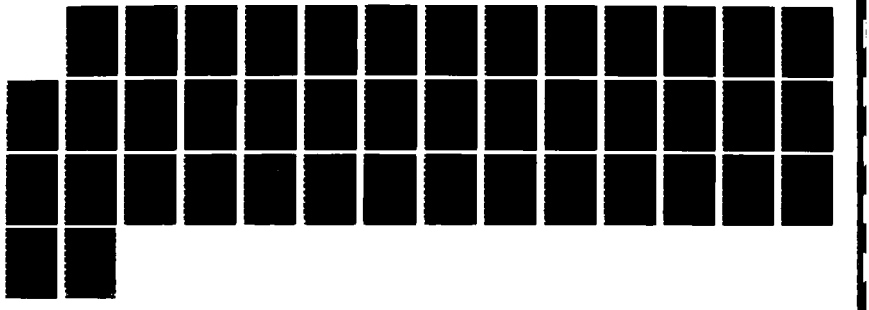
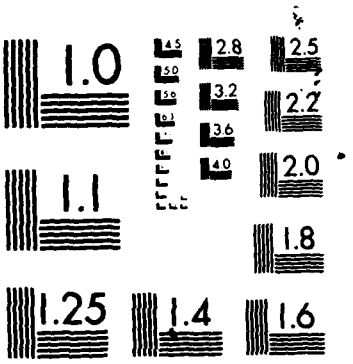
AD-A172 909 SOME PROBLEMS AND PROPOSALS FOR KNOWLEDGE REPRESENTATION(U) CALIFORNIA UNIV BERKELEY DEPT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES 1/1

UNCLASSIFIED R WILENSKY 1984 N00014-80-C-0732 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Productivity Engineering in the UNIX† Environment

Some Problems and Proposals for Knowledge Representation

Technical Report

S. L. Graham
Principal Investigator

(415) 642-2059

"The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government."

Contract No. N00039-84-C-0089

August 7, 1984 - August 6, 1987

Arpa Order No. 4871

†UNIX is a trademark of AT&T Bell Laboratories

Some Problems and Proposals for Knowledge Representation*

Robert Wilensky

Department of Electrical Engineering
and Computer Science
Computer Science Division
University of California, Berkeley

1. Introduction

Knowledge representation is widely regarded as a central problem in artificial intelligence. However, there appears to be no convergence of opinion as to the form a knowledge representation system should take, the principles it should embody, or even what its goal should be. While progress in the past decade has led to a number of interesting theories and useful programming formalisms, this research has also raised doubts about the adequacy of the foundations of many of these ideas (e. g., see Brachman 1979).

In this paper, I present some observations about the knowledge representation schemes now in common use. Some of these observations are critiques of these schemes, or extensions of critiques made by others. To remedy some of these problems, a new theory of knowledge representation is proposed. The theory attempts to encompass representational ideas that have emerged from different schools of thought, in particular from work in semantic networks, frames, frame semantics, and Conceptual Dependency.

In many cases, the problems and solutions described herein have already manifested themselves in other schemes. To the extent they have, this paper should be viewed as a codification of ideas currently in the field. However, I believe that the full implication of these developments has yet to be realized. When one follows them to their logical conclusion, a significantly different view of knowledge representation emerges.

The theory proposed here has a number of salient characteristics: It endorses a proliferation of concepts, each represented as a distinct entity. The theory is uniform with respect to different conceptual domains. Nevertheless, the representational scheme described by the theory attempts to meet certain desirata for a meaning representation. We describe these criteria as principles of *adequacy, interpretability, uniformity, economy,* and *cognitive correspondence.*

Motivated by these principles, the theory eliminates the frame/slot distinction found in frame-based languages (alternatively, node/link distinction found in semantic network-based systems). In its place is a new notion called the *absolute/aspectual* distinction. In addition, the theory incorporates as representational entities notions reminiscent of natural language metaphoric and metonymic relationships. This is done through a mechanism called *views.*

*This Research was sponsored in part by the Defense Advanced Research Projects Agency (DOD), ARPA order No. 4871, monitored by the Space & Naval Warfare Systems Command under contract N00039-84-C-0089 and by the Office of Naval Research under contract N00014-80-C-0732.

As I will attempt to demonstrate, the theory allows for the representation of some ideas that in the past have only been represented procedurally, informally, or not at all.

2. Principles of Representation

Before we can discuss the merits of various representational systems, we need to have some understanding of what it is that a representational system should achieve. Usually, we think of a representational system as a language. As such, principles that have been stated about representational systems have often been cast in terms of a properties of a representational language's syntax and semantics. Some principles that constrain a language's syntax and semantics have been stated by others. Let us begin by reviewing a version of such principles:

1. Epistemological Adequacy

First proposed by McCarthy and Hayes (1969), principle states that the more our language lets us express, the better off we are.

2. Interpretability

The inverse of epistemological adequacy is interpretability. This means that it is desirable for as many expressions of the language to be meaningful as possible. The idea here is that we want anything we can say in the language to have some interpretation (although, of course, it might be *incorrect*).

3. Uniformity

The best knowledge representation system is one that allows the expression of widest variety of objects within it. The way in which it does so must depend minimally on the content which is being represented. This is the criterion of *uniformity*, stated by researchers as far back as Quillian (1968), and elaborated upon recently by Maida (1984).

This principle appears to be rather well accepted today (although apparently, not universally. See for example the Krypton system of Brachman, Fikes and Levesque (1983)). Suffice it to say that the extreme opposite position is "rabid proceduralism", i. e., the idea that there are no interesting generalizations beyond those of list processing (or some such). In contrast, the uniformity lobby promotes a declarative knowledge base and a conscientious interpreter.

The relative complexity of the interpreter and the representation is what is at stake. Will the same interpreter handle all knowledge, or must specialized interpreters be proposed? If so, how far from rabid proceduralism (i. e., the ultimate in specialized interpreters) can we get? While the uniformity position still appears promising, and is adopted here, we have no a priori guarantee that it is correct. In the worst case, this position simply degenerates into proceduralism, so adopting it appears to be the correct research strategy.

4. Economy

We should prefer one representation over another because it is more economical, according to some metric. For example, one formalism for expressions might be able to express the same facts

more simply, or express them in a form in which they are more amenable to efficient computation, than some other formalism. This principle is an extension of the principle of *heuristic adequacy* of McCarthy and Hayes (1969).

Useful as these principles are, they omit a great deal. In particular, they largely neglect the fact that our representational languages are more like natural languages than we are usually willing to admit. Along with a syntax and a semantics, a natural language comes with a set understandings of how that language is to be used. For example, to use a natural language, we need to be guided by principles like "say that which you believe will communicate what you want to express", and "be concise". Without such principles, we would have no notion at all of what to do with a language.

A representational system, it seems, has much the same property. It consists of both a language, and a set of guidelines for expressing ideas in that language. The language is usually relatively easy to formalize. It may comprise categories like predicates, arguments and quantifiers, or frames and slots, or nodes and links, and may or may not presume some vocabulary (particular quantifiers, a frame for "thing", the link "ISA", a set of primitives, etc.). And we can state principles such as those above that may help us evaluate a langauge.

The set of guidelines on how that language is to be used is just as much a part of a representation system, although it is rarely made explicit. For example, a frame-based language suggests that, to use it properly, we should represent certain kinds of concepts as frames, and aspects of those concepts as slots of those frames. Thus the frame for "person" usually is taken to have a slot for "name", but not, say, for "asparagas". This is obvious not because of any property of the representation language, but because of some unstated intuition about the nature of concepts like "person", and the relation of such concepts to the representational framework.

Likewise, most semantic network systems make strong assumptions about what will be a node and what will be a link. These might be verbs and case relations, respectively. Similarly, in predicate calculus, certain ideas seem to be better candidates for predicates and others for arguments. "John left" is invariably represented as "LEFT(JOHN)", and not, say, "JOHN(LEFT)". What rationale there might be for doing so is not part of the language per se.

Sometimes an effort is made to explicitly present usage guidelines. For example, Conceptual Dependency (Schank 1975) has the explicit guideline of canonical form. The particular vocabulary supplied with the language is then justified in part by its conformance to this constraint. In addition, the Principle of Economy applies to representation language use as well. Different sets of expressions within a given formalism may be favored over one another because one set is simpler or more efficient to work with, etc.

In general, though, the principles that determine representation language use are much harder to articulate than those of representation language adequacy. Indeed, they are almost always left on an intuitive level. "leave" just seems like a much better predicate than does "John". It seems reasonable that we have frames for person, places, and things.

Underlying these practices appears to be an important principle. This is that the choice of a particular usage of representation is motivated by how one conceives of that which is being represented. For example, if one conceives of certain entities as individuals (whatever those are), then these should be denoted by constants in one's theory. Of course, one need not do so to have a legitimate model. But it seems that one must do so to have a compelling one.

I term this idea, that the choice of the content put into a representation language presupposes a theory of cognition, as the principle of *cognitive correspondence*. More precisely, we can state this principle as follows:

The Principle of Cognitive Correspondence:

A particular representation for a particular item must be supported by its correspondence to how that item is cognized.

Cognitive Correspondence provides a justification for having predicates corresponding to verbs, or constants corresponding to individuals, or frames for people, places and things. The justification is that such a correspondence captures a underlying cognitive reality. Of course, stating that something is cognitively real does not make it so – additional arguments, empirical or otherwise, must be supplied to support any such particular conjecture. Nevertheless, it is an appeal to this principle that seems ultimatly decisive.

More generally, there may be different kinds of meanings we propose, each of which must correspond to a difference in our representation. For example, consider sense/reference distinctions. Suppose that the phrases "the third man on the left", "her", and "the Morning Star" are coreferential with "John Smith", "Susan Underhill" and "the planet Venus", respectively. If each pair is understood as being coreferential, then, according to Cognitive Correspondence, there must be some element of our representation that denotes this common meaning. However, understanding a phrase's sense is a quite different from comprehending its referent. If we believe that a sense is comprehended en route to determining the referent, then, again according to the Principle, we are required to have another, different representation for each member of each pair. Each of these representations would denote a phrase's sense.

As this example suggests, Cognitive Correspondence is a rather powerful constraint. Consider how this principle would be used in constraining the possible representations for the meaning of natural language utterances. Suppose we want to represent a sentence like

(1) Jan give Lynn a beating.

Intuitively, this seems to share a large meaning component with some phrases involving "hit", such as

(2) Jan hit Lynn repeatedly.

This is true because one *cognizes* the first phrase as referring to hitting rather than giving. Thus, by the Principle of Cognitive Correspondence, the representation corresponding to (1) and (2) should be similar, but both quite different from that for many other phrases involving the verb "give".

Similarly, consider the following sentence:

(3) When John visited Mary at the hospital, he took her flowers.

