the natural language interface, the knowledge base that contains the domain-specific knowledge.
- To write public documentation to be used by:
  - Clerks, to operate the OCRA Expert System;
  - OCRA management, to know what reports to expect from the system;
  - Members of the public, including institutions such as car dealers and manufacturers, advising them on the regulations and procedures.

A.2 Creating the domain of discourse

Let us suppose that we as analysts are partners of a consulting firm, OSAKA Inc. (Oz Systems Analysis and Knowledge Acquisition), and we are assigned the task of working on the OCRA Expert System. We are handed the OCRA requirements specification document (Figure 9), together with the goals and objectives and the request to carry out the knowledge acquisition work and produce a design document for the expert system. We start conceptual analysis of the text.

Among the sentences we analyze, we come upon this one:

> There are a number of manufacturers, each with one unique name.

We can't help noting that this is rather artificial-sounding prose. People do not usually talk this way. In fact, the statement as it stands is either false or does not make sense without additional contextual information. We can imagine the interview of an OCRA informant by the original analyst that might have run as follows:

Analyst: We have these manufacturers here. How do we identify them?

Official: We give them a name.

Analyst: You mean you use the name they are commonly known by?

Official: No, a manufacturer such as General Motors would commonly be referred to as *GM*. We don't use *GM*; we use *General Motors*.

Analyst: So *General Motors* is the name of the manufacturer?

Official: No, the name of the manufacturer is *General Motors Corporation*, just as the name of Ford is *The Ford Motor Company of Oz*. But that's too long for us, so we use *General Motors* and *Ford*.

Analyst: Thank you. Can you supply me with the name for each manufacturer?

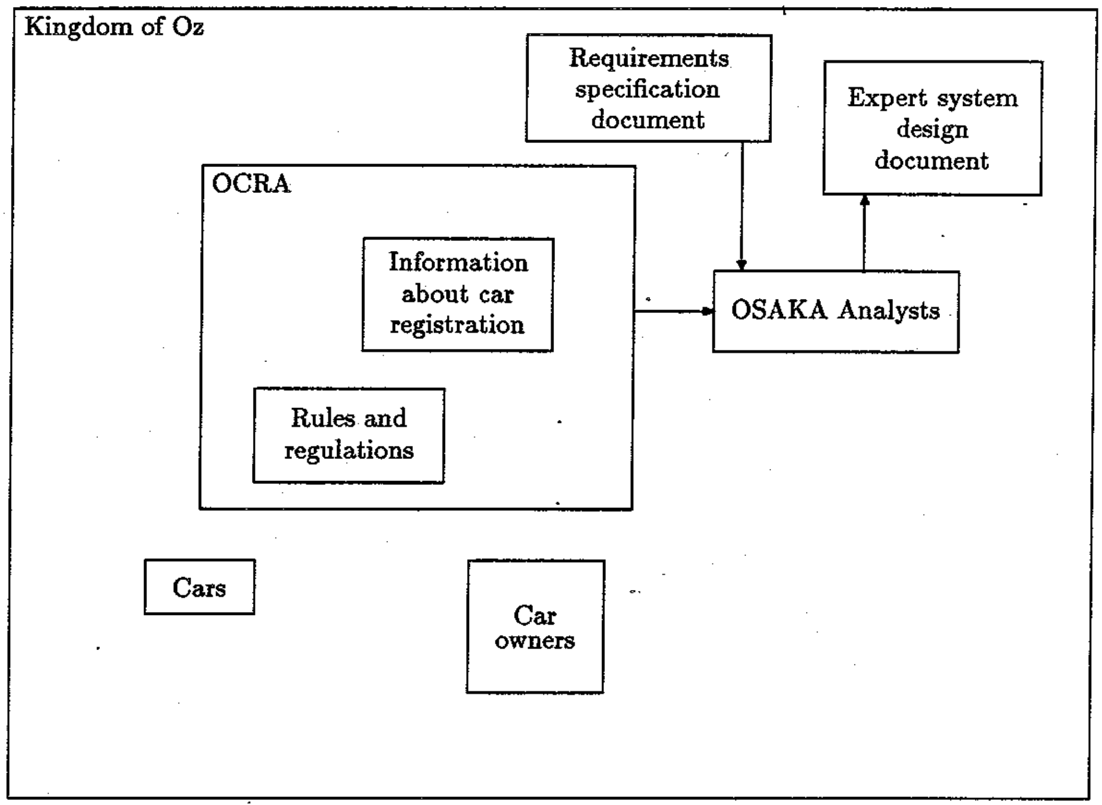

Figure 9: Third stage: Our company, OSAKA, is retained to design an expert system.

