Representing Complex Physical Objects in Memory

by Kenneth Wasserman and Michael Lebowitz

December 1982

CUCS-37-82

This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-82-C-0427.

Table of Contents

Abstract

Researchers in artificial intelligence have proposed and implemented several representation systems for use in computer programs that "understand" natural language input. Noticeably lacking from these systems is a robust and concise method of representing complex physical objects. This paper describes a language-independent scheme for encoding real-world objects in a manner that captures elements of how people describe them. Two major groups of objects are distinguished: "unitary" objects that are described by a single "shape-descriptor"; and "composite" objects that are represented by a frame-based system that focuses on the physical relations that exist among objects. The heart of this scheme is a primitive-based framework that classifies physical relations into three fundamental categories with five possible properties. Our current work on RESEARCHER, a program that employs this scheme while reading patent abstracts, is also discussed.

1. Introduction

Developing a computer program to understand the description of a complicated real-world object such as an "Auxiliary Insulated Roof System" would be a formidable task without the use of an appropriate representation system. This paper will present a language-independent framework that easily encodes this example as well as a wide range of others involving physical objects.

Most recent work on representing cognitive information from natural language input has focused on action-oriented events. Researchers in cognitive science have developed and implemented several schemes for organizing information in order to perform such activities as understanding natural language. Computer programs such as SAM [Cullingford 78], IPP [Lebowitz 80], Ms. Malaprop [Charniak 78] and others [Barr and Feigenbaum 81, Wilks 75] read stories about various topics that are based on human activities. Within the limited domains of knowledge that they deal with they have demonstrated an ability to model most human thought processes while comprehending text.

One common element that these programs embody is their action-based representation scheme; Conceptual Dependency (CD) [Schank 72] or another case-like system, is used as a vocabulary independent means to classify all actions (basically verbs) that occur in natural language. CD has been extended to apply to a wide range of situations.

Although CD is a fairly robust representation system it does not deal extensively with representing physical objects. Object representation using CD has been tried, most notably by Lehnert [Lehnert 78], where she presents a case for the existence of seven object primitives that work somewhat like the primitives in CD. However, her scheme does not attempt to encode the information needed for a physical description of objects; it supplies information about how an object might be used. which is important in understanding action-oriented stories that involve objects.

What most researchers in natural language processing have done in the past, when confronted with physical object representation, is to merely state the name of whatever object is being talked about. In some cases, more information is also supplied, such as the parts an object might contain or its point-by-point graphical representation [Kosslyn and Shwartz 77, Lehnert 78]. In certain restricted language domains, this information may be all that is needed, but in others the data provided by such representations is insufficient for a complete understanding of the input text. For example, technical text dealing with complex objects would require a much more sophisticated representation scheme in order to provide an adequate understanding. Abstracts from device patents is one area in particular need of a good schema; some examples from such texts are given in this paper.

There is a need for a language-independent representation of complex physical objects in memory. A usable representation system is central to any computer program that answers questions, makes generalizations or comprehends real-world situations where physical objects are involved. This paper endeavors to present such a representation scheme and suggest how it can be used. The question of whether the "memory" is human memory is not answered here, but is an important consideration in motivating our research. The primary objective of this work is to develop a useful physical object representation that a computer program can benefit from.

Writing about natural language processing poses a problem in semantics. In particular, this paper has the goal of presenting a computational schema for physical object representation as well as the goal of exploring some elements of how humans use language in order to describe objects. The language employed here has somewhat of a dual meaning in that it is used to convey information about both of these goals. Therefore, the reader would do well to keep in mind the objectives of this paper and not misconstrue the ideas presented as necessarily an attempt to define the way in which humans internally represent physical objects.

One final introductory note is in order. The representation scheme about to be presented is still in flux and is likely to change somewhat as it is used in various computer programs.

The plan of this paper is as follows: An overview of the representation scheme which introduces the major concepts and terms used throughout, is given; the following two sections describe the details of the lowest levels in the schema; the next section pieces together all the parts of the representation by way of an annotated example; finally some conclusions are drawn, and our current research is mentioned.