Most readers understand (3) to mean that John brought Mary flowers, rather than that he took her flowers away from her. Thus, to represent an understanding of this sentence requires some term that differentiates one sense of "take" from the other.

One consequence of this principle, then, is that it is insufficient to represent verbs such as "give" and "take" by entities that correspond directly to them. In other words, a meaning representation must not possess the same ambiguities as does natural language surface text.

2.1. Canonical Form

The Principle of Cognitive Correspondence requires that different representations be created to represent different cognitions, and that similar representations be created for similar cognitions. Thus it appears that adherence to the Principle of Cognitive Correspondence entails adherence to some version of the doctrine of canonical form. The doctrine of canonical form is usually taken to mean that the representation of identical ideas should always be identical. Indeed, some of the arguments offered above are similar to those used by Schank (1975) to motivate his representation system.

However, the doctrine of canonical form has not been widely accepted. If this doctrine is incorrect, then the more general Principle of Cognitive Correspondence is in jeopardy.

2.1.1. Arguments Against Canonical Form Revisited

I believe the lack of acceptance of this doctrine is due largely to a misunderstanding. The misunderstanding is a confusion about the relation of canonical form to what is often termed "decomposition into primitives." Once this distinction is made, the rejection of canonical form would appear to be indefensible. Moreover, its acceptance would be compatible with representational systems not subject to the decomposition restricted, which, as I shall argue below, is unwarranted.

To defend canonical form, it is necessary to review the particular objections voiced against the doctrine. Probably the most direct and forceful are those of Woods (1975). Thus I shall be concerned here largely with his objections.

Woods makes three arguments against the existence and utility of canonical forms. First, Woods claims that, even if there is a canonical form for English sentences, it may very well be uncomputable. That is, there may be no effective procedure for determining if two sentences should have the same form. The reason for this deficiency, according to Woods, is that there are certain mathematical structures for which it is known that there exists no computable function that produces a canonical form for a given expression. Rather, one must search for a chain of equivalence transformations for each pair of (possibly) equivalent expressions. If this is the case for English sentences, there can be no canonical meaning representation, as no canonical form can be computed from an individual sentence.

Next, Woods claims that the computational advantage of canonical form is illusory because one rarely needs to determine if two things are paraphrases. Rather, one is most often interested in implication in one direction, i. e., whether the one expression logically entails another, not whether they are the same. Since canonical form does not eliminate the need for inference in these cases, but only when one has full logical equivalence, the actual computational complexity of one's system is not diminished.

The final argument is that complex concepts, like "uncle", for example, need to be stored directly. The problem is that there is a kind of ambiguity in concepts like "uncle", since one can be an uncle by being either the brother of a mother or a brother of a father. Since there is no way determine from a particular assertion of unclehood which of these is the case, there is no single form to reduce this assertion to. Since there is no single form, one would be compelled to store "uncle" as a concept in the system. We could then make assertions that particular individuals are uncles. However, we might also have some assertions that some individuals are brothers of fathers, say, without the explicit assertion that these are indeed uncles. But then we would have some uncles represented one way, and others represented other ways, and our representation would not be canonical.

Let us now example these arguments in reverse order. The evidence given by Woods that there is no way to avoid having a predicate "uncle" in one's system is that "Lindsay had no good solution to this problem". Therefore "It seems that for handling 'vague' predicates such as uncle ... we must make some provision for storing such predicates directly." Of course, it is all to easy merely to dismiss this argument simply because it is, in effect, not an argument at all. A particular attempt by a particular researcher to solve a problem does not mean that there is no solution. In fact, there are a number of rather obvious solutions to this particular problem that not only preserve canonical form but also allow decomposition into primitives. For example, we could assume that "parent", rather than "mother" or "father", is a basic term. In this case, the definition of "uncle" (which, contrary to Woods claim, is not "vague" at all, but merely contains a quite precise disjunction) is easily expressible without recourse to an "uncle" predicate (i. e., "uncle" is simply "brother of parent") Alternatively, one could allow "or" to be a permissable part of one's representational vocabulary, and represent "John's uncle" as something akin to "the brother of John's mother or father". In either case, no "uncle" node is strictly necessary; the resulting system would have both canonical form as well as be decompositional.

Of course, this rather trivial falsification of Woods' argument is beside the point. The reason is that *there is no tension at all between the doctrine of canonical form and systems which have nodes corresponding to higher-level concepts.* Indeed, I shall argue below that both elements are necessary.

In particular, let us confront Woods' assertion that having terms "uncle" as well as terms like "brother", "sister" and "father" in one's representational vocabulary leads to non-canonical representations. Consider two representations, one of which is something like

(4) UNCLE(Bill,John)

and another which similar to

(5) BROTHER(Bill,Al) and FATHER(Al,John)

According to Woods, these both denote the fact that Bill is the uncle of John. However, we now must have two *representations* that mean the same thing; hence the representation is not canonical.

This assertion is simply false. It is true in both cases that that Bill is the uncle of John, but it does not follow that both representations *mean* the same thing. For example, the assertion that Bill is the uncle of John is consistent with the possibility that Bill has no brother, while the assertion that Bill is the brother of John's father is not. The two expressions are not truth-conditionally equivalent, which is probably the most accepted necessary condition for an equivalence of meaning.

But certainly, representations (4) and (5) share some meaning in common. And the doctrine of canonical form should require this commonality to be represented identically. So don't the two representations violate the doctrine in this respect?

Quite the contrary. For our representation to be correct, the "uncle" relation must be connected to concepts such as "brother", "father", and "mother" in a certain precise way (we will suggest an actual representation below). And it is precisely these same concepts and relations that would be involved in a representation of an instance of a "brother of father" or "brother of mother" concept. Thus, in a well-formed representation, our description of what it means to be an uncle overlaps with a description of what it means to be the brother of a father, say, precisely to the extent that these share a common meaning. But of course, this is exactly what conformance to

the doctrine of canonical form requires.

The problem is that Woods confuses canonical form with decomposition. But canonicalization can be achieved merely by having *overlap* in the representation of items; we need make no further assumptions about decomposition into primitives or the like. For example, the doctrine of canonical form would be adhered to if concepts such as "punching" and "slapping" were each represented with separate nodes, each node pointing to the node for "hitting" via an ISA-link, or the equivalent. That is, what was in common between them is in fact represented by the same element. In recent connectionist proposals, canonicality may be achieved by having units representing different ideas connected to other units representing a common component (Feldman and Ballard 1982). Alternatively, in a "pattern of activation model" (Hinton 1981), canonicality may be achieved by different representations sharing a common subpattern, where the subpattern represents a common component. Thus, the canonical form position is seen as being compatible with quite different representational schemes.

Popping to the previous argument, Woods claims that since most inferences are of "one-way implication" rather than full equivalence, there would be no computational advantage of a canonical representation. That is, we would want to know whether "punching" entails "hitting", not whether they are identical. Since this requires finding an "inference chain" between two things that are not equivalent, having a canonical representation does not help. Moreover, the "full equivalence" case is just a special case of this problem, so it "falls out" as a consequence of implementing such an inference mechanism, which we are obliged to do anyway.

Once again, the argument is simply false. A canonical representation directly facilitates exactly the sort of one-way inference Woods states is most common. Here's how: By our definition, in a canonical system, if two items share some meaning, their representations share a common component. In the special case of one meaning *properly including* the other then one meaning representation properly contains the other as a subpart. Hence one-way implication is reduced to the single simple operation of determining whether one representation contains another. All the "chaining" Woods refers to is thereby eliminated.

In a hierarchical system with canonical form, this process is can be made extremely efficient. In a properly constructed representation, common subparts are represented by the same nodes. Therefore, the one-way implication is done simply by determining if the node representing one assertion appears in the representation the other. For example, a proper (i. e., canonical) representation of "punching" must include a reference to the concept "hitting". To determine whether "hitting" is implied by "punching", we merely look to see if the node for "hitting" is contained in the representation of "punching". But this is trivial.

We are left with the first argument, namely, that there may be no effective procedure to produce canonical forms. Once again, we are tempted to dismiss the argument out of hand, when we consider what is offered as its support. Specifically, the evidence cited for this is the lack of such an effective procedure for certain mathematical formalisms. The problem, of course, is that this is merely suggestive. It says nothing at all one way or the other about whether such a procedure exists for natural language. Indeed, the actual efforts to produce such procedures have not met with computational difficulties.

However, we might take this opportunity to make a stronger point. Even if the lack of an effective procedure could be demonstrated, this would not matter. This is because of a rather widespread misunderstanding that I call the *effective procedure fallacy*. The fallacy is that there does not appear to be an *effective* procedure for any cognitive process. That is, cognitive processes appear to work well enough, but not perfectly. In this particular (and typical) example, all that it necessary is a procedure that works to produce a canonical form virtually all the time. The lack of existence of an effective procedure has no bearing whatsoever on the existence of such

a virtually effective procedure.

As an analogy, suppose I had a computational theory of human vision. However, suppose we were able to show that my theory postulated the existence of some process for which there is no effectively computable procedure. In response, I suggested a heuristic method. In addition, because it is heuristic, my theory will make mistakes in certain situations. Suppose further that these circumstances turn out to be exactly those situations in which the human visual system is subject to the optical illusions. Following Woods, you reject my theory because it is computational intractable; you would prefer a theory that was not subject to such drawbacks.

The fallacy is the belief that the lack of mathematical perfection of a proposal is grounds to reject that proposal, when the same lack of mathematical perfection exists in the underlying phenomenon one is attempting to model. Thus, we are rejecting the theory because it corresponds too well to the data, but not well enough to how we wished the world would be. It certainly seems plausible that natural language understanding is not entirely algorithmic, i. e., that it involves processes that break down in some situations, that sometimes produce an incorrect parse, etc. Now, one cannot say whether these phenomena will be explained by a heuristic rather than effective procedure. But the point is, their existence cries out for such procedures to be embraced as a scientifically plausible explanation, not to rejected them because then do not suit our mathematical sense of aesthetics.

There is yet another fallacy herein, one that is in fact rather widespread in some quarters of the cognitive science community. I call this the *fallacy of the long run*. Effective procedure arguments, and their kin, complexity arguments, are valid only if we make certain totally unreasonable assumptions. Two such typical assumptions are that the size of the inputs to our system are in principle of arbitrary length, and that the kinds of dependencies that can exist locally can exist at an arbitraty distance. For example, in the case of language, we need to assume that we will be working with arbitrarily long sentences. If we can assume some actual bound to the size of our input, complexity classes almost always collapse into the simpest of all possible cases.

This is true in the effective procedure argument above, for example. If we assume that the length of an input is bounded, it is easy to find the effective procedures in question. Let us assume, then, that no sentence will be longer than a trillion words. Then the existence of an effective procedure is guaranteed.

Thus, effective procedure and complexity arguments are only true in the long run. But there never is a long run. As John Maynard Keynes so aptly put it, in the long run, we are all dead.

