

Proceedings of the  
4TH KNOWLEDGE ACQUISITION FOR  
KNOWLEDGE-BASED SYSTEMS  
WORKSHOP

Banff Conference Centre  
Banff, Alberta, Canada  
October 1-6, 1989

Sponsored by the  
American Association for  
Artificial Intelligence

Workshop Co-chairs:  
John H. Boose, Boeing Computer Services  
Brian R. Gaines, University of Calgary

## Sortal analysis with SORTAL, a software assistant for knowledge acquisition

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

### Abstract

SORTAL is a software assistant for performing meaning-triangle-based sortal analysis. This paper describes its architecture and possible implementations. Conceptual analysis and conceptual modelling are central components of the informant-and-analyst-based, natural language-mediated knowledge acquisition process, but focusing on concepts is not enough. The "aboutness" of the language used in the interview forces the analyst to recognize distinctions between words, concepts, referents, and cogniting agents. Creating frame-like representations for agent-centered meaning triangles, as well as updating ontologies, keeping track of multiple domains of discourse, and the creation of knowledge bases for use in other systems are tasks that can be assisted by a software tool such as SORTAL. We sketch the requirements for such an assistant, give examples of its operation, and address implementation issues.

### 1 INTRODUCTION

Knowledge acquisition is difficult if for no other reason than that the analyst has to keep track of a vast bulk of textual and conceptual material. Software tools are essential. In this paper we describe a software system, SORTAL, a knowledge acquisition assistant for sortal analysis, that helps the analyst in her work. The development of knowledge-based software is merely a special case of a more general issue of software engineering: how can we use CASE (computer-assisted software engineering) tools to make our work easier in systems analysis, requirements specifications, and data modelling?

SORTAL is one of three theoretical or implementable software systems that we are developing. Together, they address the problems of text handling, conceptual analysis, natural language

understanding, and the extraction of knowledge from text. The systems are these:

- LUKES (Language Understanding and Knowledge Extraction System) is a theoretical, "gedanken" system for generating expert systems. LUKES is based on a three-level architecture of 'cogniting' agents, having verbal, conceptual, and subconceptual levels. It incorporates a theory, based on concept cluster attachment, of natural language understanding and knowledge acquisition from text. We describe this system and its theoretical foundations more fully in [Regoczei and Hirst 1989b].
- LOGOS ("the word, the expressed concept") is an implementable version of LUKES. In its present form, LOGOS has a two-level architecture: verbal and conceptual; *i.e.*, it omits the subconceptual level of LUKES. It is knowledge-based, and is bootstrapped by SORTAL (which we discuss next) (see figure 1). Its input is natural language text and descriptions of conceptual structures. As output, it produces knowledge bases consisting of text with concept cluster attachments.
- SORTAL, the main topic of this paper, is a KA analyst's tool that assists in the use of the KA methodology that we have described in earlier papers [Regoczei and Plantinga 1987, Regoczei and Hirst 1988, 1989a, 1989b]. In particular, SORTAL uses the meaning triangle distinctions of verbal, conceptual, external-referent, and agent domains of existence discussed in [Regoczei and Hirst 1988]. It helps the analyst to build ontologies and inventories for domains of discourse from informant-analyst

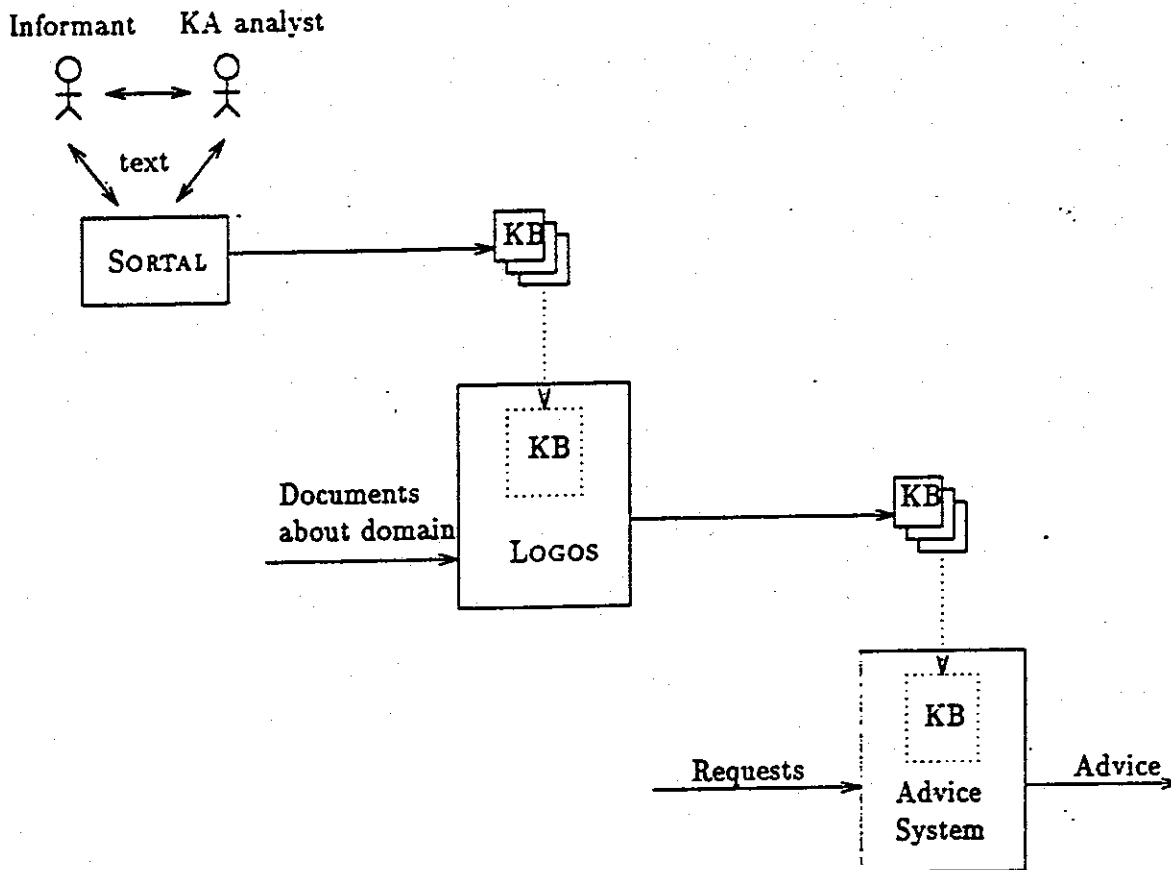


Figure 1: Bootstrapping LOGOS with SORTAL to build a knowledge-based system. (From Regoczei and Hirst [1989b].)

interviews, other textual material, and direct observation of the object domain. It incorporates version control for domain of discourse creation, and other features described below.

The problem we are addressing is to build conceptual models on the basis of natural language input to the analyst, making use of the analyst's existing knowledge base, and, having done this, to take the conceptual structures so obtained and recast them in the framework of one or more of the following:

- Knowledge base construction for expert systems;
- Requirements specifications for software development;
- Data modelling for conceptual schema design;
- Systems analysis.

We see both SORTAL and LOGOS as kinds of CASE tools that integrate some of the best current techniques with the new theoretical foundations that we are offering. We observe that there is a good deal of software development being done right now: some of it is done well and some of it is done badly. Our view is that it should be possible to form a descriptive theory of what is being done in the field, examine it and refine it, and formulate a systematic methodology that one could recommend. A methodology for KA, of which we picture *sortal analysis* (see *next section*) as one component, is best embodied in software tools. SORTAL is an example of the approach that we are taking.