2. Overview of the representation scheme

Several goals have guided our course in developing a robust and useful representation scheme. Framed based knowledge [Bobrow and Winograd 77, Charniak 78, Minsky 75] has been shown to be a very natural and easy way to handle information while processing natural language input. Organizing cognitive structures as networks of chunks of memory information, each encoded in frame form, has been suggested and implemented by several researchers ([Rieger 76, Schank 80] among others). We have adopted this form of structure for physical object representation and call an individual chunk of memory a *memette*.

Natural language object descriptions can vary enormously in extent. Large *composite* objects which comprise many parts (for example the Columbia Space Shuttle) can be referred to in toto, or the focus of description can shift to a small one-piece, *unitary* object (such as a particular heat shield tile). A complete object representation scheme should be able to handle either of these extremes, or any level of complexity in between, in a consistent and logical manner.

In order to achieve these goals and others, the following memette frame is used (somewhat simplified here):

```
(NAME: name-of-object
 TYPE: unitary or composite
 STRUCTURE: a shape-descriptor if unitary
                        or
            a list of relation records if composite)
```

The NAME is simply the name of the physical object being described, if it is known. The TYPE slot indicates whether this is a single indivisible structure (*unitary*) or a conglomeration of two or more pieces (*composite*). The STRUCTURE field contains either a description of the shape of an object, if it is unitary, or a set of relation records, if it is composite. Shape-descriptors are graphical representations of objects based mostly on visual properties. Relations are the key to this object representation scheme; they are generally binary relations between parts of an object.

Before a more detailed description of shape-descriptors and relations is given we will first examine an example of a memette structure. Consider the following sentence taken from an abstract of a US patent about a computer disc drive [West 82].

Enclosed Disc Drive having Combination Filter Assembly

> **"A combination filter system for an enclosed disc drive in which a breather filter is provided in a central position in the disc drive cover and a recirculating air filter is concentrically positioned about the breather filter."**

The memette structure for this description might look like:

```
(NAME: enclosed-disc-drive-with-filter
 TYPE: composite
 STRUCTURE: ((INSIDE-OF disc-drive enclosure)))

(NAME: enclosure
 TYPE: composite
 STRUCTURE: ((ON-TOP-OF cover case)))

(NAME: case
 TYPE: unitary
 STRUCTURE: (box open-on-top))

(NAME: disc-drive
 TYPE: composite
 STRUCTURE: unknown)

(NAME: cover
 TYPE: composite
 STRUCTURE: ((MIDDLE-OF breather-filter cover)
             (SURROUNDED-BY breather-filter
                            recirculating-air-filter)))

(NAME: breather-filter
 TYPE: unknown)

(NAME: recirculating-air-filter
 TYPE: unknown)

                    (example 1)
```

The general idea to glean from this example is that each memette represents a small chunk of memory which is connected to other pieces via the physical relations that exist among objects. It should be noted that some of the information encoded here is not stated explicitly in the text. For example, the *case* is TYPEd as a unitary memette; since virtually nothing was said about the enclosure, this information was assumed by the reader. The structure of the *case* is assumed to be box-shaped and open on top (this fact was implied by the existence of a *cover*). Likewise, the *disc-drive* itself is considered to be composite, although this information would have had to be acquired outside the context of this sample.

The implication that the *case* is box-shaped and open on top could actually have been made by reference to a stereotypical *case*. Stereotypes are important and useful concepts for understanding unknown objects. In order to process new information about an object, it is helpful to know what data about that object can be expected. A stereotype (or prototype) is a convenient means by which to convey this information. Although stereotypes are a very important piece in the grand scheme of physical object understanding, one might even say that all memettes and relations are stereotypes in some sense, they will not be mentioned further, so that we can concentrate on the representational issues.

Four relation records are used in this small memette structure: INSIDE-OF, ON-TOP-OF, MIDDLE-OF and SURROUNDED-BY. A relation record generally consists of the name of the relation followed by the subject of the relation and then the object. The only exceptions to this format is if a non-binary relation is used or if multiple subjects or objects are referenced. Multiple subjects and/or objects are easily accounted for if a list of memettes is specified in place of a single one. The only important example of a non-binary relation seems to be phrases that mean between. Thus expressions such as, "the bacon is between the lettuce and the tomato" can be special cased, and we need not consider them when discussing relations.

The name given to each relation is actually a reference to another frame which contains an explicit definition of that relation. This relation frame is a vocabulary-independent way of classifying any physical relation description. A detailed description of how relations are classified is given later in this paper.