2.1.2. Cognition and Meaning

There is a more significant point to be salvaged from the wreckage. This is that the acceptance of Cognitive Correspondence entails the acceptance of *cognition as a pertinent component of meaning*. As an example, suppose one makes the following statement:

(6) John is the brother of either Bill's father or mother.

It seems a reasonable to reply to this statement by saying

(7) Oh, you mean he's his uncle?

That is, it appears as if the recognition of one fact as an instance of the other constitutes an *additional* understanding. By the Principle of Cognitive Correspondence, this should require a

representational difference. In fact, it would correspond to the creation of a instance of the "uncle" relation, where no such instance had existed previously.

Thus, representations are canonical with respect to how things are cognized, not just according to what is true in some objective sense. We will not belabor this issue here, other than to say that we must justify a representation by how well it reflects a conceptual system as much as how well it reflects truth. We will return to this point in the discussion of "non-factual representations" below, and again in the section on *views*.

3. Critique

In this section, I describe a number of problems with existing knowledge representation systems. The critique is divided into three parts: a critique of Conceptual Dependency, a critique of frames and semantic networks, and a critique of predicate calculus. Each of these approaches embodies quite different ideas about what knowledge representation is, and has quite different advantages and problems. It is not possible to do justice to any of these systems here, so let it suffice to say that each of them has elements that are essential for any good representation. My aim is to view them in light of representational principles.

3.1. The Problem with Conceptual Dependency

Conceptual Dependency (Schank 1975) proposes a taxonomy of conceptual objects that consists of actions, states, state changes, causals, conceptual nominals (i. e., objects), time descriptors, and a few modifiers. Most conceptual objects have a fixed number of "cases" (i. e., slots that can be filled), which, when appropriately filled, form an individual conceptualization. The theory postulates that all complex conceptualizations are composed out of combinations of a small set of primitive concepts. These concepts are primitive in that they are not further decomposable into the other concepts; rather, their semantics is determined by how they are related to other elements of the systems (e. g., by inference procedures).

As an example, the primitive act **ATRANS** denotes an abstract transfer. It takes cases for an ACTOR (i. e., action initiator), an OBJECT (the thing transferred), and a RECIPIENT and a DONOR. Thus "John gave a ball to Mary" is represented as an **ATRANS** with John being the ACTOR and DONOR, the ball the OBJECT, and Mary the RECIPIENT. This is usually rendered graphically as follows:

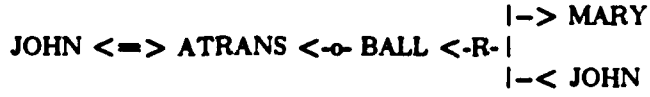


Figure 1

"John bought a ball from Mary" is represented by two **ATRANS** standing in a relation of "mutual causation" to one another: One **ATRANS** represents John giving some money to Mary; the other, Mary giving the ball to John:

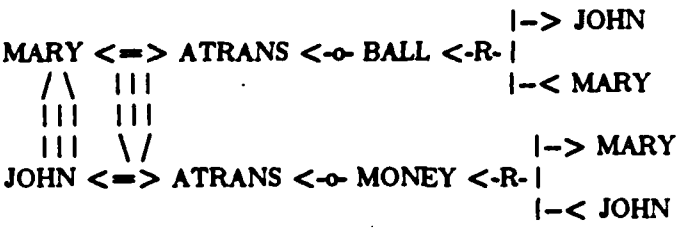


Figure 2

(Of course, the diagrams are simplified; actual representations contain additional information, such as the time of the action, etc.).

As these examples suggest, Conceptual Dependency (hereafter CD) has been concerned largely with representations for actions. Important goals of this formalism are that it be a coherent theory of meaning, that it is psychologically plausibility, and that it facilitates efficient inference. According to CD, being a coherent theory of meaning entails, among other things, that the representation has canonical form, i. e., that things meaning the same thing are represented the same way. For example, if we assume for the sake of this discussion that "Mary received the ball from John" has the same meaning as "John gave Mary a ball", then these sentences would have the same CD representation, namely, the representation shown in Figure 1.

Psychological plausibility means that the representation should help account for certain psychological phenomena. For example, CD helps explain why people have difficulty recalling the exact form of an utterance while they more readily recall its content. The explanation is that representing meaning separately from the actual words makes it possible to recall them with different efficacy. In particular, those representations that represent the content may be stored for ready recall, while those denoting the text itself may be relegated to a lesser status. Recognition phenomenon are accounted for similarly - a sentence with a different surface form but the same meaning as a previous sentence would nevertheless match the meaning representation of that sentence, thus causing the subject to "false alarm" to the stimulus.

In addition, the CD form of representation is claimed to be computationally efficient: No inference needs to be done to determine whether two utterances have the identical meaning; the representations of their meanings will be identical if and only if this is the case. A stronger claim is that inference in general is greatly aided by this representation. This claim is based on the fact that CD embodies what is sometimes referred to as "decomposition into primitives": All complicated things are represented by an assembly of primitive elements. There is no need for special inference elements for each word of a language, or each underlying idea. Instead, we have only the inference routines associated with the (small number of) meaning primitives. For example, instead of a separate "buy" notion to deal with, CD represents, and hence reasons about, buying with a structure containing only **ATRANS**s, causals, and the like. Thus, to infer from "John bought a ball" that John ends up having the ball, we need only apply the inference routine for **ATRANS** to a piece of the underlying representation. We do this without recourse to any special information about "buy".

Similarly, since all other words of a language that involve abstract transferring are mapped into representations involving **ATRANS**, no special knowledge or inference routines is need for these words to produce the correct inferences insofar as transferring is concerned. To the extent that words of a language can be decomposed into primitives, to that extent individual inference routines may be eschewed in favor of a small number of inference routines associated with the various primitive elements. Rather than a vast number of inference routines, a mere handful will do.

Note that most of the claims made for CD involve principle of representation language use, rather than principles of the language per se. For example, canonical form is a principle of language use,

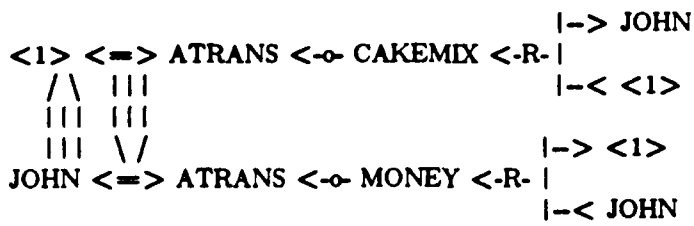
since it determines what it said in the language. Thus, the CD vocabulary for primitives is presumably just one such vocabulary that meets these goals. Other vocabularies may be proposed if they adhere to this (and other) principles.

We will save the evaluation of these principles, and of CD's adherence to them, for the next section, where principles of use are discussed. For the time being, we shall look at some more fundamental issues.

3.1.1. Critique

The main proble with Conceptual Dependency, as I see it, is that it lacks "higher-level" objects. By this I mean that, even if the meaning of a complex notion can be represented in terms of a simpler ones, the need to have a higher-level entity persists. However, Conceptual Dependency does not accommodate this need. Therefore, it is epistemologically inadequate in this respect.

This problem can be illustrated in a number of ways. For example, consider the claim that the CD reduces the inference problem by allowing inferences to be organized around conceptual primitives. An example due to Rieger (1975) involves "specification inference", that is, the process of filling in an unfilled case slot through a primitive-related inference routine. Consider the sentence "John bought a cake mix." As described above, the CD analysis of "buy" is two **ATRANS** conceptualizations in a "mutual causation" relation to one another:



Note that in this example, we do not know the fillers of several of the slots, although we do know that they are all the same entity. I have designated with the notation **<1>**. Among other things, this entity plays the role of the party which sold John the cake mix.

One desirable type of inference is the determination of the filler of this slot. Rieger's solution to this problem, and to the problem of inference in general, is to have a collection of different inference routines for each recognized type of inference and for each primitive act. For example, this kind of inference is called *specification* inference, because it involves filling an unspecified slot. The inference would be attempted by a specialized **ATRANS** specification inference routine. This routine generally supplies an appropriate default. In this case, Rieger's inference routine fills the empty slot with the representation of a grocery store.

The problem here is that the inference cannot properly be made by an **ATRANS** inference routine. There is in fact nothing about the idea of transferring some abstract quality that rightly suggests who might have transferred a cake mix to someone else. For example, someone handing John a cake mix would also be represented by an ATRANS with an unspecified donor, yet the inference that the cake mix came from a grocery story would be inappropriate. Rather, the inference should be entertained only when the cake mix is exchanged for some money. In other words, what we have here is not an inference about ATRANS - it is an inference about buying.*

Making an inference about buying is problematic in a system based on the kind of decomposition

*Actually, it is an inference about "prepackaged food buying", or some such. The fact that we need even more specific categories only exacerbates the problem.

into primitives advocated in CD. What Rieger's program had to do to be able to make such an inference is check to see if a particular ATRANS it was examining happened to be in a mutual causation relation to another ATRANS that happened to involve the transfer of money. In effect, the program had to check to see if it was dealing with a case of buying. In a sense, the program had to reconstruct the existence of buying, which is explicit in the original sentence but only implicit in the CD representation.

Now, in a sense, this problem is well-known to Conceptual Dependency advocates. Their response is that most inference requires the application of large quantities of world knowledge. The problem is in Rieger's notion of inference. This needs to be replaced by something closer to script-application, and other knowledge-intensive inference processes.

This response is correct, I feel. However, the representational consequences of it have yet to be felt. If organization of inference is a primary consideration in justifying representation, and if inference is largely the product of the manipulation of essentially arbitrary pieces of world knowledge, then there is nothing to single out particular pieces of representation: We will need arbitrary elements in our representation, and CD does not have them.

The point of all this is that decomposing conceptual objects into primitives doesn't help one make inferences any more than it gets in the way. It facilitates inferences about more abstract ideas, for example, change of possession, only at the cost of making it more difficult to make inferences about more complex ideas, such as buying. What is more significant, though, is that the idea of getting rid of non-primitive objects simply fails. In the example above, knowledge about the idea of buying sneaks in through the back door. It has to, because such knowledge is necessary to the task. It makes no difference whether the theorist decides to acknowledge the existence of this concept in the theory - the *system* acknowledges it. Not according full citizenship to such conceptual objects does not eliminate them; if merely makes it more difficult to organize inferences about them.

Note that this argument does not depend on any weakness of the particular set of primitives employed or the details of the particular illustration. I am not raising the objection others have make of CD, namely, that it is an ad hoc collection of particular representations. Quite the contrary, I have assumed the existence of a complete and correct set of elements conforming to the CD spirit of things.