## 2 SORTAL ANALYSIS

Our approach to knowledge acquisition was described in the earlier papers cited above. In these papers, we argued for the importance of creating a *domain of discourse*, and record-

ing the *ontology* and *inventory*, in the context of an informant-and-analyst-based, natural language-mediated interaction. We advocated the *meaning triangle* (figure 2), as a way of drawing distinctions between entities with different ontological status. We advanced the mechanism of *concept cluster attachment* to operationalize both knowledge extraction from text and natural language understanding. In connection with the meaning triangle and concept cluster attachment, we discussed *sortal analysis*.

Sortal analysis is a form of classification. From input of text, general signs, or other observational material, it separates and clearly labels words, concepts, referents, and agents. Sortal analysis can be regarded as a generalization of conceptual analysis. In the case of conceptual analysis one is inquiring about the concepts 'behind' the text. But we have to go beyond this. We need to distinguish between a language in the abstract sense, such as English, and 'language-in-use', such as the utterance of an English sentence to communicate with another person or to perform a task. Language-in-use is *about* something. In the case of sortal analysis, we are inquiring about what *sort* of things are being talked about, as well as recognizing text itself as a sortal category, namely verbal. Sortal analysis recognizes the central importance of the 'aboutness' of language-in-use.

Sortal analysis consists of generating and constructing concepts, concept clusters, and conceptual structures, forming mental models of the conceptual kind, categorizing, and contextualizing, while recognizing the fundamental distinctions between words, concepts, and entities that are components of an object domain in the external world.

For example, the sense of a word can be defined as an ordered pair, whose first component is lexical, *i.e.*, the word, and whose second component is a concept cluster centered upon some concept. For example:

( "piano",  
 {{{[PIANO] | [MUSIC], [SONATA], ...}}})

The central concept, before the vertical bar, is the *denotation* and the peripheral concepts of the cluster are the *connotations*.

A second example: *Sortally composite enti-*

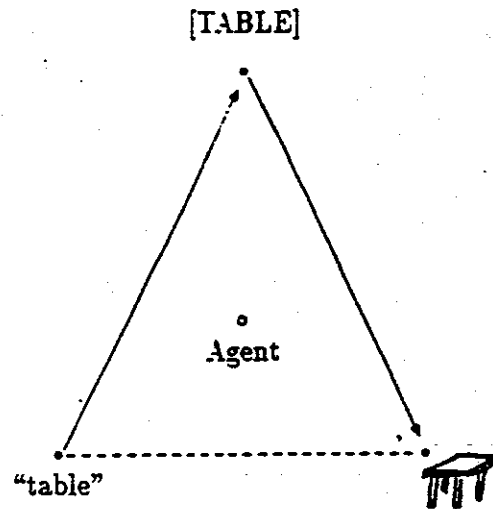



Figure 3: Meaning triangle for a sortally composite entity.

*ties* are frames. In perceiving a table, we can decompose the activity into perceiving, perceiving that something is there, and attaching the concept [TABLE] to the perception. If we verbalize the perception, then we have a triple:

(  . , [TABLE], "table" )

We can picture this triple as a frame. Such frames can be diagrammed on a meaning triangle with the contents of the individual slots diagrammed on the referent, concept, and word nodes, respectively, as in figure 3.

Sortal analysis is necessarily *perspectivist* [Bullock *et al* 1955, p. 641-2], *constructivist*, and *agent-oriented*. One cannot do sortal analysis on a sentence alone without further information and knowledge about people and the world. We will illustrate this point with an example below.

In sortal analysis, our first step is to take the agents seriously and feature them prominently, because the distinction between concept-like and referent-like entities is to be drawn relative to an agent. This categorization of entities is essential. We can note the two-level architecture in the case of the NLU approach: The English text is at the verbal level, and the con-

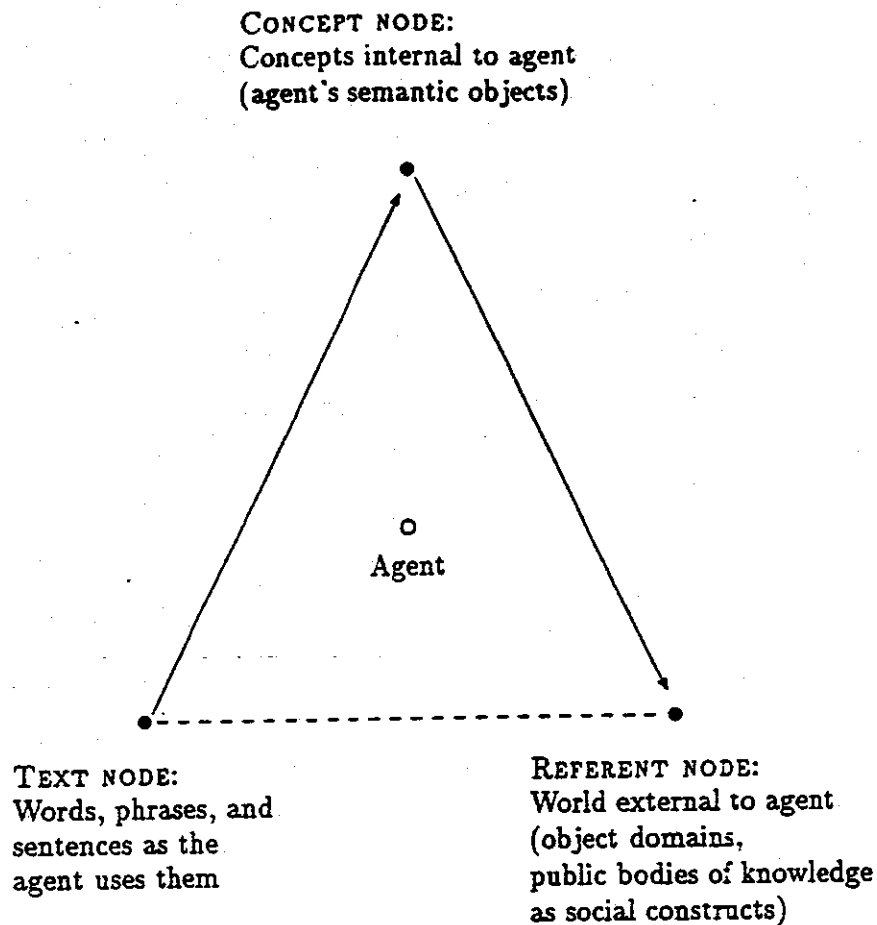


Figure 2: An agent-centered meaning triangle.

ceptual dependency representation is at the conceptual level. But this does not exhaust the ontological categories. In knowledge acquisition, we have to take cognizance of agents and the world they live in. In particular, we have to use surrogates (see section 7) to construct models of the object domains.

### 3 SOFTWARE SUPPORT FOR SORTAL ANALYSIS

Even for relatively simple KA problems, the volume of material to be handled far exceeds the capabilities of manual systems. Certainly, automated techniques are more cost-effective. SORTAL is a software assistant for the task of sortal analysis. It supports the recording of concept cluster attachments and the diagramming of the ontology and inventory on frame-like versions of the meaning triangle. Figure 4 lists SORTAL's main functions.

SORTAL's being an assistant means that the person being assisted is also a part of the sys-

tem. This has to be noted, because concept formation is something presently done by people, in this case the analyst, and not by computers (current research in AI notwithstanding). It is the analyst who decides what is an agent, how many triangles are needed, and how surrogates will be created for referent entities in the object domain. SORTAL records, organizes, and prompts, enabling the analyst to keep track of large quantities of textual and conceptual material.