Unitary memettes do not contain any relation records under their STRUCTURE property; instead they have a single shape-descriptor. "Box open-on-top" was given as the shape-descriptor of the *case*. This is not a particularly functional piece of information, as stated, and there is an obvious need to codify shape-descriptors. Some ideas for arriving at a useful system are given in the next section.

3. Shape-descriptors

A shape-descriptor is an abstract representation of the physical form of a unitary object. There are two ways, useful in natural language processing, of describing physical shapes. The first and most obvious way is to form some sort of abstract visual image of the object. The other possibility would be to develop a symbolic representation scheme that provides object shape descriptions by linking together more fundamental representations.

This second method is somewhat like the composite object representation introduced earlier. That is, by "knowing" the relations connecting object parts one can infer the overall form of the total object. This general method for determining unitary object shapes has been employed by several researchers in the past [Winston 77].

An abstract visual representation of a unitary object has a strong appeal to it. Probably the most complete work along this line was done by Kosslyn and Shwartz [Kosslyn and Shwartz 77]. Basically their model of visual processing breaks objects into a point-by-point polar coordinate representation. They use lists of points to identify unitary objects and compare these lists in order to compute the composite object form. In our object representation scheme, only a single list of points (corresponding to a single unitary memette) need be used as a shape-descriptor.

Most likely, the best shape-descriptor format would be a combination of these two methods. The visual aspect would help in generating or recognizing information supplied by diagrams accompanying the text (this is of particular interest in reading complex patent abstracts). Symbolic encoding of shape-descriptors could be advantageous in parsing natural language text because the information would already be in a form similar to the organization of composite objects. Therefore some of the same machinery used for "understanding" composite objects could be applied to unitary objects.

Shape-descriptors have another purpose in this representation scheme. Aside from describing unitary objects they can also serve as a way to describe the shape of an enclosure. In the memette structure shown in example 1 of the previous section, a "box open-on-top" was given as the description of the *disc-drive case*. The relation

INSIDE-OF described that the *disc-drive* was enclosed by the *case*. The representation of the relation INSIDE-OF contains a description of the shape of the boundry between the *disc-drive* and the *case*. The boundry description is in the form of a shape-descriptor. Exactly how relations, such as INSIDE-OF, are represented by a frame structure will now be presented.

4. Relations

In order to develop a concise system to represent arbitrary physical relations, it must first be recognized that three major classes of relations exist. In English (and presumably most other natural languages) descriptions of relations between objects may or may not assume that an observer is present. Observer independent relations, those which can be described without regard to either the position of an observer or the scale at which the relation holds, are termed *absolute* relations. Those that are completely dependent on an observer's position and the scale at which the relation is viewed are called *subjective* relations. A third class of relations, *subjective-absolute*, is needed to cover those descriptions that are independent of the observer's position but depend on the scale of description. The table given below summarizes these classes and gives some sample relations that fit each class.

	RELATION CLASS		
	ABSOLUTE	SUBJECTIVE-ABSOLUTE	SUBJECTIVE
observer's position:	independent	independent	dependent
scale of view:	independent	dependent	dependent
example phrases:	on top of held by	between inside of	to the left of next to

To clarify the subjective-absolute classification, consider the phrase "inside of". A potential car buyer, standing around a new car show room, would describe the car seats as being inside the car. However, if he's sitting in the driver's seat, he is unlikely to say that "inside the car" is the way to describe what he is sitting on. Although the position of an external observer is not important here, the focus or scale of description is crucial.

While the potential buyer is sitting on the seat, the salesperson would be accurate in describing the customer's position as "sitting on" the seat, no matter where the salesperson might be located. Thus "sitting on" is considered to be an absolute relation, in which neither the observer's position nor scale of view is relevant. No matter how close or far from the car seat an observer might be, he must conclude that the driver and the seat are in contact.

The two kids on the back seat of the car claim the John is "to the left of" Mary. When the parents turn around to face their kids they determine that John is "to the right of" Mary. Since both observations are correct there must be a dependency on the observer's position, when describing this relation. Therefore to codify a subjective relation a frame of reference must also be specified (more about this later).

A fourth possible classification of relations, would be to have relations which are independent of the scale of view and dependent on the observer's position. This possible class of relations seems to have no basis in the reality of human descriptive terms.