The situation is not ameliorated by refining the system because most concepts have gestalt properties, in the sense of Lakoff (1977). I do not mean anything mysterious or anti-computational by this terminology. Rather, I use the term to convey what I consider to be the rather straightforward observation that concepts have properties not readily deducible from their components. Thus, even if we believe that we have a correct decomposition of a concept like "buying" into its parts, we cannot determine from these parts that the buying of cake mixes goes on largely in grocery stores.

To put it another way, having a useful rendering of a conceptual entity in terms of components does not eliminate the need for having that conceptual entity in one's system. But decomposition into primitives a la CD denies this need. Moreover, the inferential advantages attributed to CD vanish with this realization. It is no longer sensible to talk about organizing inferences around a small number of objects. Instead, all important inferencing becomes the manipulation of a large knowledge base of facts not derivable from a fixed set of components.

It is important to note at this stage that I have not challenged most of the tenets of CD in making this critique. In particular, I have not questioned that one reasons about concepts rather than words; that ideas may consist of combinations of simpler ideas; that canonical form is an essential requirement of any meaning representation, or even the existence of conceptual primitives. I

merely have challenged the idea that it is feasible to have a meaning representation that does not include within it a large and in principle unbound set of high-level objects.

My critique may seem a quibble, then. However, I believe that other errors follow from this one, and that a quite different representation is necessitated by it. As evidence for this, I cite the existence of what I call "unmotivated levels" in CD. To introduce this notion, consider how a complex idea like "threaten" is represented in CD. There is no specific CD element corresponding to "threaten"; rather, English sentences that refer to events involving threatening are mapped into a complex of primitive acts, similarly to the way "buy" was mapped into a complex involving **ATRANS**s shown above. Thus, "threaten" per se is not an object in CD, where all objects are primitive elements.

However, analogous to the "buy" example above, there are important inferences that require the existence of information that is quite specific to "threaten". For example, it is necessary to know that threatening is a way of getting somebody to do something they might not do otherwise, that it has certain prerequisites to its being carried out successfully, that it may have drastic consequences, that certain kinds of threats are illegal, and so on. These facts are specific to "threaten" in that they are not computable from its underlying primitive components.

Thus, to make the inferences requiring these facts, "threaten" is introduced back into CD-based systems. However, it is not introduced as an action. Instead, it is given the status of a *plan*. Plans exist on what is called the "knowledge structure level" of representation, and are outside of the domain of CD proper (Schank and Abelson 1977). This is in contrast to elements like "tell", for example: This word is generally mapped into a pure CD representation involving the primitive MTRANS (denoting "mental transfer"); no knowledge structure equivalent is postulated.

Observe first that an entity that was banished on the CD level reappears as an entity on another level. This is as it should be, since, as I have suggested, the explicit acknowledgement of these concepts is necessary eventually. However, there appears to be no good reason to propose two separate *levels* of representation. This must be done here only because CD is unable to digest such large concepts. In fact, this inability has apparently led to a serious error: The difference between "threaten" and simpler actions, e. g., "tell" is not that one is a plan and the other an action. "John threatened to kill Bill" describes an action just as assuredly as "John told Bill he went home." In addition, intending to tell someone something may be a plan to have that person know something, just as assuredly as threatening someone may be a plan for a less benign goal.

In sum, it makes no sense to classify "threaten" as something other than an action. It differs from CD actions only in its complexity, not in its epistemological status. Making this distinction violates the principle of uniformity, since it introduces distinctions where they are neither necessary nor desirable.

The evidence for the lack of motivation of levels increases as additional levels are considered. For example, *scripts* (Schank and Abelson 1977) are a variety of complex events that, obstensibly, dwell on the knowledge structure level (i. e., they are not part of CD proper). However, if one looks at how references to scripts are made in actual systems (e. g., Cullingford 1978), one usually sees "John ate at a restaurant" represented in a manner quite like a CD form: Instead of a primitive act, one sees a reference to the restaurant script; a CUSTOMER case is filled with the particular customer (in this case, John), and a RESTAURANT case with a representation for some restaurant.

The problem raised here is twofold: First, the actual representation appears not to be formally different from a CD representation. So the claim that somehow a different "level" is involved seems not to be borne out functionally. More significantly, once one allows the existence of such complex objects anywhere in the system, there seems to be no way to prevent an avalanche. If

we allow a "do the restaurant thing" object into the system, with associated CUSTOMER and RESTAURANT cases, to be used to represent a particular instance of eating at a restaurant, then why not also allow in a "buy" object, with associated BUYER, SELLER and OBJECT cases, to be used when representing an individual buying event? As I have been suggesting, there is no reason not to do so, other than the otherwise unmotivated claim that these are on different levels.

It is possible to cite many other such cases. However, they each point to the same conclusion. CD tries to do without many conceptual categories, only to find them reappearing in some other part of the system. By not acknowledging this need, moreover, unmotivated non-uniformity is created where a uniform theory is possible and desirable.

One may wonder, then, why there is such apparent resistance to letting entities proliferate in CD, if similar entities are being proposed anyway. I believe there are two answers. One is that the resulting system would hardly resemble CD in spirit anymore. Rather than a small, fixed set of priviledged objects, we would have a large, open-ended set of undistinguished ones. The system would also appear to be less language-independent, for while a small set of primitives may not be language specific, this is less likely to be true for all complex concepts. The system would not even support decomposition into primitives in the CD sense, because most of the time we would refer to high-level objects rather than their decompositions.

But perhaps a more distressing problem for CD aficionados is that this proliferation of entities appears also to violate canonical form. For example, if we let in "buy", we would appear to also have to let in "sell"; we would have to have a "give" concept as well as a "receive" one. The commonality meaning of the underlying ideas appears to have been lost, and this is perhaps the most important goal of the representation.

I believe there is no way out of the first bind. There is simply a large number of conceptual objects at arbitrary levels, and they all must be accommodated. However, the second, and perhaps more serious problem, I believe must and can be solved in a satisfactory way. The representational system described below is one attempt at such a solution.

In all fairness, it should be emphasized that the advocates of the CD representational system have long since extended their repertoire to include many other objects not of the original CD ilk (e. g., Schank and Abelson 1977, Schank 1982). In addition, I believe that CD theorists have been much more concerned about the content of the knowledge they represent than postulating (or even acknowledging) a general structure for it all. I am enamored with much of this analysis and seek to preserve as much of it as possible. Nevertheless, I believe that my criticisms are still accurate and applicable. Moreover, the proliferation of conceptual entities currently being postulated by CD advocates would seem to strengthen the need for a unifying strategy such as the one I propose below.

3.2. The Problem with Frames and Semantic Networks

Semantic networks and frame-based languages are probably the most popular candidates for a knowledge representation scheme. Some of the problems with the former have been suggested by Brachman (1979); these arguments generally consist of pointing to ambiguities and omissions in most semantic net formalisms. My arguments are mostly in the same spirit as these. However, I aim my arguments at frames rather than semantic networks. The primary reason for this is that advocates of frame-based systems appear to have been less sensitive to these arguments than have semantic-network theorists. Moreover, if we add defaults and procedural attachment to semantic networks (as has been done in most recent systems), it is impossible to differentiate semantic nets from frame-based systems. Thus the criticisms offered below are applicable to semantic networks, although they are stated in terms of frames.

The equivalence of frames and semantic networks may need a bit of clarification, as they are not generally acknowledged to be so. As a theory, frames are large chunks of information, with an emphasis on default-oriented reasoning (Minsky 1974). As such, they are similar to the schemata of Bartlett (1932) and Rumelhart (1975), the scripts of Schank and Abelson (1977), the scenes of Fillmore (1977) and the cognitive models of Lakoff (1982). Certainly, these theories of knowledge structuring serve an important role in current models of cognition. I have no bone to pick with them as theories.

However, practice is another story. Several quite interesting attempts have been made to use the theory of frames as a basis for a knowledge representation language. In all of these, most notably FRL (Roberts and Goldstein 1977) and KRL (Bobrow and Winograd 1977), a frame is implemented as some sort of object supporting a number of labelled slots. The frames themselves are generally arranged in a hierarchy. Most frame languages have some built-in mechanisms for frame and slot manipulation. The most basic of these mechanisms (a) allow slot values to be inherited down a hierarchy; (b) constrain the filler of a slot to be in accordance with some user-supplied specification; and (c) allow the user to attach to slots procedures that are invoked in a variety of ways.

For example, it is typical in frame systems to have a "Person" frame. Such a frame is apt to come with slots bearing the names "Address", "Name", "Age", and so on. In addition, each slot may be constrained to tolerate a certain kind of filler. For example, "Age" may be constrained to be filled by a number between 0 and 120, "Name" by a character string, and so on. To represent a particular person, the frame system user creates a particular instance of the "Person" frame, and fills in those slots for which information is currently available. Thus, if I want to represent a person named John Smith who lives at 123 Main Street, I could create a new element, say "Person1", which I attach under "Person" in my frame hierarchy. In addition, I fill the "Name" slot of this element with the value "John Smith", and the "Address' slot with the value "123 Main Street".

It is likely that the "Person" frame is itself a point on a hierarchy. For example, "Person" is likely to be represented as a kind of "Animate-Being", or some such. In addition, various kinds of "Person" probably are represented, such "Man", "Woman", and "Child".

In such a realization of frames, there is a natural, one-to-one mapping of frames onto semantic networks. Namely, the frame itself could be interpreted as a node, and the slots of the frame as links. Since, in most frame systems, the fillers of slots are also frame objects, the links would point to other nodes under the interpretation given here.

The slots of more abstract frames, which perhaps have no fillers but only constraints, can be interpreted as the specification of which links a particular semantic network node might manifest in a given instance. In some semantic network systems, such general assertions are not expressed in the semantic network language per se; so in these cases frame languages may be said to have somewhat greater expressive power. However, the distinction is not intrinsic. For example, KL-ONE (Brachman and Schmolze, 1985*) is a semantic network based system in which these more abstract assertions can be made.*

In the other direction, we can interpret a link in a network as a slot in a frame. In particular, most semantic network systems support some sort of ISA hierarchy with inheritance (in fact, the idea originated in that context in Quillian 1968). These are directly interpretable as statements

*The fact that frame and semantic networks are notational variants is interesting when one considers that the two schools of thought appear to have little concern for one another. For example, the article by Brachman and Schmolze mades no reference whatsoever to the nearly identical notations proposed by frame-oriented researchers.

about hierarchical status in a frame language.

So the primary distinction between frame languages and semantic network languages is procedural attachment, that is, the ability to automatically cause a procedure to be executed upon certain events occurring. However, this feature not properly a representational feature, but, rather, a programming device. (To the extent that one would want to consider procedural attachment a representation, it is argued below, to that extent the representational force of the language is compromised). Putting it another way, one could add a "demon"-like facility to a semantic network system that does essentially the same thing as the procedural attachment facility of a frame system. Moreover, one could do so without compromising the fact that one's representation language is a semantic network. Rather, one would have merely implemented some part of one's algorithm or interpreter in a data-driven fashion. Indeed, such a facility is provided in KL-ONE.