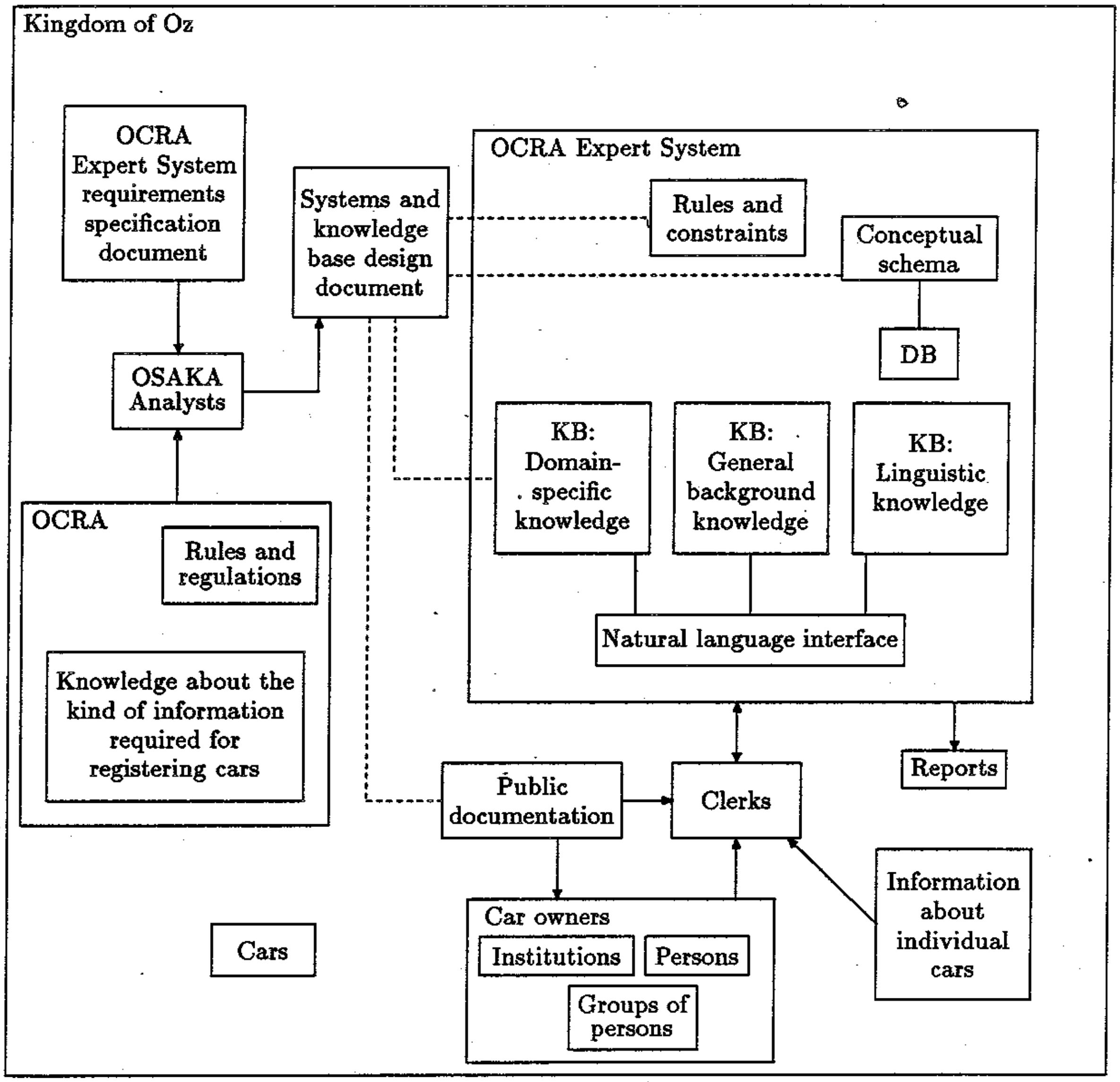

Figure 10: The OCRA Expert System and its context.

Official: Yes.

Following the interview, the analyst wrote in his notes:

> Each manufacturer has a unique name.

Later this was revised and appeared in the specification document, as we saw, as:

> There are a number of manufacturers, each with one unique name.

(We leave it as an exercise for the reader to do conceptual analysis on the terms *has* and *with*, in light of the dialogue above.) Subsequent analysts working on the project, such as OSAKA analysts, may take the latter statement as a fundamental fact, and perhaps record it as a set of canonical graphs, as in the answer to Exercise XXXXXX [shown on page XXXX of this book], not being aware of the actual conversation that took place at some stage of the knowledge elicitation process.

In a sense, the domain of discourse is defined by the requirements specification document; it was created by the first analyst, who decided what to record, and thereby created the document in which manufacturers were said to have unique names. But if the transcript of the above dialogue were included in the discourse, then the domain of discourse would contain preferred names, official names, common names, selected names, and a host of other kinds of abstract entities. Thus the domain of discourse is shaped by the admissible discourse fragments and decisions of what to include and what to exclude.