These three fundamental classes of relations provide a gross distinction among relational phrases and concepts but are not nearly sufficient for a concise and useful representation scheme. To form a language-independent representation of any relational phrase, five properties of a relation are needed in addition to the fundamental class. These five properties are outlined below:

property	description	value(s)
distance	is used for relations that refer to disjoint objects. (e.g. near, remote)	a single integer from 0 to 10. 0 - close, 10 - far
contact	is used to describe the degree to which objects are in contact with each other. (e.g. touching, affixed)	a single integer from -10 to +10. -10=very close +10=very loose
location	is used to indicate in which direction an object is located relative to another. (e.g. above, left)	a 2D or 3D angular identification along with a reference frame indication.
orientation	is used to describe the relative orientation of two objects. (e.g. parallel, perpendicular)	a 2D or 3D angular identification.
enclosure	is used for relations which describe objects, where one is either fully or partially enclosed by another. (e.g. encircled, cornered)	full or partial plus a shape description of the interface between the enclosed and the enclosing objects.

note: The example words given above have been chosen to illustrate the role of each property and are not necessarily fully described by that one property alone.

Before a description of each of these properties and examples of its use are given, a few points about this representation scheme should be noted. Firstly, not every relation has every one of these five properties. In fact most relations are adequately described by only two of these properties. Secondly, the fact that a relation has a particular property is often more important to consider than the value that this property takes on. In particular the scale values for the *contact* and *distance* properties are rather arbitrary, however their relative values have meaning.

Distance is probably the simplest of the five properties listed. It is used to indicate that two objects are separated from each other by some length. Because there seems to be an unlimited number of ways (in English) to describe distances, some method of reducing this range is needed. By forcing all distance descriptions into one of eleven possibilities, distance relations become more manageable. The eleven possibilities are taken to be the integers from 0 to 10. In some cases, particularly in technical prose, the actual distance in some specific measurement unit (e.g. inches or meters) might be given. If this is important data the slot values for the distance property could be expanded to allow for this information to be explicitly inserted.

A zero valued distance property would be used to indicate a relation such as "microscopically close to". On the other extreme, "astronomically far from" would certainly be a 10. A more mundane word, like "nearby", would register a 4, perhaps.

The *contact* property is much like the distance property in that they both describe relative degrees of closeness by using an integer value. It would be extremely unusual if a single binary relation (in a natural language) required both contact and distance properties to represent it. Thus aside from futuristic forces, like *tractor beams*, they are mutually exclusive properties of a relation.

Contact values arbitrarily range between -10 and +10, allowing for 21 degrees of contact. The bond formed between two oppositely polarized magnets could possibly be valued as CONTACT=-9 while a good quality record turntable has CONTACT=+8 between the tonearm and the record (while it is being played). The motivation behind using both positive and negative values for the contact property comes from the analogy to the distance property. The larger the number the further away an object is, even though it is still toughing another object if contacted. Negative numbers indicate that the objects are being forced together by some means.

The *location* property is very often used to define relations which describe objects in everyday settings. Phrases like, "its the building on the left, when you face the church" and "write your name on top of the paper" are good examples of the use of this property. In the first example both the relative direction ("left") and the reference frame ("facing the church") are explicitly given. The second phrase has implicit in it that the student has a piece of paper with the normal orientation, placed in front of him.

The appropriate values for a location property are therefore a reference frame along with an angle ("left" would be 180 degrees, "top" would be 90 degrees). The frame of reference is important because a person standing at the church's front door and looking out would find the building to the right (0 degrees). Angular values, in 2-dimensions, can be any number from 0 to 360 degrees. Thus phrases like "below and to the right of" might imply an angle of 315 degrees, depending on the reference frame.

3-dimensional angular values are also possible fillers for the location property. A solid angle would be specified along with a reference frame that must provide more information than a 2-dimensional description would. 3-dimensional relational descriptions are fairly rare but they do exist; for example "the knob is in front of, just below, and to the left of the radiator". Admittedly this is a somewhat contrived example that could well be represented as several separate 2-dimensional relations, but it illustrates the generality of the location property.