To summarize, to say that something is a semantic network system rather than a frame language is more a function of one's background than a fact about one's program. In addition, while the theory of frames seems to conform to some compelling intuitions about cognition, it is unclear that frame *languages* capture these intuitions, or at least, capture them in a way that is importantly different from the way other formalisms, in particular, semantic networks, might capture them.

3.2.1. Critique:

Problem 1: The Meaning of a Slot is Completely Unconstrained

Despite the apparent usefulness of frames, what it means to be a slot in a frame is rather ill-defined. The meaning of a slot generally appears only procedurally, if at all. For example, consider the "Person" frame alluded to above. The problem with this example is that there is no reason to believe that the "Address" slot filler represents this individual's address and the "Name" his name, and not, say, the other way around. Of course, we human knowledge-hackers immediately appreciate the difference. But what guarantees that the system will?

The usual answer to this sort of question is that the various programs using this information are designed to manipulate these slots and values in a manner consistent with our intuitive understanding of them. For example, a natural language system processing the query "What is John Smith's address?" will know to look in exactly the right places to retrieve the correct answer.

This may in fact be the case, but it doesn't solve the problem. It is merely an admission that the meaning of slots like "Name" and "Address" are encoded procedurally. That is, they are encoded by the way in which routines manipulate things, rather than in an explicit and declarative representation language. However, this places the representation of their meaning outside of the frame system itself. We are now relying on a piece of code to establish the correct interpretation of any symbol in our system. Moreover, for symbols like "Name" and "Address', the amount of code needed appears arbitrary and open-ended.

It is important to emphasize just how arbitrary the relationship between slot and frame may be. For example, the "Name" slot is built to hold the name of an individual "Person"; but the "Address" slot holds the address of the *residence* of that "Person". Among other things, the latter slot posits the existence of an additional object, namely, the person's residence, while the former slot posits no such thing. All this merely illustrates that slots are truly meaningless symbols.

In sum, the frame system itself says nothing about the meaning of the slots in its frames. It

cannot be said to be representing knowledge, as the true knowledge is encoded procedurally in the various and arbitrary procedures that are unstructured and external to the representation system.

Problem 2: What May be a Slot in a Frame is Completely Unconstrained

A frame system advocate is likely to respond that much useful information is encoded in his formalism, but in different ways. For example, frame systems allow the user to specify the slots associated with each element; this structures the knowledge and, thereby, the routines associated with it, in an important way. And, after all, structuring knowledge is the name of the game.

The problem with this argument is that there appears to be no "in principle" answer to the question of which frames can support which slots. For example, as noted above, it is customary to allow "Name", "Address" and "Age" to be slots in "Person". Without stretching credulity, it one could imagine slots for "Father" and "Mother" as well. Presumably, the criterion that is used here is some intuition that each person has a name, and each person has a mother; therefore these should be slots in the "Person" frame.

While we're at it, we can add slots for brothers and sisters as well. This may require some extension of the frame language to handle lists of fillers, as a person may have more than one of each. Let us give frame languages the benefit of the doubt here, because without such an extension they could not represent these notions at all.

But where does it all end? For example, each person's mother has a maiden name; therefore, we can safely conclude, each person has a mother's maiden name. Likewise for father's first name. By the same reasoning used above, we should postulate a "Mother's-Maiden-Name" slot and a "Father's-First-Name" slot. Similarly, we can posit an slot for "Accountant" and "Tax-advisor" (at least among "Computer-Science-Professor"s). We can also add "Best-Friend-in-High-School" or "Favorite-Movie-Starring-Robert-DeNiro" or "Doctor-Who-Did-Delivery" or in fact, any other category we feet like making up at the moment.

If this argument is correct, then the "frames supply structure" defense of frames is clearly undermined: There would be no particular set of slots belonging to a particular frame, as virtually anything could end up a slot on anything else. Therefore, there could be no set of slots to provide the much touted structuring.

Now there are at least two partial responses to this seemingly unlimited proliferation of frame slots:

(1) The first counter-argument goes like this: The first name of one's father can be computed from other knowledge (e. g., the fact that a person has a father who is also a person, and that every person has a name) and therefore doesn't require a slot of its own. Moreover, "Father's-First-Name" clearly "belongs" to the representation of one's father, and not to the representation of "Person" per se. Thus we can simply eliminate such slots; this will help keep down the number of seemingly arbitrary slots.

(2) The second argument is that the presence of a slot is, among other things, an efficiency consideration; we don't normally have all these funny slots because we normally don't need them.

Both counter-arguments are false, though. Consider the first counter-argument. As was suggested above, concepts like "Address" seem to belong to one's residence, rather than "Person" per se. Indeed, we could compute a person's address from knowing that person's residence, and from

knowing the address of that residence. Therefore, if we are to eliminate "Father's-First-Name" and the like because of counter-argument (1), then, by the same token, we would have to eliminate slots like "Address" as well. In either case, our frame representation is erroneous.

Now, frame aficionados might be willing to eliminate the particular slot "Address" from the particular frame "Person" in order to salvage frames in general. However, we still have no way to tell which frame a particular slot "belongs" to. As I demonstrate shortly, the whole concept of "belonging" to a frame is not entirely cogent. However, even without that stronger argument, the criterion of counter-argument (1) will probably overturn a considerable fraction of the structure of actual frame systems. Even worse, we still have no way to eliminate the uncountable number of slots like "Favorite-Movie-Starring-Robert-DeNiro", as there does not appear to be an intervening object already associated with "Person" that should properly carry this slot.

The second argument is simply an admission of guilt. Namely, it defends the particular slot choices by appealing to efficiency considerations. But this is clearly not a representational issue. It merely amounts to saying that frames do not have representational status. That is, we get to design frames in accordance with what we feel like doing, not in accordance with what things mean. The theoretical issue of how to represent knowledge is circumvented.

Most frame advocates seem to share the intuition that complex elements like "Best-Friend-in-High-School", etc., just aren't meant to be frame slots. In actual practice, frame systems users appear to represent such knowledge outside the frame system. For example, complex elements would be represented as a conjunction in a predicate calculus-like formalism. The problem with this is that now there are two systems of representation. We have no way of decide what would be represented in which, or what it would mean to represent it one way rather than the other.

Moreover, if we allow some items to be represented in another notation, then what is to stem the tide? If I represent "Best-Friend-in-High-School" as "Best-Friend(x,y)&Went-to-High-School-together(x,y)", or some such, they why not represent the fact that a person x is named y as Named(x,y)? We can do this with each less controversial slot name, and eliminate frames altogether.

Each alternative is bad: If we allow both notations, we have an unprincipled system, and one in which our frame language itself is limited and incomplete; if we try to be more uniform, we drive out the frame notation altogether. Thus frames must be either inadequate or unnecessary.

It should be noted that some frame advocates actually take this position. For example, Charniak (1981) claims that a frame is merely a convenient notation for entering predicate calculus formulae. By the previous critique, this would still leave us without a theory of representation per se.

The "Belonging" Fallacy

Most researchers seem to decide which slots to attach to which frames in accordance with something I call the "belonging" fallacy. The "belonging" fallacy is the idea that a given element should be awarded a particular slot in our representation because, in English, we would be given to say that the slot "belongs to" that element, or that elements of the type in question "have" other elements of the type that should fill the slot. Thus, persons *have* names, ages, and addresses; physical objects *have* weight and height; rooms *have* floors and ceilings, and so on. Alternatively, we can talk about the same notions using the possessive construct: We can specify John's age, the book's weight, the concert room's ceiling. One seems to belong to the other; hence the motivation for the representation.

This is a fallacy because these natural language constructs, and I claim, the underlying concept of

"belonging", has so many different interpretations that it is rendered virtually content free. That is, it is perfectly correct to specify "John's apartment", "John's car", "John's girlfriend" and "John's bodyguard". But these phrases have radically different interpretations: In first case, the phrase should probably be interpreted as the apartment John rents; the second as the car John owns; the third as women he dates, and the fourth as the person protecting him.

Clearly, these are radically different relationships. In fact, there appears to be no more to the meaning of these constructs than that there is some completely unspecified relationship between the two entities. This sort of natural language form is meaningful to us because we natural language processors supply the context and knowledge needed for proper interpretation. In fact, we may use such forms whenever the context or associations allow ready interpretation. Context may change the interpretation radically; e. g., if we are talking about real estate investments, "John's apartment" is likely to be one he rents to someone else; in a auto race, "John's car" may refer to the one is driving.

Thus, the claim that a concept "has" a slot is vacuous. It means only that there exists some relationship between two things. But as we have seen, the relationship may vary with context, may be contingent upon any number of deictic factors, and is arbitrary in content. The problem with frame/slot representations, then, is that they assert that there is a relationship between two entities. But the relationship is arbitrary and has no epistemological status.

Even some epistemologically sophisticated systems seem incorporate this fallacy. For example, in KL-ONE (Brachman and Schmolze 1985) slots (roles, actually) are explicit, structured objects, as is advocated here. However, the motivation for associating roles to concepts appears to be the "Belonging" fallacy. Thus "Thing"s *have* "Subpart"s and "Company"s *have* "President"s. As argued above, these are essentially meaningless assertions.

Problem 3: Many Concepts Do Not Get Defined

The gravamen of my critique of frames is that what we have been calling "slots" seem to be perfectly good concepts in their own right. These concepts are not only undefined - they tend to be completely unrecognized in frame systems. For example, the concept of "age" has a well-defined meaning (in fact, more so that does "person"). Namely, the "Age" slot implicitly refers to a concept which is the amount of time since the creation of an object until some other moment in time. Similarly, "Address" is a "referring object" for a location; "Name" is a "referring object" for a person, etc. It is nothing short of ironic that the concepts that more obviously have definitions are never represented in frame systems, while the ones that less obviously have definitions are dealt with explicitly.

As stated in the exposition of the first problem, the meaning of the slots in a frame system is represented only procedurally, if at all. It is not surprising, then, that concepts that roughly correspond to slot names in frame systems are not defined therein.

Some frame systems improve the situation somewhat, for example, by letting slots themselves have slots, or by creating frames that roughly parallel slot names. For example, it is possible in some systems to create a "Name" (or perhaps, "Person-name") frame, and then specify that the "Name" slot in "Person" can only be filled with a "Name" frame.

There is something right about this approach. In fact, I will advocate something similar to this below. The problem with this solutions is that in most frame systems, it leads to a duplication of symbols with unclear semantics. In the example just given, we have a "Name" slot in "Person" and a "Name" frame, and it is unclear how the two are related.