One can easily see how different domains of discourse can be constructed for what is ostensibly the 'same' object domain, each one 'correct', and each capturing different aspects and reflecting different concerns, viewing the object domain from different perspectives (cf Section 2.4).

A.3 Constructing the ontology

The domain of discourse is defined by the ontology. Recovering this ontology is our next task. It will contain the items shown in Table 4—that is, everything that makes up the world, how it's put together, and how it works.

In conceptual analysis of the text, we establish a number of types, kinds, or categories of entities. The creation of this taxonomy is very much at the discretion of the analyst and informant. They decide what 'exists' within the domain of discourse. Early versions of the taxonomy are usually quite informal, and largely determined by source documents and preliminary interviews. For example, the first version of our OCRA taxonomy might appear as in Table 5. We note that each of these taxonomic categories is itself defined by a concept. These concepts are shown in Table 6. And above all, we need the concept [CONCEPT] as well as the subtypes of it that define the extended meaning triangle:

> [WORD], [WORD-CONCEPT], [REFERENT-CONCEPT], and [REFERENT].

Entity-like types are all those that are not roles, attributes, events, processes, relationships, or actions. This corresponds to the intuitive use of the English word *entity*. Thus the entity types in Table 6 are:

Table 4: Items that will be included in the ontology.

Catalog of modes of existence
Concepts
Relations
Facts
Principles
Type labels
Canonical graphs
Schemata
Laws of the world

Table 5: The first version of the taxonomy for the OCRA Expert System.

'Event' concepts:
[ADD], [DELETE], [REGISTER], [DESTROY]
'Physical object' concepts:
[CAR]
'Physical, animate agent' concepts:
[PERSON], [ANALYST], [OCRA-OFFICIAL], [CLERK]
'Lexical object' concepts:
[REQUIREMENTS-SPECIFICATION-DOCUMENT], [SYSTEM-DESIGN-DOCUMENT]
'Non-lexical object' concepts:
[CAR], [PERSON], [KINGDOM-OF-OZ]
'Attribute' concepts:
[MODEL], [YEAR-OF-PRODUCTION], [DATE-OF-DESTRUCTION], [FUEL-CONSUMPTION]
'Abstract entity' concepts:
[MANUFACTURER], [DEALER], [CORPORATION], [INSTITUTION], [AGENCY], [GROUP-OF-PERSONS], [FUEL-CONSUMPTION]
'Role' concepts:
[CAR-OWNER], [MANUFACTURER-OF-CARS]
'Group entity' concepts:
[ALL-REGISTERED-CARS-PRODUCED-BY-A-MANUFACTURER-IN-A-PARTICULAR-YEAR], [GROUP-OF-PERSONS]
'Relationship' concepts:
[OWNERSHIP], [YEAR-OF-PRODUCTION], [DATE-OF-DESTRUCTION]

Table 6: Concepts defining taxonomic categories.

[EVENT]	[ATTRIBUTE]
[PHYSICAL-OBJECT]	[ABSTRACT-ENTITY]
[PHYSICAL-ANIMATE-AGENT]	[ROLE]
[LEXICAL-OBJECT]	[GROUP-ENTITY]
[NON-LEXICAL-OBJECT]	[RELATIONSHIP]

[PHYSICAL-OBJECT]	[PHYSICAL-ANIMATE-AGENT]
[LEXICAL-OBJECT]	[NON-LEXICAL-OBJECT]
[ABSTRACT-ENTITY]	[GROUP-ENTITY]

These are by no means all the entity types that one can imagine, or all the ones that will come up in conversation, but they are the only types included at this stage, by the mutual decision of the analyst and informant.

Strict version control should be exercised on the ontology and the canonical graphs that govern it. For example, at this stage of the partial model we cannot use the ontology to say things like *The car broke through the barrier.* Such sentences would be ill-formed, or non-interpretable, given the present state of the domain of discourse, because nothing in the ontology allows an inanimate object to be the agent of an action. The analyst and informant may agree to add the concept [PHYSICAL-NON-ANIMATE-AGENT] to the ontology, or they may agree that all references such as cars breaking through barriers are *category mistakes*, violating canonical graphs for [CAR] and [BREAK-THROUGH], and they have to be rephrased by including a case-frame slot for an [ANIMATE-AGENT] such as the driver; thus, one would say, for example:

> The car with an absent-driver broke through the barrier. The owner is responsible.

The issues of volition, intention, and responsibility are raised. The analyst should be aware that these issues in practical knowledge acquisition work are often covered up by impersonal phrasing, such as illustrated by the following dialogue:

X: The car broke through the barrier.

Y: No, it didn't. The driver let it roll down the hill.

X: He's quite young. I guess the owner will have to pay for the damage.