### 4 HOW SORTAL HELPS

To put it very simply. SORTAL sets up frames, and prompts the user to fill the slots.

What makes SORTAL significant for our purposes is the distinction drawn in categorizing entities into verbal, conceptual, agent-like, and referent-like. Furthermore, it distinguishes public concepts from other entities on the referent node, and supports the construction of sortally composite entities.

- Helps analyst draw meaning triangles.
- Supports the construction of agent-centered meaning triangles in frame-like notation.
- Supports domain of discourse creation.
- Records ontology.
- Records coreference links.
- Provides version control.
- Effects categorization into verbal, conceptual, referent-like, and agent-like entities.
- Supports the construction of sortally composite entities, e.g., word-concept cluster pairs, or named concepts, or conceptualized referents.
- Keeps track of hybrid entities on the referent node, for possible future refinements of the analysis.

Figure 4: Main functions of SORTAL.

When confronted by a piece of text, the analyst asks questions such as: who is uttering this text, what are this person's mental models, what is the text referring to, and is there in fact a referent? That is, she makes a distinction between understanding text, understanding people, and understanding the world [Regoczei and Hirst 1989b]. She makes the decision as to how many and what kind of agent-centered meaning triangles to start with, and whether or not to include a triangle centered upon herself as an agent. Next, she decides which text fragments to deal with, how compositional the concept cluster attachment should be, and whether she should follow a literal interpretation or a more figurative one. In setting up multiple domains of discourse, she separates out the various conceptual domain components on the concept node of the meaning triangle. She decides whether or not to include *platonic domains* either on the concept node or on the referent node.

SORTAL helps to populate the referent domain with surrogates. Because the object domain on the referent node is structured by projecting concepts onto the node, a judicious policy has to be adopted to trade off:

1. The inclusion of sufficient surrogates not only to cover the referents of the concepts but also to provide a rich-enough background for breakdowns and unexpected deviations [Winograd and Flores 1986];
2. Not including so many surrogates that the inventory becomes unmanageable.

In deciding what to include on the concept node and what to place on the referent node, the guiding consideration is the trade-off between models of mental models and models of 'the truth'. This is the great thematic distinction between 'objective' and 'subjective', and it takes a certain amount of humility on the part of the analyst to realize that what appears to her as 'God's Own Truth' is little more than a verbalizable version of a conceptual model, which, in turn, models her own mental models, some of which may be at the subconceptual level.

In the case of an agent-centered meaning triangle, there are at least two agents: the agent in the center of the triangle, and the analyst constructing the triangle. We have to distinguish between the agent structuring the object domain and the analyst structuring the object domain. This is a non-trivial distinction. The analyst and the agent may see the world very differently, i.e., from different perspectives. SORTAL can keep track of such distinctions. Thus not only does it keep track of multiple domains of discourse for a single agent, but it has to keep track of different views of the 'same' domain, whether the object domain or the domain of discourse.

## 5 AN EXAMPLE

To illustrate sortal analysis and SORTAL, we will now present a situation that we will use as a running example, interwoven with our theoretical discussion. We will consider a sentence similar to one originally analyzed by Schank [1973, p. 228] to demonstrate his conceptual dependency representation. We want to highlight how different a KA analysis of text is from that taken in conventional research in natural language understanding (NLU). The text is:

(S1) John sliced the salami with a knife.

Schank produces a semantic representation of this in his CD notation, employing his set of

primitive actions; see figure 5.<sup>1</sup> Being able to produce such diagrams and then use them to answer questions is said to constitute language understanding.

Our approach necessarily starts not with sentences but with utterances—that is, the particular use of a sentence by an agent in a particular situation in the world. Our analysis takes into account a consideration of the background to the utterance: certain cultural or anthropological observations and certain pragmatic considerations.

Firstly, in North America, salami usually comes shrink-wrapped or sliced at the delicatessen. Very few people slice salami. They hardly even slice bread any more. They do know, however, that salami comes sliced. The slicing is done not with a knife but with a machine with a large rotary blade that can be set for different thicknesses of slices. Moreover, many people would, if pushed, realize that they are confused by the notion of salami. They would have difficulty telling salami from Polish sausage or from jagdwurst. For example, is bologna a type of salami? What about Genoa salami? The relevance is that bologna is relatively easy to slice with a knife, but soppressata is not. Schank didn't have to worry about these issues because there he had no real salami slicing. His main concern was computational linguistics of a 'pure' kind—agentless, timeless, contextless, ethereal—whereas for us dialogue is there to acquire knowledge possessed by an agent and used in an actual situation for a particular purpose. (Yes, we are using a pretend world too, in order to make our example small enough to fit in this paper, but we are nonetheless aiming at some measure of realistic and dramatic verisimilitude.)

The second observation concerns discourse rules. The sentence seems 'unnatural' as a fragment of actual dialogue. It is contrived; one could almost say that it was written in a kind of artificial natural language. In the case of a

<sup>1</sup>Schank's example was actually *I sliced the meat with a knife*. But his analysis of *John sliced the salami with a knife* would be exactly the same as his actual example but for the tokens *John* and *salami* replacing *I* and *meat*. In contrast, as may be inferred from the discussion below, the two sentences would give rise to very different analyses in our approach.

more casual dialogue, one would set up a context such as one in which John and some other people were preparing some cold cuts, in order to make a point such as this:

- John did the slicing.
- He was slicing salami, not bread.
- He used a knife, because he was skilled enough to be able to cut thin, even slices for the sandwiches.

This is a lot of new information to squeeze into a single sentence. Hence it stretches the bounds of the rules of co-operative discourse [Grice 1975]. The point is not that a sentence such as *John sliced the salami with a knife* would never be uttered; obviously it can be uttered, and is a perfectly acceptable, reasonable, well-formed English sentence. But without setting the context, establishing the agents, talking about the various referents, situations, and domains of discourse, one can do neither knowledge acquisition nor sortal analysis. So, according to the rules of discourse, the speaker must supply enough information, but not too much, and must make sure that the utterance is interpretable. In other words, the sentence has to be contextualized.

As we said, this example is originally from research in NLU. To make it a KA problem, we must put it in context. Let us assume that we are automating the sandwich production facility of a large catering company. We talk to the supervisor, asking about the source of some cold cuts on a plate, and he casually says:

(S1) "John sliced the salami with a knife."

Now we have to 'extract' the knowledge from this sentence. (For the operationalizability of this metaphor, see [Regoczei and Hirst 1989b].)

## 6 WORKING THE EXAMPLE BY HAND

### 6.1 A first version

What would this example look like if, instead of using Schank's approach, we tried conceptual analysis techniques? We would get a much richer analysis, and results that are much more suitable for knowledge base construction, although conceptual analysis still falls short of

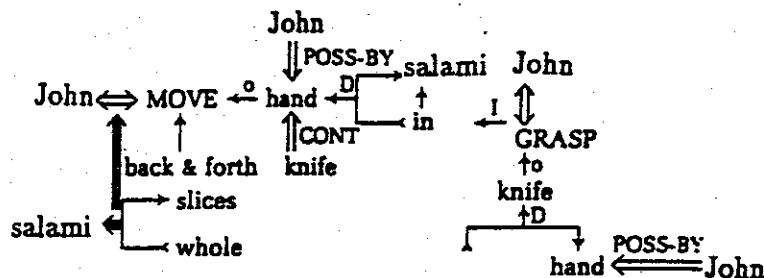


Figure 5: Schank's representation of *John sliced the salami with a knife.*

sortal analysis, as it addresses the ontology, but not the inventory, of the particular case or situation.