In sum, frame systems allow us a multipicity of objects, but seem to lose some basic criteria of a meaning representation in the process. In particular, they tend to divide up the world into frames and slots, the latter not having true concept status. But the latter do appear to be full-fledged concepts. Frame systems neither recognize this fact nor allow for the expression of the meaning of these items. Thus frame systems fall short on interpretability and uniformity.

3.3. The Problem with Predicate Calculus

Predicate calculus (PC) has as its advantage the uniformity required by a general representational scheme. It also is rather well-defined, i. e., it conforms to the Principle of Interpretability. It has also been applied successfully in a number of enterprises. However, PC has a number of important shortcomings as a solution to the knowledge representation problem.

Probably the most important shortfall is that PC is not really a representation system. This is because it does not make a commitment to the Principle of Cognitive Correspondence. If we do make such a commitment, PC seems to have certain difficulties.

For example, few individuals would want to claim psychological plausibility for PC. This is particularly true when one considers that PC includes a system of formal deduction, i. e., theorem proving. It is hard to imagine that one could entertain formal deduction as a serious theory of ordinary human reasoning, in light of the fact that humans are illogical, contradictory, and just plain bad at proving theorems.

One may still approve of the language of PC while ignoring its inference method. Even so, PC suffers from serious epistemological inadequacies. Only a tiny fragment of the notions found in ordinary human thought are treated explicitly in PC, namely, those natural language forms thought to be truth-conditional: the connectives "and", "or", and "implies", operator "not", and some quantifiers. For a theory of mathematics, this may be fine. But it stretches credulity to believe that such notions are at the basis of human thinking, and hence, appropriate for knowledge representation schemes.

In this vein, it is important to recall that PC usually is interpreted as first order predicate calculus (FOPC). To adequately express any natural language utterance, and especially those involving imbedded clauses, extensions are required. Logicians have tended to formulate each extension as a separate kind of logic. Thus, there are many extensions to PC, including temporal logic, logics of believe and necessity, and so on. Each of these logics introduces a few special operators, e. g., "believes", or "is necessary", plus some inference rules that integrate these operators into the general inferential scheme of things.

The trouble begins when we realize that each so-called propositional attitude requires its own separate logic. In addition to the logic of belief, for example, we also need a logic of knowing and a logic of obligation; moreover, we need a logic of desire and a logic of hoping; we need a logic of seeing and a logic of hearing; and we need a logic of saying.

One problem here is that it is hard to decide which logics we should actually have. For example, in addition to a logic of saying, should we also have a logic of telling? a logic of informing? a logic of remembering and forgetting? a logic of telephoning and telegraphing? a general logic of perception in addition to separate logics of seeing and hearing? a logic of smelling (to handle "The dog smelled the cat enter the room")? a logic of telepathy? a logic of liking and a logic of loving, hating and being indifferent to?

To have each of these logics, each propositional attitude becomes a term that appears explicitly in the rules of inference of the associated logic; that is, each is given a special status, compared to

the "ordinary" predicates that never appear in the rules of logical inference themselves. So "believes", "says" and "loves" join "all", "implies", and "and" as those terms recognized are part of the logic itself.

First of all, this to violate Cognitive Correspondence. No evidence suggests that people actually cognize things this way - that "tell" and "believe" are privileged predicates, compared, say, to "find", "give", or "digest". Moreover, we seem to have lost a certain intuitive appeal for the familiar family of logics by allowing in the distant cousins. Is it really meaningful to say that we have a logic of desire or a logic of saying? Such logics blend right in to our notion of meaning that we apply to ordinary predicates. That is, we cannot be said to be building a logic of "believing" anymore that we are building a logic of going to a restaurant or a logic of taking and giving. In all cases, we are merely describing the structure of some particular concept. But the distinctions of logic have become irrelevant.

As an example, note how the various "logics" alluded to above cry out to be organized into some hierarchy. That is, it seems reasonable that, if we are going to have a logic of seeing and one of hearing, then we should make them both "subclasses" of a logic of perception, with rules of inheritance, exception-handling mechanisms, and the like. But this move requires an analysis of these terms that is identical to the semantic analysis normally reserved for those predicates that are not considered to be properly part of a logic at all. That is, we have unmotivated the distinction that required certain terms to be recognized explicitly as part of the logic, and others to be dealt with as just ordinary predicates.

Of course, some of the logics alluded to above are better known and better worked out than others. Some seem less silly than the others. However, the motivation for working on one particular logic as opposed to another appears to be based on philosophical considerations of what constitutes an important and interesting problem. But this is merely another ad hoc assumption. What is more troubling, though, is the thought about what would happen if all these logics were developed. Logicians tend to develop each logic separately; in this way, rules can be developed that integrate each logic into the FOPC while still maintaining the formal properties logicians consider so desirable. But if the operators from *all* these logics are considered at once, the result is likely to be enormously chaotic. As pointed out by Israel and Brachman (1984) and others, the nice formal properties of the individual logical schemes become problematic as additional representational power is attempted.

Finally, we should recall that logic it at its best when truth is an issue. But it is not clear that most natural language statements can or should be evaluated with respect to truth as a way of dealing with their meaning. Fillmore (1985) suggests that a different sort of semantics is necessary for dealing with many utterances.

3.4. Non-factual Representations

One problem with all the representational ideas mentioned above is that they tend to represent only factual information. However, according to the Principle of Cognitive Correspondence, how one thinks about something is an important determination of its representation.

Consider for example the implications of the recent work by Fillmore (1982), Lakoff (1982) and Coleman and Kay (1981). In particular, their work on *frame semantics* and *idealized cognitive models* is directly relevant to issues in AI representation languages. The thrust of this work is that the meaning of words (and, as I will interpret it here, the structure of underlying concepts) cannot be adequately expressed in simple feature list models.

Fillmore and Kay use the classic decomposition of the word "bachelor" as a case in point.

Bachelor is supposed to mean "male of marriageable age who has not yet been married". (The long-winded definition is needed to exclude 2 month old boys and forty year-old divorcees.) However, this definition is quite unsatisfactory. As the authors point out, if strictly interpreted, the definition will include as bachelors the following categories or individuals:

(1) Members of long-term homosexual relationships

(2) Men living out of wedlock in stable relationships with women

(3) Tarzan

(4) Pope John Paul II

These examples demonstrate that the definition of "bachelor" offered above is problematic. For example, a robot informed only by this definition might try to fix up an eligible female acquaintance with one of the above. Obviously, this would be flawed behavior.

One way to remedy this situation is simply to add further specifications in the definition. It is not obvious that this can or cannot be done in this example. However, doing so will still leave us with the following uncomfortable question: If such "correct" specifications can be found, why is it so difficult to state some of them initially, in contrast, say, to those ready-accessible specifications like "male" and "unmarried"?

Fillmore and Kay offer a different solution. They propose that, instead of a simple feature list, the definitions of some words are made with respect to *background frames*. A background frame is some shared social schema. For example, in the case of "bachelor", the relevant background frame might be called "Traditional-Manhood-Path". According to this frame, boys reach a certain age, begin dating, and, if they meet the right girl, get engaged and then married. Another choice in this frame is to go on dating forever. It is against such a background frame that word meanings are stated. For example, "bachelor" could be defined as that option within "Traditional-Manhood-Path" in which the "dating" life-segment is never transcended.

(The term "frame" is not troublesome in this context because it is being used to refer to the theoretical notion of a large knowledge structure, not the representation language notion of slot-filler system. One could use a different term, as Lakoff does, without jeopardizing one's position. Thus the arguments directed at frame languages above have no direct bearing on this particular idea.)

Lakoff develops a similar idea, which he terms *idealized cognitive models (ICMs)*. The "Traditional-Manhood-Path" in the previous definition is probably a highly idealized reality. This can help explain the introspection of our definitions. Within this background frame (or ICM), being male and unmarried are necessary for distinguishing the "bachelor" concept from others within the framework. But we need not worry about how to exclude Pope John Paul II, etc., from this definition because no such alternatives are present in this ICM.

As Lakoff (1986) points out, these ideas are antithetical to much traditional linguistics and philosophy. For AI representation purposes, the implication is that a good representational system must reflect such a cogntive structure. A representation that is adequate for representing only factual information is not an adequate representation.

Another important consideration for a representational system is the role of metaphor (Lakoff and Johnson 1980) and metonymy (Lakoff 1986) in everyday language. Lakoff and Johnson give many instances of linguistic regularities that might best be explained by assuming some sort of metaphorical or analogical structure that allows the interpretation of a set of items in one domain in terms of a set of items in another. For example, they give the example of the "up is good" conventionalized metaphor. This metaphor is to account for a wide variety of phenomena, including

such utterances as "I'm feeling up today" or "Things are beginning to look up".

The import of this work to the task at hand is as follows: The conventionalized metaphor Lakoff and Johnson point out *structure* our cognitive system. Thus in representing the content of sentences that involve such metaphors, this structuring must be taken into account. Below I argue that many rather mundane natural language forms are best represented using a facility based on a notion similar to Lakoff and Johnson's metaphors.

4. KODIAK

KODIAK (Keystone to Overall Design for Integration and Application of Knowledge) is a knowledge representation language being developed at the Berkeley Artificial Intellgience Research Project. KODIAK is an attempt to redress the above grievences, and to incorporate facilities for dealing with non-factual representations. KODIAK allows for the multipicity of concepts required by any system, but does not abandon the criterion of canonical form that is usually not adhered to by frame-based systems or semantic networks.

4.1. Basic KODIAK Notions

4.1.1. Relations

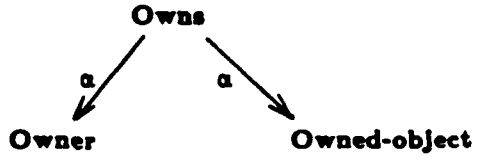
KODIAK is a *relation-based* system. The most important element of KODIAK is the *named relation*. The idea is that what is usually called a slot in a frame is really a particular relation. In KODIAK, this relation, rather than the frame, is the primary object.

KODIAK relations have a fixed number of argument positions. Moreover, each argument position of a relation is itself a full-fledged object. In general, the meaning of these argument-objects is derived from the named relation that hold between them. For this reason, we call such argument-objects *aspectuals*.

For example, we might posit a named relation called **Owns**. This relation has two aspectuals, namely, **Owner** and **Owned-object**. While we have not yet established any meaning for these terms, the idea is that **Owns** denotes the concept of something being owned by someone, and **Owner** the concept of an owner. **Owned-object** denotes the idea of an owned-thing. Note that these are all unique symbols; a reference to the same name elsewhere is a reference to the same concept.

We represent the fact that a particular named relation holds between a particular set of aspectuals using a special relation called α. Relations like α are special in that they are knowledge to the KODIAK interpreter (that is, the set of programs that uses the representation). Therefore, these are called *epistemological relations*. (Other epistemological relations are introduced below.) If a named relation is α-related to an aspectual, we say that the relation *manifests* that aspectual.