At this stage, we can use the meaning triangle to reflect on the state of affairs. In Figure 11 we show what is available at this stage from the point of view of an outside observer conducting a walk-through of the analysis project. The state of affairs is best summarized by centering the

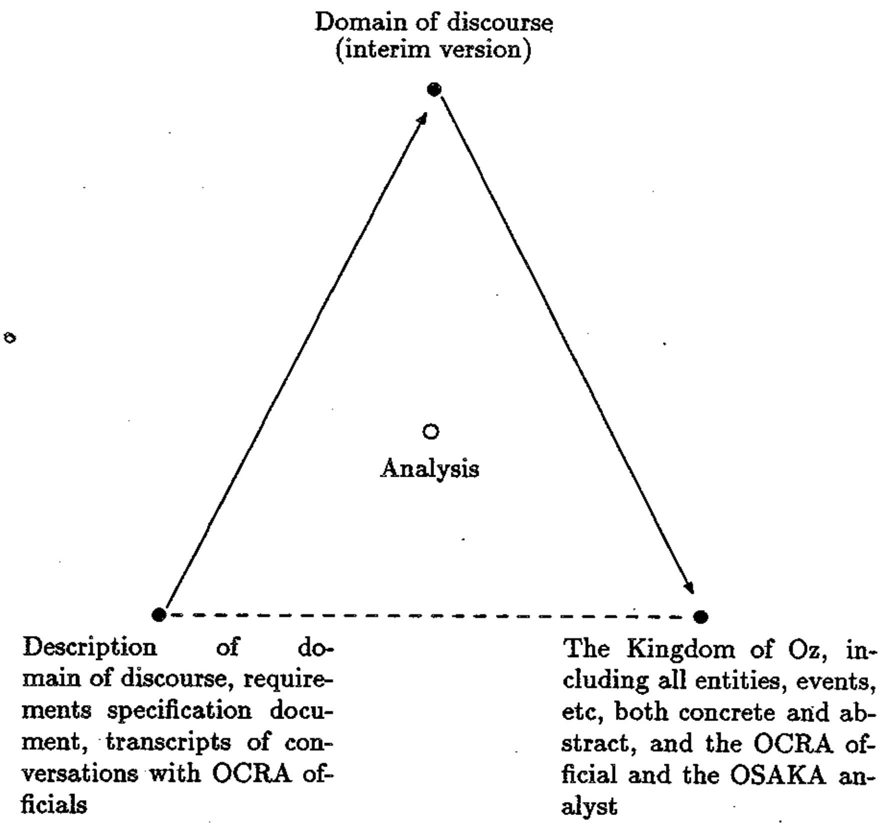

Figure 11: Analysis-centered meaning triangle.

meaning triangle on the analysis. The OCRA official sees the world somewhat differently. Her own perspective is summarized in the meaning triangle centered upon her (Figure 12). Likewise, the view of the state of affairs from the perspective of the OSAKA analyst is shown in Figure 13.

The construction of ontologies is best done with automated tools, similar to CASE tools (Computer-Assisted Software Engineering) and data dictionaries. Manual techniques are too labor-intensive for anything but the simplest applications.

A.4 Concepts and conceptual relations

In creating a formalized ontology, one has to decide which concepts to include as concepts, and which to are to be packaged into conceptual relations. For example, in the case in Section 3 of *The sky is blue*, we could write

[SKY] → (ATTR) → [COLOR: blue]

or

[SKY] → (COLOR) → [BLUE]