Let us try to work this example 'by hand', using conceptual analysis to explicate the knowledge 'behind' the text [Regoczei and Hirst 1989a] and use Sowa's conceptual graphs to record the results of the analysis [Sowa 1984]. Later on, we will introduce sortal analysis techniques. A simpler analysis takes sentence S1 as the text, generates the concepts closely corresponding to the lexemes, and arrives at a graph such as that in figure 6. This graph—let us call it Version 1—says, basically, that the activity of slicing has an agent, John, an object, the salami, and an instrument with which it is being carried out, the knife.

The ontology for our microdomain at this stage consists of four concepts, three conceptual connectives, and one statement of fact regarding John's activities. At this stage, we know nothing further about [KNIFE] and [SLICE], so for the moment they are to be considered semantic primitives relative to the microdomain we are building. Knowledge about knives or the activity of slicing may come from the informant or may be supplied from the private knowledge base of the analyst. For proper version control, we should be able to keep track of where this additional input is coming from. We will return to this very important point below.

## 6.2 Recording the resulting knowledge

Now let us consider the difficulties that the analyst faces in the task of making a permanent record of the interim findings. Until recently it had to be done by paper and pencil, or equiv-

alent. Alternatively, one could use a word processing package and the frame-like notation for meaning triangles suggested by Regoczei and Hirst [1988]. With a tool such as HyperCard, the analyst could actually draw triangles on a set of linked hypertext cards (figure 7).

But this is not much of an advance, as it still has some serious shortcomings, such as these:

- There is no connection between the entities in the text, concept, and referent fields. In particular, there are no coreference links between these entities.
- There is no support for forming sortally composite entities such as the ones that we need to clarify the example of Eddington's two tables [Regoczei and Hirst 1989b]:

```
(<#table, [TABLE-COMMONSENSE])
(<#table,
 [TABLE-NUCLEAR-PHYSICS])
```

The projection of concepts onto the referent node is necessary to structure the object domain here, and to label the surrogates. Alternatively, we could attach descriptive labels on the surrogates in a metalanguage which, of course, would look exactly like English. This causes a confusion—responsible for much philosophical head-scratching in the past—between the population of the text node and the explanatory labels attached to the surrogates.

- There is no facility for treating the populations of the text, concept, and referent nodes as different abstract datatypes with different permissible operations. In particular, the items on the nodes are not



[JOHN]-(AGNT)-[SLICE]-(OBJ)-[SALAMI]

↓  
(INSTRUMENT)-[KNIFE]

Figure 6: Conceptual graph for sentence S1.

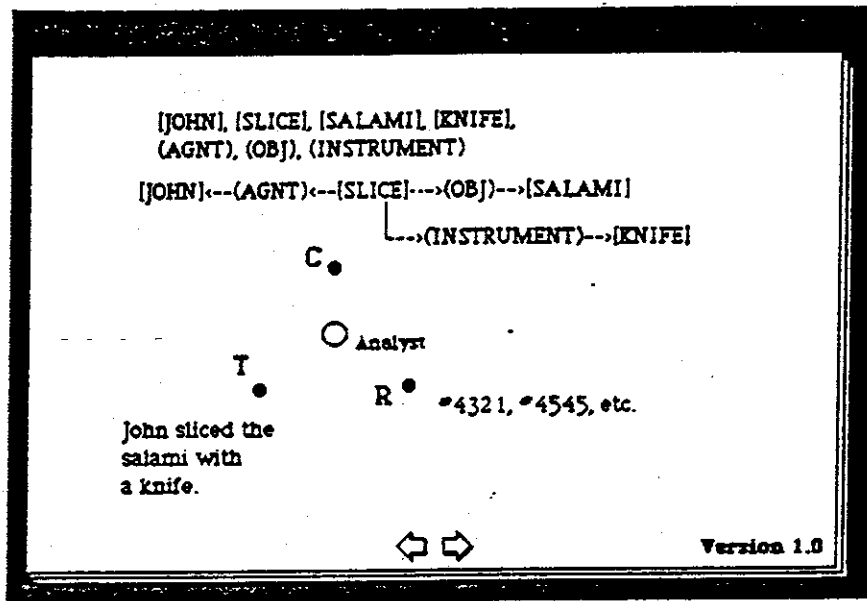


Figure 7: Version 1 of the analysis, hand-diagrammed in HyperCard.

available for further processing, such as, for example, generating knowledge bases from the information already available.

- There is no support for intuitively-needed human operations such as concept cluster attachment [Regoczei and Hirst 1989b]. Attaching concept clusters to words and referents is very much like free association or being reminded of one thing by another. As such, it is a subjective, agent-oriented activity not unlike forming coreference links or sortally composite entities, or the labelling of referent surrogates (see section 7 below).

This brief list makes it clear that a software tool supporting sortal analysis would have to provide substantial advantages beyond manual techniques, even when the manual techniques are supported by word-processing, database, and text handling capabilities.

### 6.3 Continuing to work the example by hand

Resuming where we left off above, we can look at the conceptual graph that expresses the factual statement of John's activities (figure 6) and perhaps notice some peculiarities having to do with the naming of concepts. We notice, for example, that the concept [SLICE] refers to the activity of slicing, rather than the piece of salami that is the final product of the activity. Ideally, the names of concepts should be intuitively obvious both to analyst and informant, and anyone else later reading the documentation of the interviews. We could decide to use the name [SLICING] instead, though Sowa does not use the gerundial form in his graphs. In any case, using the gerund consistently for an activity is unnecessary, because difficulty arises only when a word, such as "slice", is both a verb and a noun designating the product of the action. It might be easier to explain the concept with comments, rather than to agonize over the exact choice of the name. In casual terms, we

would like to elaborate and explicate—in effect, talk about the concept—including explaining that possible labels such as [SLICE-ACTIVITY] and [SLICING] are alternative names for the same concept that ‘defines’ a type of activity out there in the ‘real world’.

When there is no software support, the considerations we just described, although occurring in the thoughts of analyst and informant, do not get recorded. A text window, associatable with each concept, is a necessity for recording these explanatory comments. The text so generated may, at a later stage in the interview, be incorporated in one of the later versions.

Comparing our graph above with Schank’s (figure 5), we notice that Schank’s is more complicated. Its complexity is introduced by:

- Elaborating on the activity of slicing, in particular, slicing with the knife.
- The salami being turned from an integral, whole entity into a collection of smaller pieces. Schank, however, stops short of describing each piece as having a particular shape (i.e., being delimited by parallel planes) and a specified orientation relative to the salami as a whole.

We can incorporate similar considerations into our analysis by drawing canonical graphs (à la Sowa) for the concepts [SLICE-ACTIVITY] and [SLICE-OBJECT]. We can also consider adding further information through *type hierarchies* such as:

```
SLICE-ACTIVITY
  < CUTTING-ACTION
    < ACTION
```

or considering adding concepts with referent fields (not the same as surrogates!) such as:

```
[PERSON: John]
```

and:

```
[CUTTING-ACTION: John’s-slicing]
```

We may broach the issue of whether the distinction between action and activity needs to be specified at this stage, the intuition being that slicing is more like a prolonged activity than a single act or action. However, there is no need to take a rigid position on what may be included in the microdomain. Without a predetermined directory of concepts and a taxonomy, the analyst has to make decisions as the

microdomain is being constructed on what to include and in what form to include it. But it would be useful to have a software tool to keep track of these decisions for future use, possibly to scroll back to an earlier version to construct a different account of the knowledge being accumulated. With manual techniques, as noted above, the thinking is done, but is too cumbersome to record. Even if recorded, it is too difficult to reuse.