We diagram the **Owns** relation and aspectuals it manifests as follows:

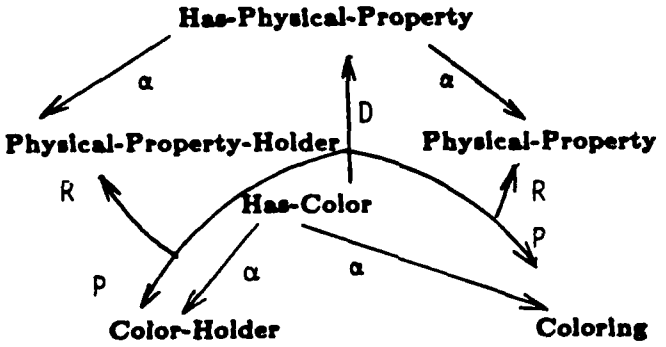


Another example of a relation is **Has-Color**. The aspectuals this relates are termed **Color-holder** and **Coloring**. These three objects are supposed to convey the idea that certain kinds of objects can "have" color. To make this relation meaningful, we will have to express the fact that the *relationship holds between* physical objects and colors*. We will do so in a moment by making predications about the aspectuals of this relation.

The point of having a particular **Has-Color** relation is to express the fact that physical objects manifest "have" colors in a very particular way. For example, the idea of owning something also involves a kind of having, but presumably of a very different sort. Similarly, the idea of a physical object having a weight, say, would seem to resemble the way in which a physical object has a color, but much less so the idea of ownership. These are the distinctions not captured by frame or semantic network systems that we hope to capture here.

By the doctrine of canonical form, we are obliged to capture the commonality in these relations, as well as express their differences. As we suggested above, this can normally be done by a conventional ISA-type hierarchy. Therefore, we introduce another epistemological relation. This one is called **DOMINATE**. This is modeled after the inheritance cables of KL-ONE. That is, **DOMINATE** permits a number of additional relations between the aspectuals of one relation and that of another in a **DOMINATE** relation to it. These relations are called **Role/Play** links.

For example, we might introduce another relation called **Has-Physical-Property**. This relation may come with the aspectuals **Physical-Property-Holder** and **Physical-Property**. We might then indicate that having a color is a physical property by the following representation:



That is, **Has-Color** is a kind of **Has-Physical-Property** where **Coloring** *plays* the *role of* the **Physical-Property**, and **Color-Holder** the role of the **Physical-Property-Holder**. The relation between **Has-Weight** and **Has-Physical-Property** would be similar.

Note that in this example, the object **Physical-Property** is not meant to be merely a meaningless place holder, i. e., a slot in a frame. Rather, this is a meaningful object in its own right: It denotes the concept "being a physical property". Furthermore, we could introduce a **Has-Property** relation that **DOMINATE**s **Has-Physical-Property**, and which manifests the aspectual **Property**. This aspectual would then denote the concept underlying a sense of the English word "property".

While every aspectual is a distinct concept, not every one will be familiar. For example, the concept of "property" is readily recognizable, but not so that of "being a property holder". The

*Actually, a "coloring" should include some sort of color pattern - something more complex than a single hue. Also, some category more general than physical objects can manifest colorings. We will ignore these details in the example.

existence of an aspectual in KODIAK does not require its lexicalization in language.

Intuitively, the concept of being a property requires that there be some object that manifests that property. Thus the concept underlying "property" presumes a relation to some other concept. This is the motivation for calling such concepts aspectuals. In contrast, the idea of something having a property seems relatively complete. Hence these kinds of objects are called *absolutes*.

In addition to named relations, KODIAK has other kinds of absolutes. These correspond to the objects found in most representational systems. For example, **Physical-Object**, **Person**, and **Red** are all KODIAK object absolutes. The intention in KODIAK is to derive the meaning of object absolutes from that of relations and aspectuals, although it is not clear this can always be done. We will discuss the derivation of absolutes below.

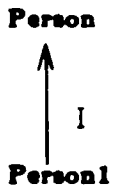
As in most representation languages, objects can be in **DOMINATE** relations to one another. In addition, we can use **DOMINATE** relations to express constraints on aspectuals. For example, to express the fact that the **Physical-Property-Holder** aspectual is always a physical object, we can assert a **DOMINATE** relation between this aspectual and the absolute **Physical-Object**. Thus the semantics of **DOMINATE**, while well-defined, differs somewhat depending of the types of the objects associated by it.

Unlike frame-based languages, object absolutes in KODIAK do not have slots. Rather, they may participate in various relations. It seems reasonable to specify aspectuals for relations without running into the problem of "slot proliferation" described above. Thus, by attaching aspectuals only to relations, and not to object representations, the problem of unprincipled slot attachment can be circumvented.

4.1.2. Structured Mappings

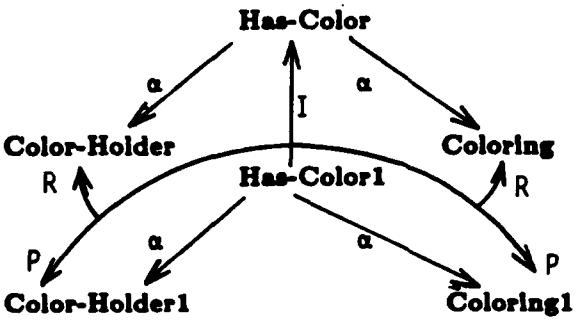
The **DOMINATE** relation is one instance of a general class of KODIAK relations called *structured mappings* (The term is borrowed from Gentner (1983), who uses structured mappings as a framework for analogy). These are where the basic representational power of KODIAK may be found. For example, while **DOMINATE** relates two objects in a class-subclass type relationship, the structured mapping **INSTANTIATE** relates two objects in an class-membership type relationship.

The simplist examples are the representation of instances of object absolutes. For example to denote that fact that some particular individual is a person, we would use the following:



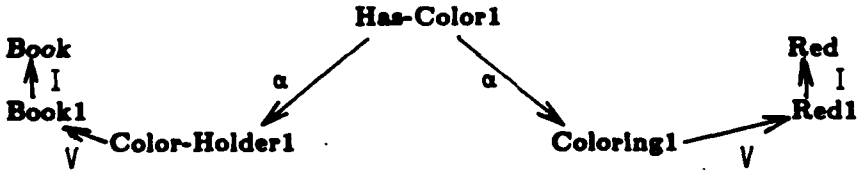
Both **Person** and **Person1** are KODIAK absolutes.

A more interesting use of **INSTANTIATE** involves representing particular facts. For example, as shown above, we can represent the idea of having a color by the particular KODIAK relation **Has-Color**. Now, to represent the fact that a particular object has some particular color, we would do the following:



That is, we create a new relation between new aspectuals. These are all unique objects. In most cases, we would have additional information to represent. For example, if the pertinent natural language utterance refers to a red book, then we would add this information by making further assertions about **Color-Holder1** and **Coloring1**.

However, we do not represent this information by asserting a fact directly about these objects, for example, by asserting that **Coloring1** is a red color. In KODIAK, aspectuals such as **Coloring1** denote intentions. For example, if the assertion **Has-Color1** is about John's book, say, then **Coloring1** would denote the concept "the color of John's book". To specify the fact that John's book is red, we need to introduce a new KODIAK relation, called **VALUE**. **VALUE** relations always hold between an aspectual and some absolute. For example, to assert that **Has-Color1** holds between something that is a book and some red hue, we would add the following to the representation shown above:



The point of this notation is that we can represent the meanings underlying sentences like "Bill didn't know the color of John's book" because we have separated the assertion about what the color is from the idea of the particular book having a particular color. A natural language understanding system could produce the same representation for "the color of John's book" regardless of whether the phrase appears in a context in which its value is also present, or in which its value is unknown.

Such concepts are difficult to represent in most frame and semantic network based systems. However, the representation shown here appears to be similar to that used in KL-ONE.

Unlike KL-ONE, however, in KODIAK there is no such thing as an individual concept per se. Rather, the notion of an individual is meaningful only with respect to another concept. For example, all of the rather general category concepts mentioned above may be individuals of other categories. For example, all of them could be individuals of the concept **Category**, should we introduce such a term in the system. Less abstractly, it is not unreasonable to postulate concepts like **Good-Idea** which may have "generic" concepts, like **Canned-beer**, among its individuals.

The particular properties of some concepts that usually leads to typing objects "individual" or "generic," as in KL-ONE, are here considered to be peculiar properties of physical objects rather than something intrinsically representational in nature.

4.3. Examples

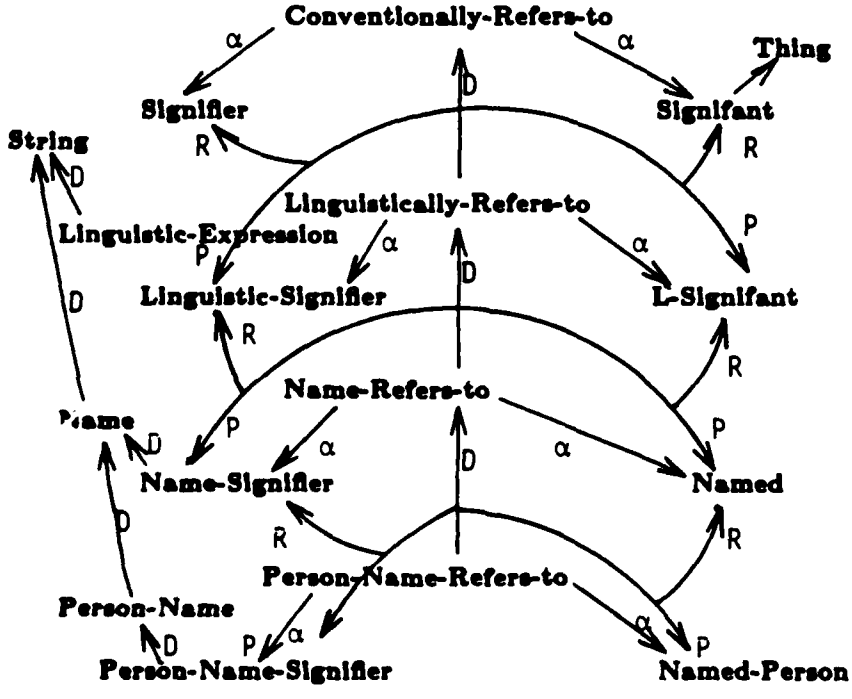
At this point, it is useful to present a few examples of the KODIAK notation introduced so far.

Consider first the simple fact of representing someone's name. For example, a natural language processing system that reads the sentence "John Q. Smith accepted his party's nomination" needs to represent the interpretation of "John Q. Smith" as the name of some person. In a frame-based system, we might represent this fact by putting this string in the "Name" slot of some individual person frame. However, as suggested above, this is a rather meaningless assertion.