In the example, "write your name on top of the paper", it was mentioned that normal *orientation* of the piece of paper was implicit. The orientation property refers to the rotational disposition of an object about its own axis, relative to another object. What this means is that if we're talking about railroads and use the phrase "the tracks are perpendicular to the ties", the orientation property of this relation would get a value of 90 degrees.

The orientation property is not used much in day-to-day language, but is quite useful in specifying relations in technical prose. For example, a phrase such as, "the barrier strip running alongside the transformer" would use an orientation value of 0 degrees to express the parallelism. As with the location property, orientation values can be any angle between 0 degrees and 360 degrees. 3-dimensional values are also possible, but not common in natural language descriptions.

The remaining property, *enclosure*, is very different from the other four in that its value can be a complicated shape-descriptor. As mentioned earlier, the shape-descriptor used in this case specifies the shape of the boundry between the enclosed and the enclosing objects. For example, if "the tire encircles the wheel" then a shape-descriptor of a circle would be the appropriate value for the enclosure property. Another piece of information provided is whether the enclosure is a *full* one or only a *partial* enclosure, as in the case of "a hand grasping a baseball". Although the full versus partial information can be (and is) inferred from the shape-descriptor, it is handy to have this fact readily available because the type of the enclosure can be easily deduced from it.

To help see how these properties fit together into a relation frame, we will consider several examples.

Example 1 (shown earlier) uses the relation INSIDE-OF. "Inside" can take on several possible meanings, but in the context of this patent abstract (and because we assumed the *case* was box-like) we know that the *disc-drive* is enclosed in the *enclosure*. Furthermore, from our stereotypical knowledge of disc drives, we can conclude that the *disc-drive* is probably not in direct contact with the *enclosure*, but is connected to it by some spacing device (which will be ignored here). Thus we could derive the frame slot fillers as follows:

```
(REL-NAME: inside-of
 CLASS: subjective-absolute
 ENCLOSURE: (full box-shape)
 DISTANCE: unknown
 subject: disc-drive
 object: enclosure)
```

In practice, a relation frame need not be given a meaningful name; it is only important that the correct correspondence be maintained between the memette and the relations that it employs. The subject and object slots are only included in this example for ease of reading (the positions of the subject and object in the memette STRUCTURE slot contain this information).

Another relation used in example 1, ON-TOP-OF, requires the use of two different relation property slots.

```
(REL-NAME: on-top-of
 CLASS: absolute
 CONTACT: unknown
 LOCATION: (side-view  90 degrees)
 subject: cover
 object: case)
```

The information embodied in this relation frame is a good example of implicit knowledge. There is no data in the sample text to help in processing what ON-TOP-OF means, in fact the relation ON-TOP-OF was only implied by the use of a *cover*, and was not explicitly mentioned. It seems quite natural to think of one object being on top of another when looking from a side-view. However, the frame of reference used in the LOCATION slot could have been from another perspective.

Note that in both of these relation frames a value of "unknown" was used. As stated before, the existence of the property is often of greater importance than its value. In each of these cases, since no reference to either the degree of distance or contact was made, "unknown" was used instead of picking an arbitrary value to fill the property slot with.

5. Putting it all Together

In order to get a better feel for the whole information structure that this schema encodes we will analyze a complex real-world example. This text is taken from an actual US patent entitled "Auxiliary Insulated Roof System" [Carlson and Brissey 81] and is written in patent legalese. Although patents are not written in everyday English, they provide complicated descriptions of complex physical objects, and are thus a good test bed for this representation scheme.

The text for this example will be given interspersed with its resultant representation. This should give the reader an indication as to how the parsing process might proceed, as well as allow for some comments on the encoded frames.

"In combination with a building structure including

At this point we are expecting parts of a building which is the top-level memette.

opposite upstanding sidewalls, opposite upstanding end walls and a roof surface

Now we know some of the parts and something about their relative orientation. That is, the walls oppose each other in two sets (end walls and sidewalls). We now hope to find some indication of how this structure is connected up so that we can build the relation records.

operatively connected to and extended between upper ends of said sidewalls and end walls,

We can now build a preliminary memette structure based on the information provided.

```
(NAME: building
 TYPE: composite
 STRUCTURE: ((STAND-OPPOSED sidewall#1 sidewall#2)
             (STAND-OPPOSED endwall#1 endwall#2)
             (SPAN-CONNECT-UP roof-surface
                              (sidewall#1 sidewall#2
                               endwall#1 endwall#2)))

(NAME: end-wall
 TYPE: unknown)

(NAME: side-wall
 TYPE: unknown)
```