Focusing our attention now on the use of semantic primitives, we note that the primitives that we use are not absolute, but rather are relative to the microdomain being constructed. Since we are striving for a minimal covering ontology of sentence S1, at times we make the decision to treat a concept as a one-node primitive, and not expand it in terms of further canonical graphs. Schank takes a more absolutist position on what the ‘true’ primitives are. But Schank’s primitives may have more to do with his own psychological “thematic preferences” [Halton 1973] than with any platonic, true, or real structure of knowledge. For example, the action of slicing is expanded in terms of primitive actions and hence is not a primitive. But, we may ask, why is *knife* a primitive in Schank’s version? Why are actions to be explicated, but objects not? Is this because Schank is an active person? There is nothing simple about knives, nor about objects in general. We can think of some typical cases of knives as being things that are sharp-edged, and have a handle and possibly a point. We have strong subconceptual notions about knives and different kinds of knives. Many of these notions are conceptualizable. But some of the concepts used in building private conceptual models of knives do not have names. In talking about them, people usually resort to expressions such as “you-know” and “thingamajig”.

To keep the analysis within bounds, it is often wise to make an explicit decision on how much detail to include. This is to be construed as a design decision, and not in Schankian terms of having reduced the analysis to some collection of atomic elements that cannot be decomposed any further. For many purposes, there is no point in elaborating any further than to note, perhaps in the form of text added to Version 2,

that:

"The knife has a handle to grasp and a sharp edge to cut with."

Thus, at the Version 2 stage we can introduce concepts such as [HANDLE], [GRASP], [EDGE], [SHARP], [CUT], etc. We introduce them first by increasing the text, and then seeking to construct the covering ontology. Other techniques would be to do concept cluster attachment [Regoczei and Hirst 1989b] and then elaborate on the concepts introduced. As we are developing larger and larger minimal covering ontologies for the text available, we exercise version control. We have version control; Schank does not. However, enforcing version control with manual techniques requires the kind of self-discipline that usually cannot be attained in a production environment.

To conclude: At this stage we have two versions of the analysis. Version 1 was shown earlier in figure 7. Version 2 appears in figure 8. The name that appears as the agent's name in the triangle is that of Stephen Regoczei, one of the authors of this paper, because it was he who did the analysis. Had it been done by someone else, or had it been done by Regoczei at a different time, the results might have turned out somewhat differently. (As in data modelling for databases, there are no unique answers, only pragmatic design decisions.)

## 7 SURROGATES FOR REFERENTS

Probably the greatest difference between what we are doing and what the NLU approach does relates to surrogates. "John" is the name of a person, and that person is an entity with the concept [PERSON] attached to it. However, the entity to which the lexical object "John" and the conceptual object [PERSON] are attached—if such an entity exists at all—is made of flesh and blood and cannot be placed into silicon chips. So let us agree to appoint a unique character string, such as #35467829, which will 'stand in' for this flesh-and-blood entity, acting as a surrogate. Thus our meaning triangle will appear as follows (in frame notation):

Agent: Stephen Regoczei  
Word node: "John"  
Concept node: [PERSON], [JOHN]  
Referent node: #35467829

Now we can form some sortally composite entities:

{#35467829, [PERSON], "John"}

This says that the entity whose surrogate is #35467829 is being considered a person by Stephen Regoczei, and the name of this entity is "John". Furthermore, the concept [JOHN] bears a relation to the concept [PERSON]. The exact nature of this relationship is, however very difficult to state, because there are several different ways of thinking about it. For example, we could say that John is an *instance* of a person. In Sowa's graph notation [Sowa 1984], this would be expressed as

[PERSON: John]

Or we could say

[JOHN] ← (IS-A) — [PERSON]

and leave the meaning of IS-A appropriately vague.

We can continue looking at the relationship between the concept and the referent nodes by noting that the concept [JOHN], when projected onto the referent node, is mapped onto #35467829. We give 'meaning' and 'significance' to the uninterpreted string #3546789 by attaching the concept [JOHN] to it.

We can extend our meaning triangle by establishing coreference links and by listing newly constructed sortally composite entities. These entities are not diagrammed on a vertex; they can be pictured as being located on the 'sides' of the triangle. Adding the relation

Coreference ("John", [JOHN], #35467829) would indicate that these character strings all refer (in a sense!) to the 'same thing', but they are not the same thing! Words, concepts, and referents are different. Much misunderstanding in knowledge acquisition work, and in interpersonal communication, is caused by the faulty use of coreference links.

Sortal composites are *n*-tuples, such as word-concept pairs or surrogate-concept-name triples. Thus we can expand our meaning triangle frame by adding, for example, a slot for composite entities:

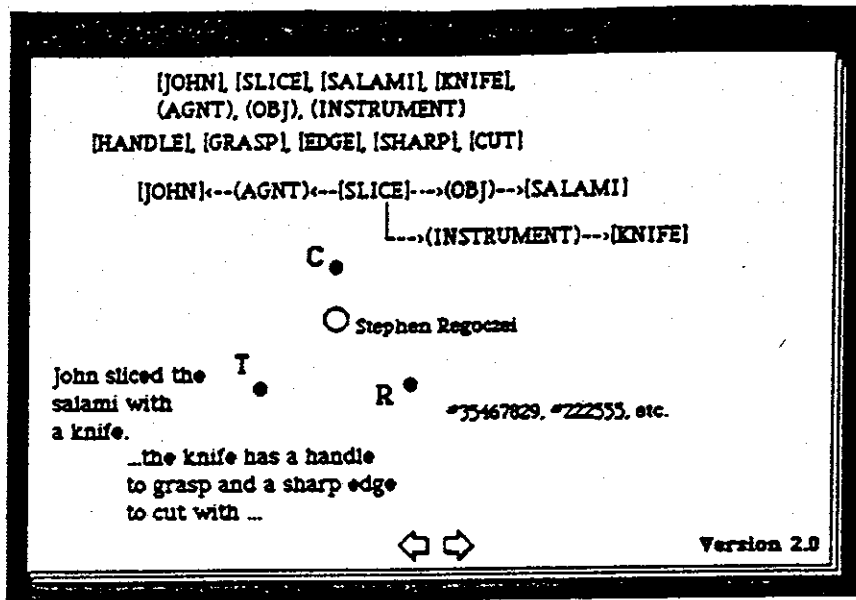


Figure 8: Version 2 of the analysis.

**Composite entities:**

- Word-concept: (<"John", [JOHN])
- Referent-word: (<#35467829, "John")
- Referent-concept: (<#35467829, [PERSON])
- Word-concept-referent: (<"John", [JOHN], #35467829)

Coreference links are created by the analyst as an annotation on the triangle, but the sortally composite entities are construed as being 'out there'.

In the example *John sliced the salami with a knife*, is there a referent for "John" in the NLU approach? Probably not. The sentence was a piece of fiction; its author was talking in a 'pretend' sort of way (cf our own disclaimer above). For a pretend world, there are no surrogates. At best, substitute surrogates are constructible as possible or potential entities by projecting the pretend concepts onto the referent node.

Which entities require surrogates depends on the focus of attention. In the sentence about John's salami-slicing activity, there are few clues about focus and emphasis. Suppose the point to be conveyed were the following:

John knife-sliced the salami, rather than machine-slicing it.

or:

John sliced the salami—with a knife, no less!