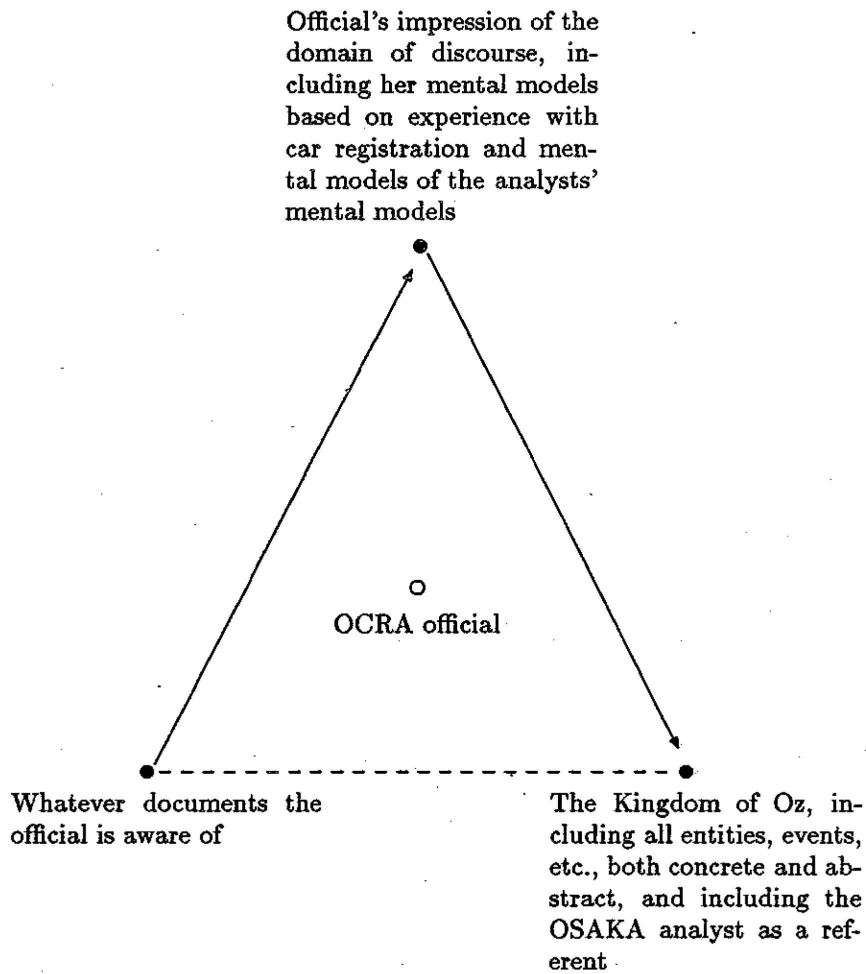

Figure 12: Meaning triangle centered on OCRA official.

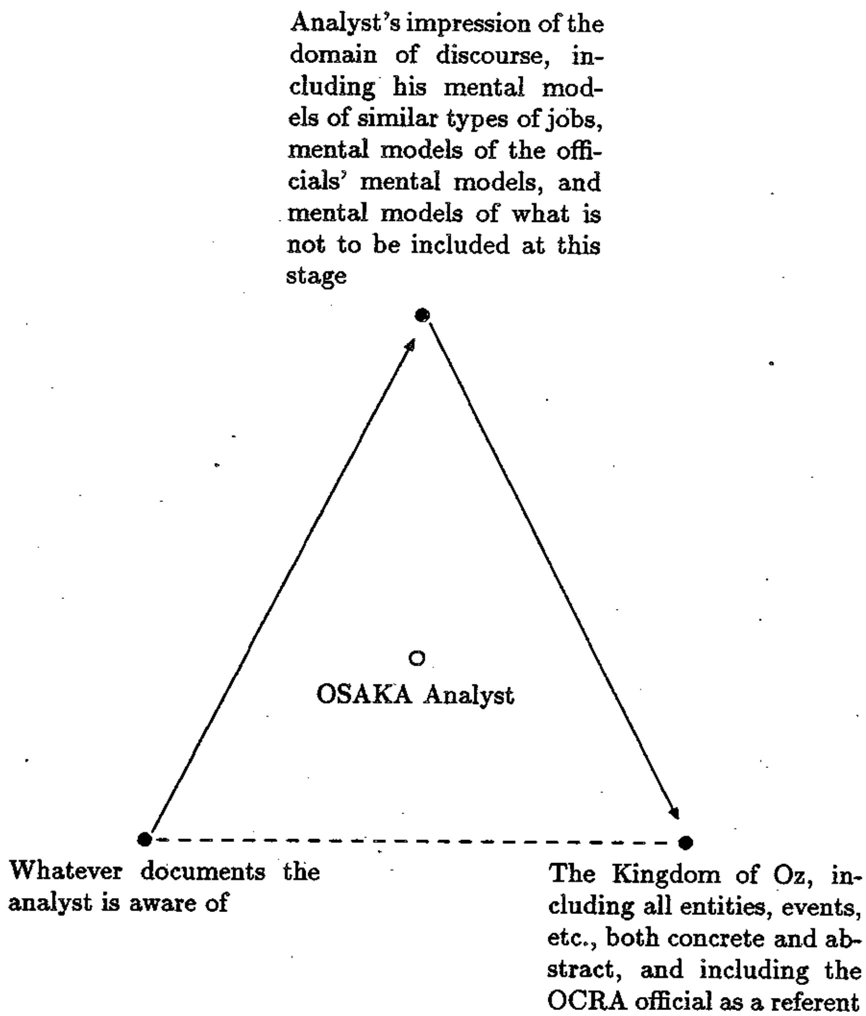

Figure 13: Meaning triangle centered on OSAKA analyst.

A general rule of thumb to follow: Never pack too many concepts into conceptual relations, but do pack into relations all the background structure that it is not necessary to explicate for this domain.

In the OCRA system, the concept [OWN] may best be turned into a conceptual relation (OWN), since we may not want to elaborate on the social and cultural knowledge that forms the background of what it means to own something. [OWNERSHIP] and [OWNER] are now definable in terms of the conceptual connective (OWN).