The relation frames look like:

```
(REL-NAME: stand-opposed
 CLASS: subjective-absolute
 DISTANCE: unknown
 ORIENTATION: 0 degrees
 LOCATION: (side-view  180 degrees))

(REL-NAME: span-connect-up
 CLASS: absolute
 CONTACT: unknown
 ORIENTATION: (side-view  90 degrees))
```

The reader may have noticed that only one memette named *sidewall* has been created, however the text indicates that two *sidewalls* are present. In our representation we use the notation #1 and #2 to indicate two separate instances of the same kind of object, thereby conserving memory space.

So far this representation is a simple *building* with four walls and a roof. We have no information about the structure of the end walls, the sidewalls or the roof.

an auxiliary insulated roof structure comprising

Now we have hit the reason for this patent to have been granted (probably). The "auxiliary insulated roof" is undoubtedly the same one mentioned in the title and we are about to find out about its structure.

a generally continuous panel structure of insulation material, support means interposed between said panel structure and roof surface for maintaining said panel structure in position on said roof surface,

We can now modify our top-level memette (i.e. *building*) to reflect the fact that a roof structure is what's on top, not simply a roof surface. However we do not yet know how it is connected to the building. We can at this point define new memettes that represent the roof structure.

```
(NAME: roof-structure
 TYPE: composite
 STRUCTURE: ((BETWEEN-TOUCH panel-structure
                            support-means
                            roof-surface)))

(NAME: panel-structure
 TYPE: unitary
 STRUCTURE: (continuous))

(NAME: support-means
 TYPE: unknown)
```

note: BETWEEN-TOUCH is a special case relation and is not shown here

It is apparent from comparing the memette representation and the source text that some information has been lost in the encoding process. In particular, the composition of the panel structure was said to be "of insulation material". A complete object representation system should indeed capture this fact. We have neglected such attributes of objects in this paper, but in our actual computer implementation, this type of information is stored under a slot called, PROPERTIES, in the memette frame.

It is also important to maintain a slot for holding the PURPOSE of the memette's use. Thus the purpose of the *support-means* would be for "maintaining said panel structure in position on said roof surface". Of course the PURPOSE slot does not have this data in raw form, it is represented in much the same manner as the STRUCTURE information is.

The next clause provides the necessary information to allow us to connect up the *roof-structure* with the rest of the parts of the *building.*

> **and fastening means secured to said panel structure and building sidewalls for fixing said panel structure in position on said roof surface,**

What we now know is that there is a fastening device (means) that connects the building's sidewalls to the panel structure. Thus we should modify the top-level memette to indicate this. The new *building* memette looks like:

```
(NAME: building
 TYPE: composite
 STRUCTURE: ((STAND-OPPOSED sidewall#1 sidewall#2)
             (STAND-OPPOSED endwall#1 endwall#2)
             (SPAN-CONNECT-UP roof-surface
                              (sidewall#1 sidewall#2
                               endwall#1 endwall#2))
             (SECURED fastening-means panel-structure)
             (SECURED fastening-means (sidewall#1 sidewall#2)))

(NAME: fastening-means
 TYPE: unknown)

(REL-NAME: secured
 CLASS: absolute
 CONTACT: -7)
```

One item to take note of here, is that a value of -7 was assigned to the CONTACT property of SECURED. This is a somewhat arbitrary value but it does convey the idea that securing something implies fairly tight, forced contact of some kind.

If we continue reading we find some information that further refines our memette structure.

> **said support means comprising a plurality of insulated support blocks, ..."**

This results in an improved representation of the *support-means*.

```
(NAME: support-means
 TYPE: composite
 STRUCTURE: ((unknown (many support-block))))

(NAME: support-block
 TYPE: unknown)
```

The patent goes on to describe what each *support-block* is made of. As can be seen from this example, a tree-like memette structure is created incrementally while reading the input text. A single memette can represent a large complex physical object (such as the *building*) or a small component of a *support-block*.