The point is that this time around the mental models of the agent are different because the focus on a knife-like action emphasizes the type of action being performed but not the instrument. Which particular knife was used is no longer of significance. So in constructing surrogates for the entities in the object domain, it may or may not be necessary to construct a surrogate such as #222555, specified by the sortally composite entity:

(<#222555, "the knife that John actually used")

Similarly, in contrast to the conventional NLU analysis, changing *a knife to the knife* commits us to very different surrogate structures on the referent node. We would then have to identify the particular knife. Often we are faced with the unknown structure of the world, and either accept the vagueness or have to hypothesize surrogates as projections of concepts generated by an agent. In the next section we discuss techniques for coping with vagueness.

**8 DISCOURSE ANALYSIS IN SORTAL ANALYSIS**

The analyst must pay attention not only to what is said but also to what is not said. She

must decide what seems to be missing and try to fill in the blanks. There are many different kinds of lacunae, each requiring a different treatment. Three kinds are particularly important:

- Recovering missing verb modifiers. The modifiers of a verb are its argument positions: the agent, patient, instrument, time, reason, place, and so on. (Not all verbs allow all kinds of modifiers.) For example, the utterance "The salami was sliced" leaves open the questions of how the salami was sliced and by whom.
- Making presuppositions and tacit assumptions explicit. Utterances frequently presuppose much more than they directly assert. For example, "The sandwich hand sliced the salami" presupposes that there is a person employed as a sandwich hand whose role probably requires further investigation.
- Guessing at information withheld by the informant. Compare these two exchanges:

"Is the system cheap and high quality?"

"Yes."

"Is the system cheap and high quality?"

"Yes, it's cheap."

It is well known that informants will withhold information for good reasons of their own. Ferreting out this information is crucial for knowledge acquisition, and the success of the project is largely dependent upon the analyst's skill.

We plan that SORTAL will have a facility to include notes and additional textual and conceptual material by the analyst to supplement the text uttered by the informant. This is important not only in cases when attention is paid to situations such as those we described, but also in the cases of ellipsis, hard-to-resolve anaphora, metonymy, metaphor, and other uses of figurative language. SORTAL will facilitate the generation of explicit augmentations both at the textual and the conceptual levels, tag them as such, and provide an 'audit trail' of attributions.

Prompting for the filling of lacunae is easier than is commonly thought. For example, in im-

personal phrasing, an inanimate object is substituted for a cogniting agent, such as in this sentence [Regoczei and Hirst 1989a]:

The car smashed through the barrier.

Driven by whom? Was it, perhaps, driverless? If John was driving the car, there is a difference between saying this sentence and saying that:

John smashed through the barrier.

The end result, in terms of the referent or state of affairs in the world, may be the same, but there is a difference at the conceptual level and the issue of volition and culpability may come up. So a distinction has to be noted. In logical terms, the identity of "John" and "the car" may become an issue. The possibility of such apparent anomalies may lead one to consider that the sentence contains figurative language—in particular, metonymy. This kind of possibility may be detected with relatively low-level, syntactic and lexical techniques. The system need not 'know' too much about the concepts and the referents behind "John" and "the car" to tag them for further investigation.

In the current version of SORTAL, some prompting facility will be built in for textual augmentation. These facilities are necessarily limited at present, because the creation of enhanced facilities would depend on two crucial prerequisites. First, research in natural language understanding, especially in discourse analysis and pragmatics, has to be further advanced. Second, the provision of the augmentations is knowledge-based and discourse dependent. We see the generation of such supplementary material as akin to concept cluster attachment. As we argued in an earlier paper [Regoczei and Hirst 1989b], concept cluster attachment is a knowledge-based activity that requires considerable specific knowledge on the part of a cogniting agent. To enhance SORTAL's ability to prompt, we need, in effect, to add small expert-system modules to it that contain expertise regarding the kind of discourse analysis techniques and the kind of missing verb modifiers that are appropriate to prompt for in specific circumstances. We describe these difficulties not only to specify the current limitations of SORTAL, but also to point out the important research directions for the future that are precon-

ditions for further enhancement of the present version.

## 9 A TYPICAL SESSION WITH SORTAL

We will now briefly outline what a typical session with SORTAL would look like, following this example. We want to emphasize that the sequence of interaction, *i.e.*, the user-SORTAL dialogue, is not prewired. Each user can set up the kind of prompting sequences that best suit her cognitive style. There would be a library of these interaction templates available: verbose; brief; superficial analysis; deep analysis; concentrating on the verbal, conceptual or referent node; multiple agents; multiple triangles with joins and unification; multiple domains of discourse for the same agent; emphasizing coreference links, sortally composite entities, or the platonic domain; and so on.

In our case, the *dramatis personae* include the analyst, the supervisor, John, and an impersonal super-agent probably representing the company or the policies and procedures of the company from whose view the 'correct way' of doing things gets formulated.

We have to decide who the 'cogniting' agents are. John may play no greater role than slicing salami, in which case he is not a cogniting agent. On the other hand, if his understanding of whether Genoa salami or summer sausage is to be knife-sliced or machine-sliced matters, then his mental models are to be recorded on an appropriate meaning triangle centered upon him.

As the analysis proceeds, a consensus emerges. For recording this consensus, a consensus triangle, with supervisor and analyst on the agent node is to be created. The population of the concept node in this consensus triangle may be transferred to the policies-and-procedures triangle, as judged or deemed appropriate by a joint decision of supervisor and analyst.

A typical session with SORTAL may start as follows:

*SORTAL:* Establishing meaning triangles.  
What criteria would you like to use?

*Analyst:* Agent.

*SORTAL:* List agents.

*Analyst:* Me, Supervisor, Consensus, Corporate

*SORTAL:* Which triangles are the input triangles?

Here SORTAL displays the frames for the triangle to allow the analyst to indicate which triangles are the input triangles. The analyst uses a mouse to indicate that her own triangle and the triangle of the supervisor are input, the consensus triangle is an intermediate workbench or breadboarding device and the corporate triangle is for the 'objective knowledge' that the knowledge base is intended to capture and present. The analyst indicates that the consensus triangle has joint agents consisting of herself and the supervisor.

SORTAL continues:

*SORTAL:* Which triangle would you like to work on?

*Analyst:* [Indicates supervisor's triangle]

*SORTAL:* Text?

The analyst enters the text in a separate window on the screen and uses a mouse to carve the text into fragments. As she touches various fragments of the green text, yellow versions of the text fragments, indicating the corresponding concepts, are transferred to the concept node. Likewise, she can create blue surrogates, which are then transferred to the referent node. There are two styles of display for SORTAL's main working screen. The first is intended to be more convenient for use by the analyst, while the second, a more triangle-like display, is for reviewing the contents of the fields with the informant; see figure 9.

The supervisor-centered meaning triangle created by the analyst reflects her mental models of the supervisor's mental models. This is an important point. The supervisor may not agree with the analyst's interpretation. If he does, the agreed-upon material should be transferred to the consensus triangle. But usually a discussion takes place about the areas of disagreement. This dialogue should be carefully recorded for future analysis. Before this, however, a version-control procedure is activated.