OWNERSHIP < RELATIONSHIP
[OWNERSHIP] -
[OWNER] → (OWN) → [PROPERTY]

For example:

The dealer owns cars.
[DEALER: #] → (OWN) → [CAR]

A related decision that has to be made is what is considered as a *semantic primitive* and what is defined in terms of other concepts. For example, if the concept [OWN] were included in the ontology, but neither explicated further nor defined in terms of other concepts, then it would be a semantic primitive relative to that ontology, *i.e.,* a concept defined by a one-node conceptual graph.

We can continue building up the ontology by taking blocks of text referring to such things as dealers, manufacturers, and cars, conceptually analyzing text, generating concepts of various types, and matching these concepts to text fragments. Examples of such entities lists are given in Tables 7 and 8.

Later in the knowledge explication process, the concepts are checked by OCRA officials. Definitions and schemata are constructed in conceptual graph form. The appropriateness of the names for the various concepts is evaluated. Concepts may be renamed, revised into more general or more particular concepts, or grouped together according to family resemblances [Sowa 1984, pg. 15].

Trying to pick the right conceptual relation makes people uneasy. Here are a few hints for practical work.

One can view conceptual relations as indicators of structure, devoid of content. When one thing X is somehow associated with another thing Y, this can be simply expressed as an ordered pair:

(X, Y)

In conceptual graphs, the relation (ORDERED-PAIR) would be the connecting node between [X] and [Y], as in

[X] → (ORDERED-PAIR) → [Y]

Otherwise, we may simply indicate the linkage between [X] and [Y] as

[X] → (LINK) → [Y]

Table 7: List of concepts associated with text fragment about dealers.

Events regarding dealers:

[START-TRADING]
[STOP-TRADING]
[ACQUIRE-CAR]
[DIVEST-OF-CAR]

Attributes of dealers:

[NAME]
[MANUFACTURER-LIST] (at most 3 members)
[TRADING]
[NON-TRADING]

Table 8: List of concepts associated with dealers as a group (for maintaining files on dealers).

Events:

[CHECK-DEALER]
(check dealer name against list of valid dealers)
[ADD-DEALER]
[DELETE-DEALER]

These would be typically neutral conceptual relations, containing structural knowledge, but 'drained' of all conceptual content relating to world knowledge. Likewise, we may decide to represent production rules (see section A.6) as

[ANTECEDENT] → (PRODUCTION-RULE) → [CONSEQUENT]

and leave the handling of the (PRODUCTION-RULE) connective as an implementation detail. This way we can elaborate on [ANTECEDENT] and [CONSEQUENT] without worrying about the exact nature of production rules as constructs.

There are three basic strategies that can be followed to decide what is or is not a concept or a conceptual connective.

- Follow a standard list, such as the conceptual lexicon in [Sowa 1984, Appendix B].
- 'Drain' the relations, as shown above, of all conceptual content as it relates to world knowledge, and have them express only structural knowledge; for example:

  [X] → (ORDERED-PAIR) → [Y]
  [X] → (LINK) → [Y]

- Conversely, pack lots of conceptual content into the connective nodes, especially the kind of content we do not want to worry about or emphasize at this stage of the analysis. For example, we can make the term *ownership* into a connective, as in

  [OWNER] → (OWNERSHIP) → [PROPERTY]

  The concept [OWNERSHIP] may have several large schemata associated with it, so rather than elaborate on the concept of ownership, we instead use the conceptual relation (OWNERSHIP) as a connective.

A.5 Conceptual knowledge about abstract entities: Knowledge explication

Life is not simple, and it is the job of the knowledge acquisition analyst to point this out to the client. For example, the analyst may decide to further explore concepts such as [OWN], [OWNER], and [OWNERSHIP], perhaps even using the full Sowa-Sloman algorithm (see section 4.3.1). The interview may start with the text from the OCRA specification document:

> At any time, a car may be owned by either its manufacturer, a trading dealer, a person, or a group of persons.

The analyst, building upon his own experience with database design, might want to explore questions relating to constraints on the modification of records:

> Do cars necessarily have to have owners?
> Can the content of an owner field be deleted from a car record?
> Can the owner field in a car record have a null value?

The OSAKA analyst may start a dialogue with the OCRA official to further explicate the knowledge:

Analyst: Do all cars have owners?

Official: Yes.

Analyst: Is it possible to register a car without an owner?

Official: No.

[Up to this point, things are rather easy.]

Analyst: Then my understanding is that an owner can divest himself of a car only by passing it to another owner.

Official: Yes, that's right.

[This seems to mean that an owner can be stuck with a car forever. How does one get rid of a car that no one will buy?]

Analyst: Does the new owner have to consent to the transaction, or can the old owner just foist the car on him—maybe not even telling him of the 'gift'?