6. Conclusions

The preceding sections have described the format and use of a robust representation scheme for physical objects. All of these concepts have been incorporated into a program designed to read patent abstracts, called, RESEARCHER. One aspect of the functioning of this program is to parse the input text into memette structures that are part of a large tree-like network. The network contains the information supplied by many patents and generalizations based on this data. Ultimately RESEARCHER will also build new concepts of objects formed from the generalizations it makes about real-world physical objects.

As mentioned earlier, this information is more than just structural descriptions of physical objects. The purposes behind the necessity of an object are encoded as well as the features (i.e. physical attributes) an object might have. Furthermore, data used in structuring and maintaining the memette network is also present in each memette.

This paper has used examples of the type that RESEARCHER reads. They are a rather extreme case of complex object descriptions in that patent abstracts tend to embody more information about how physical objects are structured than do most other kinds of prose. Therefore, it is reasonable to conclude that a system capable of understanding such complex objects should also do well with less complicated ones.

Text samples with less complex object descriptions are probably that way because they are concentrating on things other than physical objects. If these other things are action-oriented events then it might be that an action-based scheme would be the best system in which to represent them. However, a mix of a CD-like system and our object representation scheme could provide a more complete understanding of the input text. Therefore, one important avenue of research would be to work out the necessary connections for merging these two schemes and possibly others.

There is still much work to be done on refining and expanding the system presented in this paper. The exact form of shape-descriptors needs to be worked out (with particular attention to incorporating pictorial information from accompanying diagrams). In some domains of natural language processing, it might be helpful to allow for different values in the property slots of relation frames. For example, allowing the DISTANCE property to store exact lengths of the space between objects.

We have presented a frame based system in which complex physical objects can be represented. In addition the physical relations among objects are encoded in a language independent scheme of the same flavor as Conceptual Dependency.

RESEARCHER uses this representation system as the backbone of an integrated understanding system designed to deal with the domain of patent abstracts.

References

[Barr and Feigenbaum 81] Barr, A. and Feigenbaum, E.A.. *The Handbook of Artificial Intelligence.* William Kaufman, Inc., Los Altos, Calif., 1981.

[Bobrow and Winograd 77] Bobrow, D.G. and Winograd, T. "An Overview of KRL, a Knowledge Representation Language." *Cognitive Science 1* (1977).

[Carlson and Brissey 81] Carlson, D L and Brissey, D C. Auxiliary Insulated Roof System. US patent office #4,288,951 on 15 September 1981. Assignee: Scientific Applications Incorporated

[Charniak 78] Charniak, E. On the use of framed knowledge in language comprehension. Tech. Rept. 137, Yale University Department of Computer Science, 1978.

[Cullingford 78] Cullingford, R. Script application: computer understanding of newspaper stories. Tech. Rept. 116, Yale University Department of Computer Science, 1978.

[Kosslyn and Shwartz 77] Kosslyn, Stephen M. and Shwartz, Steven P. "A Simulation of Visual Imagery." *Cognitive Science 1* (1977), 265 - 295.

[Lebowitz 80] Lebowitz, M. Generalization and memory in an integrated understanding system. Tech. Rept. 186, Yale University Department of Computer Science, 1980. PhD Thesis

[Lehnert 78] Lehnert, W.G. Representing Physical Objects in Memory. Tech. Rept. 131, Yale University, 1978.

[Minsky 75] Minsky, M. A framework for representing knowledge. In P. H. Winston, Ed., *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975.

[Rieger 76] Rieger, Chuck. "An Organization of Knowledge for Problem Solving and Language Comprehension." *Artificial Intelligence 7* (1976), 89 - 127.

[Schank 72] Schank, R. C. "Conceptual Dependency: A theory of natural language understanding." *Cognitive Psychology 3*, 4 (1972), 532 - 631.

[Schank 80] Schank, R. C. "Language and memory." *Cognitive Science 4*, 3 (1980), 243 - 284.

[West 82] West, T H. Enclosed Disc Drive having Combination Filter Assembly. US patent office #4,329,722 on 11 May 1982. Assignee: Priam Corporation

[Wilks 75] Wilks, Y. "An Intelligent Analyzer and Understander of English." *Communications of the A.C.M. 18* (1975), 264 - 274.

[Winston 77] Winston, P. H. Scene Description. In P. H. Winston, Ed., *Artificial Intelligence*, Addison Wesley, 1977, pp. 227 - 234.