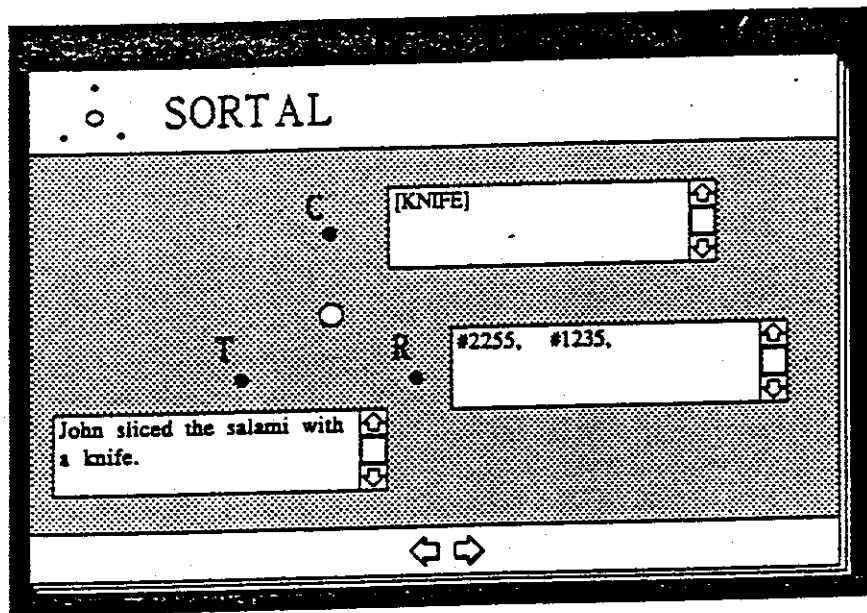
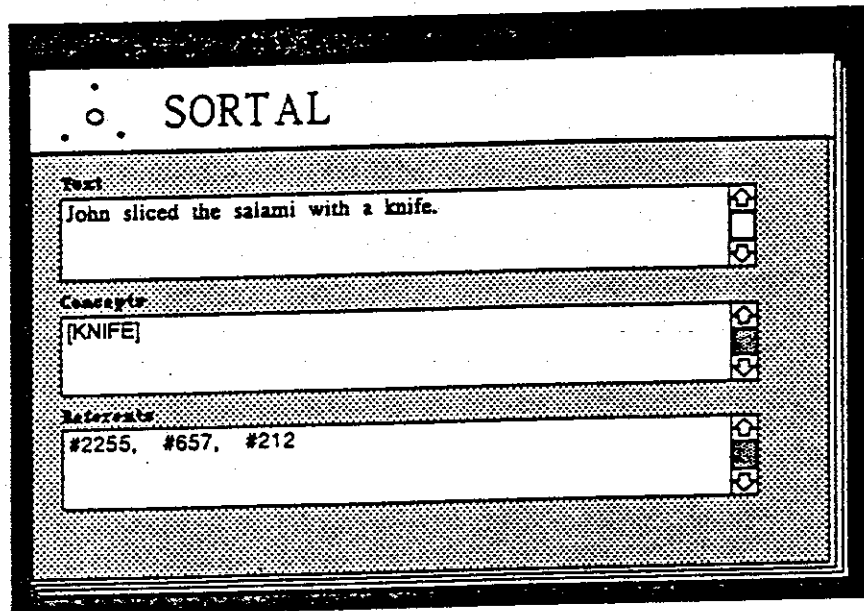


Figure 9: Two styles of display for SORTAL's main working screen.

The meaning triangles are transferred to a version library and are labelled as Version 1.

The analyst inquires further about the details of the situation to which the sentence refers. Eventually, the following comment may arise:

*Supervisor:* No, John wasn't doing it properly. In fact, now that I think about it, it probably wasn't John after all. It was somebody else—I don't know who. [Hence no surrogate is needed.] The point is that hard salamis have to be sliced on the machine, because otherwise we can't control thickness. With softer things like bologna, they can use the knife instead if the machine would have to be cleaned.

The analyst and supervisor are now working towards the consensus triangle, which may contain a rule such as this:

If salami is hard, then use the machine.

If salami is soft, then preferably use the machine; as an exception, one may use a knife if the machine needs cleaning first or would need cleaning afterwards and this would result in an unacceptable time delay.

After further consultation, SORTAL may be used to transfer this text to the word node of the corporate triangle, together with the conceptual translation of it either in conceptual graph notation, frame notation, logical notation, or Telos (see section 10 below). The 'objective' knowledge base would be generated from the corporate triangle by taking the contents of the word node and attaching concept clusters from the concept node [Regoczei and Hirst 1989b].

## 10 IMPLEMENTATION OF SORTAL

A common error in designing software is to distort the architecture by a premature consideration of what can be easily implemented. Thus often the compromise system is the first and only design that ever gets produced. Here and in our other papers we try to distinguish clearly between:

- What *must* be built as the essential core of the system (the "must-have" list).
- What one would also *like* to build (the "nice-to-have" list).

- What *can* be built: given the state of the art and resource limitations.

At present, there are not enough utilities available to implement the SORTAL architecture in a cost-effective way. Basically, we need concept processing rather than word processing. The main capability needed is being able to go back and forth between diagrams and text easily. Hence, ideally we would need a hypertext-, database-, or frame-based knowledge representation facility. Probably the closest approximation would be expert systems shells such as Goldworks or KEE, with an extension whereby one can generate concepts and referent surrogates and also interface with graphics diagramming. Ideally, we would have a database with screen handling and freeform icons and diagramming, with a facility to attach text and concepts to parts of the diagram. Until such utilities become available, we can emulate the implementation of the SORTAL architecture—even using simple word-processing packages, relying upon a user-imposed discipline.

At present, we are investigating the feasibility of several different alternatives for embodying a substantial kernel of SORTAL functions. The alternatives are, in decreasing order of desirability (as we see it at the moment):

1. Hypertext, with a strong graphics capability so that diagrams can be fragmented and concept clusters attached to the fragments. This is a capability that could be also used to fragment sentences and associate concept clusters with the fragments. Icons generated with the graphics facility could be used as surrogates. (There are advantages to using such non-textual surrogates.)
2. A state-of-the-art knowledge representation language such as the Telos system being developed at the University of Toronto [Koubarakis *et al* 1989a, Koubarakis *et al* 1989b].
3. An expert-system shell with the appropriate knowledge-structuring capabilities.
4. An already-existing KA tool that could be adapted for our purposes.
5. An already-existing CASE tool that could be adapted for our purposes.



6. A relational database with a fourth-generation language interface with screen-management capabilities for setting up forms.
7. A text file, under an editor, using certain human-enforced, disciplined notational conventions.

At present, we are investigating options 1, 2, 3, and 7 to establish a range for what is feasible and cost-effective. We are prototyping SORTAL in HyperCard, and planning to do a production implementation in Smalltalk. For the analysis work, we have been using conceptual graphs to record our findings on the concepts node. Conceptual graphs are well suited for knowledge representation at the analysis stage because they are concept-oriented and their surface form is not entirely dissimilar to natural language as actually used by analyst and informant. But for the purposes of processing the population on the concept node, we would want to transfer to a language like Telos.

We can summarize gradually more enhanced implementations of the SORTAL architecture by outlining five levels of practicing sortal analysis:

1. Essentially manual: The analyst draws pictures of meaning triangles on the screen using a tool such as HyperCard. There is no processing of what appears.
2. HyperCard prototyping: Separate fields for the text, concept, and referent nodes, and support for coreference links and sortally composite entities (as seen in figure 7). The contents of fields are available for some limited processing.
3. SORTAL implemented with Smalltalk processing behind a HyperCard user interface: Implementation of populations on the nodes as abstract data types.
4. SORTAL with Telos: To give fuller manipulation capabilities in the concepts field. A conceptual-graphs-to-Telos interpreter is required.
5. Interaction between the natural-language level and the conceptual level: Three-way communication between the text node, conceptual graphs, and Telos, to output knowledge bases in the form of text-concept pairs.