Official: The new owner has to sign the instrument of transfer.

Analyst: Can one stop being an owner by destroying the car?

Official: Yes.

Analyst: An owner can't just abandon a car?

Official: Yes, he can abandon a car, but he is still the owner. He is still responsible for it.

This is where the technique of drawing simple conceptual graphs with generic conceptual connectives can best be employed. The informant is saying that the way she construes the domain of discourse is that the concepts [ABANDON], [OWNER], [CAR], and [RESPONSIBILITY] should be connected as in Figure 14. The existence of the path on the left-hand side of the Figure does not sever that on the right. We can summarize our findings in this graph:

[OWNER] → (LINK) → [RESPONSIBILITY-FOR-CAR]

We use the generic connective (LINK) in this graph to avoid specifying the details of the connection.

We note parenthetically that the graph

[OWNER] → (OBLIG) → [RESPONSIBILITY]

can also be diagrammed in many different ways, such as

[OWNER] → (ATTR) → [RESPONSIBILITY]

where

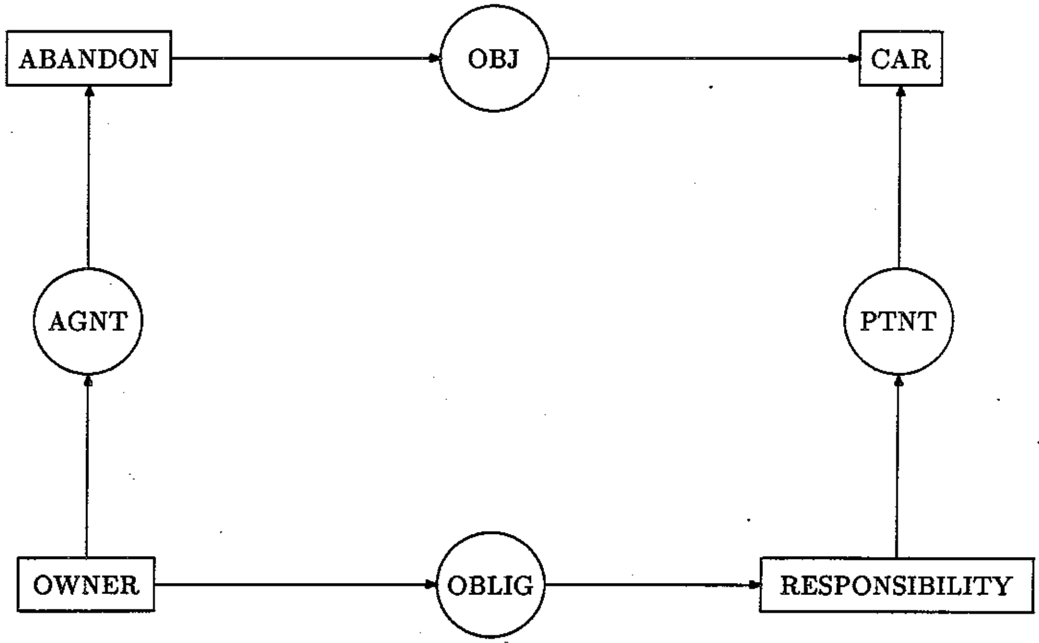

Figure 14: Conceptual relationships between ownership, responsibility, and abandonment.

RESPONSIBILITY < OBLIGATION.

We continue the knowledge explication process.

Analyst: What if the owner of the car dies?

Official: I don't know. I've never personally run into that situation. You'll have to ask the Registrar.

We have located a point in explicating the knowledge where either new concepts have to be introduced into the domain of discourse, or restrictions have to be made to specify what will remain excluded. Since we can't outlaw dying, we may explore the possibilities of its consequences for car ownership:

- The dead person still owns the car; but then what happens to ownership responsibilities?
- The dead person's estate owns the car—in which case we have to introduce a new category of ownership, enlarging what the requirements specifications stated.
- Ownership reverts to the Kingdom of Oz, and the Public Trustee impounds the vehicle, assuming all responsibility.
- Perhaps the Registrar informs the analyst that the case he is thinking of cannot arise under native Oz customs. In Oz, people are given a Viking-like burial, in which owners are cremated *in* their cars or the car is placed on the deceased's funeral pyre. Hence cars, upon the death of an owner, are immediately disabled and are treated as funeral accessories.

Similar problems are treated in detail at the knowledge explication stage. For example, when do cars come into existence, and when do they cease to exist? Suppose a car has come off the assembly line with a defect that prevents it from being released for sale, although it is sitting in the manufacturer's parking lot. Is this car already manufactured, or is it still in process?

New abstract entities may come into being, such as those defined by the concept [GROUP-OF-PERSONS]. We could reify this concept by projecting it into the object domain. But sometimes there is an effort to obey Occam's Razor and reduce abstract entities to more concrete ones. For example, in the specification document we read

> If a car is owned by a group of persons, each is regarded as an owner.