The features of the implementations will be tested on a large body of material gathered by the first author over years of systems analysis, data modelling, and knowledge acquisition work.

## 11 KNOWLEDGE ACQUISITION MODULES FOR KBMSs

Knowledge-base management systems (KBMSs) are systems that assist in the creation, management, and use of knowledge bases [Brodie and Mylopoulos 1988, Schmidt and Thanos 1989]. SORTAL is a tool for building knowledge bases that may be passed on to a KBMS for further management. But there is another, more unified way of looking at things. SORTAL can be considered a part of the KBMS: its knowledge acquisition module.

In considering SORTAL as a KA module for a KBMS, we need to think about the different types of KA modules possible. In particular, we need to distinguish between software *assistants* for KA and *automated* KA modules. The specifications for these two types are very different. The main purpose of a software assistant is to enhance the capabilities and improve the performance of the people using it. The software is not expected to produce conceptual structures or knowledge bases all by itself. The people using it are very much part of the system, and it is this system as a whole that produces the final product. SORTAL is such a software assistant. Its effectiveness is partly to be judged on the basis of how well it enables people to do their jobs. Automated modules, on the other hand, are expected to work reliably, independently of people's input. No human intervention should be required. LOGOS [Regoczei and Hirst 1989b] is an example of such an automated module for the acquisition of knowledge from text.

### 11.1 Processing the results

The output of SORTAL is triangles with a population on each node. Each population can be considered as being of an abstract data type, to be manipulated according to appropriate admissible operations. The population on the text node is to be handled by standard text-processing or natural language-processing packages. The population on the referent node is to

be structured by organizing it into object domains, and structuring these domains by projecting the concepts from the concept node onto the referent node.

The population on the concepts node is like a knowledge base. One form of output, as mentioned before, consists of conceptual structures together with natural language equivalents attached. Another way of handling the population of the concept node is to transfer it into a language such as Telos, thereby making it usable with a KBMS.

To summarize: The population of the text node is text, to be handled as such. The transition from the text node to the concepts node is concept cluster attachment. The population of the concepts node, translated into Telos, is handled by a KBMS. The population of the referent node is structured by projecting from the concepts node, either by forming sortally composite entities or through coreference links. Explanations can be attached to referents by forming referent-text pairs. Furthermore, the comments of the analyst can be attached as text windows to referent surrogates, as they can be to text fragments, conceptual structures, and sortally composite entities.

## 12 CONCLUSION

In this paper, we have discussed ideas concerning sortal analysis, the use of the meaning triangle, and concept cluster attachment. However, the main purpose of this paper is to outline how to put these ideas into action, by embodying them in software tools. We described the requirements for such a tool, SORTAL, described desirable features of the architecture, reviewed the implementation problems, and suggested a feasible emulation of the most vital components of the architecture.

To repeat the main point: The acquisition of knowledge from text relies not only on the understanding of text, but also the understanding of people and the understanding of the object domain. Concept cluster attachment, the hypothesized mechanism for attaching knowledge to text, brings about the generation of a great deal of textual and conceptual material. The recording and organizing of this material requires software assistance.

An additional complication is introduced by taking seriously the 'aboutness' of language-in-use, and hence recording not only concepts but also an adequate inventory of surrogates for entities that populate the referent domain. To make this task feasible, better software tools are to be developed for analysis and conceptual modelling. We believe our research to be contributing to this continuing development.

## Acknowledgements

For discussions on these topics we are grateful to Tom Addis, Bob Carter, Marc Linster, John Mylopoulos, Doug Skuce, and Steve Twine. Our research is supported by grants from the Natural Sciences and Engineering Research Council of Canada, the Trent University Committee on Research, and the Information Technology Research Centre (an Ontario Government Centre of Excellence) through its KBMS project (John Mylopoulos, principal investigator).

## References

- [Brodie and Mylopoulos 1988] Brodie, Michael L. and Mylopoulos, John (editors) (1986). *On knowledge base management systems*. Springer-Verlag.
- [Bullock et al 1988] Bullock, Alan; Stallybrass, Oliver; and Trombley, Stephen (editors) (1988). *The Fontana dictionary of modern thought* (second edition). London: Fontana Press.
- [Grice 1975] Grice, H. Paul (1975). "Logic and conversation." In: Cole, Peter and Morgan, Jerry L. (editors). *Syntax and semantics 9: Speech acts*. New York: Academic Press, 41-58.
- [Halton 1973] Halton, Gerald James (1973). *The-matic origins of scientific thought: Kepler to Einstein*. Cambridge, MA: Harvard University Press.
- [Koubarakis et al 1989a] Koubarakis, Manolis; Mylopoulos, John; Stanley, Martin; and Jarke, Matthias (1989). "Telos: A knowledge representation language for requirements modelling." Technical report KRR-89-1, Department of Computer Science, University of Toronto, January 1989.
- [Koubarakis et al 1989b] Koubarakis, Manolis; Mylopoulos, John; Stanley, Martin; and Borgida,

- Alex (1989). "Telos: Features and formalization." Technical report. KRR-89-4, Department of Computer Science, University of Toronto, February 1989.
- [Regoczei and Hirst 1988] Regoczei, Stephen and Hirst, Graeme (1988). "The meaning triangle as a tool for the acquisition of abstract, conceptual knowledge". *Proceedings, Third Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, November 1988. To appear in *International journal of man-machine studies*, 1989.
- [Regoczei and Hirst 1989a] Regoczei, Stephen and Hirst, Graeme (1989). "Knowledge acquisition as knowledge explication by conceptual analysis." In: Sowa, John F; Foo, Norman; and Rao, Anand (editors). *Conceptual graphs for knowledge systems*. [To appear.] Also published as technical report CSRI-205, Computer Systems Research Institute, University of Toronto, January 1988.
- [Regoczei and Hirst 1989b] Regoczei, Stephen and Hirst, Graeme (1989). "On 'extracting knowledge from text': Modelling the architecture of language users." *Proceedings, Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Paris, July 1989, 196-211. Also published as technical report CSRI-225, Computer Systems Research Institute, University of Toronto, January 1989.
- [Regoczei and Plantinga 1987] Regoczei, Stephen and Plantinga, Edwin Peter Owen (1987). "Creating the domain of discourse: Ontology and inventory." *International journal of man-machine studies*, 27(3), September 1987, 235-250. Reprinted in: Boose, John H. and Gaines, Brian R. (editors). *Knowledge acquisition tools for expert systems* (The knowledge-based systems series 2). London: Academic Press, 1988, 293-308.
- [Schank 1973] Schank, Roger Carl (1973). "Identification of conceptualizations underlying natural language." In: Schank, Roger Carl and Colby, Kenneth Mark (editors). *Computer models of thought and language*. San Francisco: W.H. Freeman, 187-247.
- [Schmidt and Thanos 1989] Schmidt, J.W. and Thanos, C. (editors) (1989). *Foundations of knowledge base management*. Springer-Verlag [to appear].
- [Sowa 1984] Sowa, John F (1984). *Conceptual structures: Information processing in mind and machine*. Reading, MA: Addison-Wesley.
- [Winograd and Flores 1986] Winograd, Terry and Flores, Fernando (1986). *Understanding computers and cognition: A new foundation for design*. Norwood, NJ: Ablex Publishing.