To operationalize concepts such as [GROUP-OF-PERSONS] as perhaps part of the second test of the Sowa-Sloman algorithm, it is always useful to try to go from the more to the less abstract—here, to [PERSON].

We should recall that the knowledge acquisition process we are describing deals with conceptual knowledge about abstract entities. These abstract entities are social constructs; that is why the analyst has to talk to people to find out about them. Explicating knowledge about abstract entities can be illustrated, for example, by emphasizing the importance of drawing a distinction between terms and concepts (see section 2.4.3). In the requirements document, the three terms *Oz*, *Land of Oz*, and *Kingdom of Oz* seem to be used interchangeably. Do they refer to the same concepts? We have to ask people. The *Kingdom of Oz* sounds very much like an abstract legal entity, perhaps the clearest example of a social construct. But for patriotic purposes, especially for anti-monarchist inhabitants of Oz, the object of nationalist loyalty and pride may be the *Land of Oz* and not the *Kingdom.* As for the term *Oz*, it may refer to the generic concept [OZ], of which the following may be subtypes:

- [OZ-THE-COUNTRY: #], that is, the referent Oz.
- [OZ-AS-A-NATION].
- [OZ-AS-A-GEOGRAPHIC-ENTITY].
- [KINGDOM-OF-OZ-AS-A-LEGAL-ENTITY].
- [OZ-AS-A-COLLECTIVE-CONCEPT-OF-WHICH-THE-OTHERS-ARE-SPECIALIZATIONS].

This last concept is very close to the root node of a type hierarchy. It is a useful concept to express vagueness, where vagueness may be practiced deliberately. It may turn out that [OZ] is just the same concept.

A.6 Production rules and constraints

To control the updating of the registration database, we need a knowledge-based system that captures the expertise of being able to decide what is a valid transaction. We may express this knowledge as *production rules.* Production rules are condition–action pairs that could, at times, be written as logical *if–then* statements. Typical examples are these:

- If X is the case, then take action Y.
- If X is the case, then take action to make Y the case.
- If X is the case, then Y is the case.
- If X is the case, then conclude that Y is the case, with probability Z.

Conceptual graphs provide a wide range of notational facilities to express the production rules exactly as the informant and analyst intended. For example, the above rules have the following structures.

- [CONDITION] → ⟨ACTION⟩
- [CONDITION] → ⟨ACTION⟩ → [RESULT]
- [CONDITION] → (IF-THEN) → [CONCLUSION], or
  [ANTECEDENT] → (IF-THEN) → [CONSEQUENT]
- [CONDITION] → ⟨ACTION⟩ → [CONCLUSION] → (PROB) → [NUMBER: #]

In a generic way, a production rule can be simply represented as:

[CONDITION] → (PRODUCTION-RULE) → [ACTION].

Some car registration knowledge is best captured as production rules. For example, if there is an attempt to register a car without an owner, permission to effect the registration should be denied. The appropriate expert system rule would say something like this:

> If the owner field is blank, deny the file update operation and print diagnostic message #328.

This is a condition–action pair, with two actions representable, in general, as

[CONDITION] → ⟨ACTION⟩

with

[CONDITION] –
  [OWNER-FIELD: @null]

(or other notation appropriate for a null value), and

⟨ACTION⟩ –
  ⟨DENY-FILE-UPDATE⟩
  ⟨PRINT-DIAGNOSTIC-MESSAGE: #328⟩.

A.7 Implementing the knowledge in expert systems using traditional architectures

As we noted in section 6, a frame-based expert system shell that accepts and executes conceptual graphs would give a direct implementation of a knowledge base and an expert system that were specified in the conceptual graphs notation. However, there is no need to wait for such a shell. We can use traditional architectures right now to implement knowledge-based systems. As shown in Figure 10, rules and regulations for car registration can be encapsulated in an expert system of rules and constraints that control the database update.

The ontology of the OCRA expert system can be used to design a conceptual schema for a semantically rich database. Database management system (DBMS) software, with a data dictionary, query language, and report writer would satisfy most requirements. The same type of ontology can be used to create the knowledge base for the natural language interface. Domain-specific knowledge relating to OCRA car registration and related topics would reside in this knowledge base as conceptual graphs from which text may be generated (following [McKeown 1985]).

Last, but not least, documentation in the form of standards and procedures manuals, instructions for the public, user documentation for the operating clerks, and on-line help menus all contain world knowledge that was captured in the knowledge acquisition process. As the OCRA example illustrates, conceptual graphs are a versatile notation that can be used as a blueprint for a knowledge-based system. The implementation can use present-day, traditional architectures, such as a rule-based expert system shells, DBMSs, and database-schema-driven text generation systems.