In KODIAK, we define a relation that captures the idea of naming. We make this relation meaningful by relating it to other relations. Thus, we introduce a **DOMINATE**-hierarchy of relations beginning with a relation called **Conventionally-Refers-to**. This relation will take two aspectuals, called **Signifier** and **Signifant**. Thus, this relation designates the most general notion of using signs to refer to things.

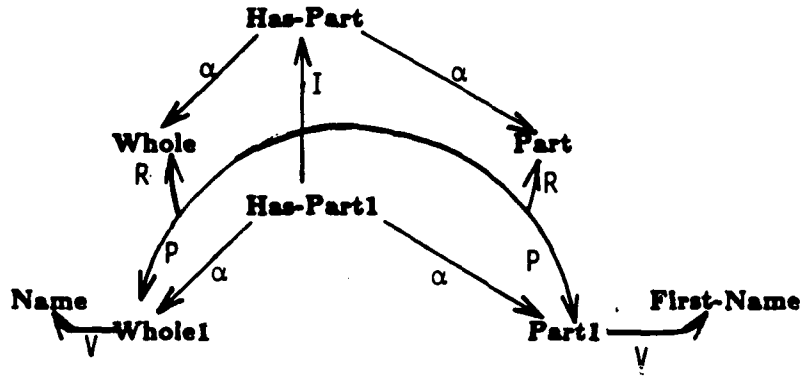
Conventionally-Refers-to itself **DOMINATE**s **Linguistically-Refers-to**. The aspectuals of this concept differ from those of its ancestor only in that **Linguistic-Signifier** is **DOMINATE**d by **Linguistic-Expression**. That is, we are specializing the notion of referring via signs to that of referring via languages.

We specialize this further by introducing the concepts **Name-Refers-to** and a specialization of this, **Person-Name-Refers-to**. Parallel with these are further specializations along the object hierarchy, introducing **Name** (a specialization of **Linguistic-Expression**), and **Person-Name** (a specialization of **Name**). These notions **DOMINATE** the corresponding aspectuals of the "refers" relation hierarchy. This gives us the following initial set of KODIAK assertions:



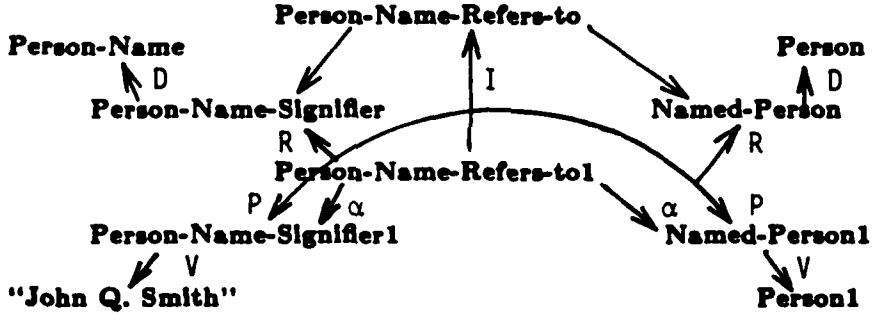
Of course, the above diagram is rather incomplete. **Person-Name** itself requires further structure. For example, we should introduce the concepts of **First-Name**, **Middle-Name**, and **Last-Name**. These would each be in a **Has-Part** relation to **Person-Name**. **Has-Part** in KODIAK is just another named relation. It bears the aspectuals **Part** and **Whole**. Therefore, to

represent the fact that a first name is part of a name, we would produce the following KODIAK assertion:



Of course, other KODIAK assertions would be necessary to describe other aspect of how names work, such as the relative name order of the name components. In addition, the names just described are actually Western names, and other conventions and intermediate nodes in the hierarchy would be appropriate for an accurate description. We will ignore this level of detail from now on, but note that it is in fact in the spirit and capabilities of KODIAK to have such a detailed network.

Having produced such a structure, we can represent a particular fact about Mr. Smith:



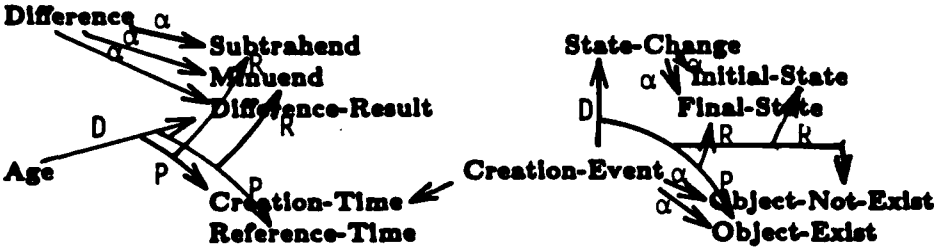
In this example, as in all KODIAK assertions, the same name always represents the same concept. Thus the symbol **Person-Name-Refers-to** here refers to exactly the same object as it did above. There are no slot-like entities whose interpretation is context-dependent.

Person1 is a node designating a particular person. Subsequent assertions about this individual would be made by referring to this particular object. The concept **Named-Person1** means something like "the person named "John Q. Smith".

To represent the fact that "John Q. Smith" is cognized as composed of a first, last and middle name, we would have to include particular instances of the **Has-Part** relation. This is left as an exercise for the reader.

Age

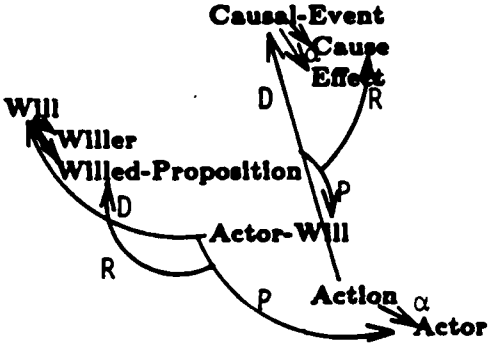
As mentioned about, a strong motivation for KODIAK was to be able to represent the semantics of concepts like "age". Given the above relations, we can define an **Age** concept which is the difference between the creation of a thing and some other time:



In this representation, **Age** is represented as a **Difference-Result** of the **Difference** between **Creation-Time** and a reference point. **Creation-Time** is further defined, although the representations of **Object-Exist**, etc., are abbreviated.

Action

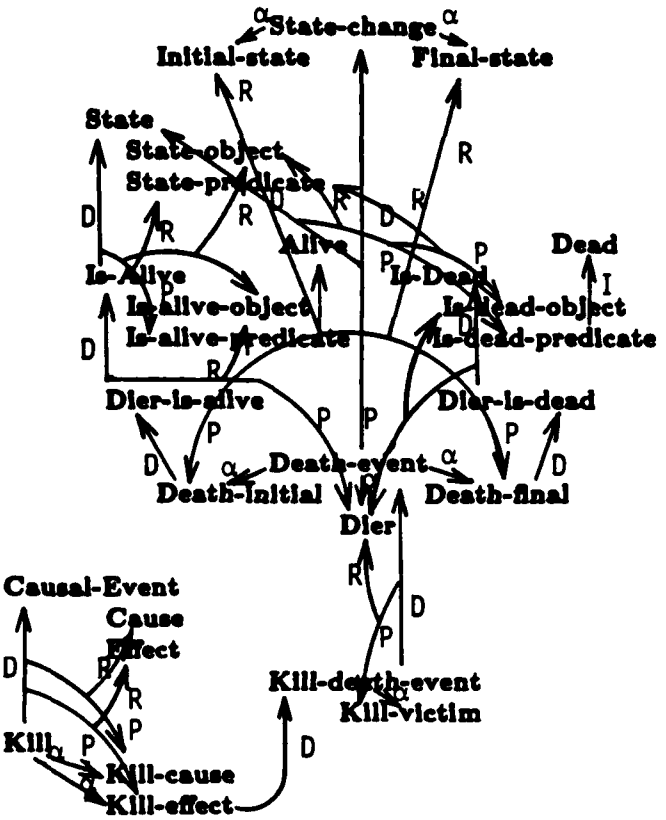
In KODIAK, an **Action** is just another type of **Causal-Event**. In particular, it is the class of such events where the **Cause** is the **Actor** willing some intended state. We can thus represent the general idea of **Action** as follows:



Here we neglect to represent that the concept **Will** is a kind of **Mental-State.**

Kill

We also use **Causal-Event** in our analysis of "kill". This is represented as a causal whose effect is someone's death. Death is itself represented as a state change from being alive to being dead.



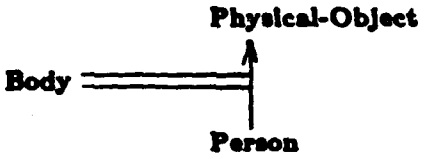
4.3. Views

An important aspect of the theory underlying KODIAK is the representation of non-factual information. In particular, we want to be able to talk about viewing one concept in terms of another. This idea was first suggested as a representational technique in MERLIN (Moore and Newell 1973) and in KRL (Bobrow and Winograd 1977). KRL does not admit to a notion of definition, and treats all perspectives as equally valid. We do not adopt this extreme position, but want to allow the flexibility of viewing a (possibly defined) concept as something other than its "ordinary" interpretation.

For example, it is desirable to realize that a person can have properties, such as weight and color, that are generally considered to be general properties of all physical objects. In most representational schemes, to capitalize on this knowledge about physical objects, it is necessary to assert that persons are a kind of physical object. This is peculiar, because such a view of people is at odds with a normal working distinction between people and physical objects.

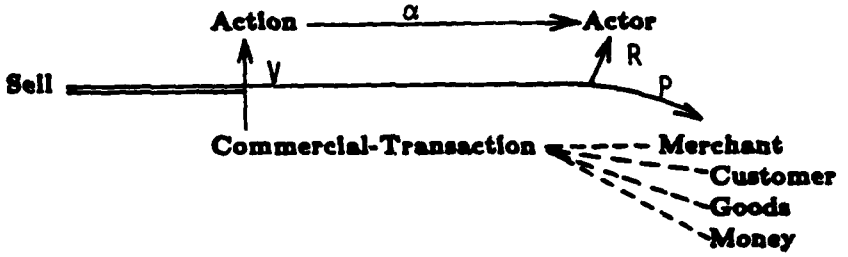
In KODIAK, we resolve this problem by introducing the relation **VIEW**. The idea is that one concept can be thought of in terms of another. In addition, this view of one concept as another is itself a concept . For example, in KODIAK, we can assert that **Person** is **DOMINATE**d by **Living-Thing**, or some such concept. In addition, we also assert that it is possible to **VIEW** a **Person** as a **Physical-Object**. Moreover, the **VIEW** of **Person** as a **Physical-Object** is itself another concept. Namely, it is the concept **Body**.

We depict this view graphically as follows:



Views can be used to address some representational problems in which canonical form and epistemological adequacy appear to be in conflict. In particular, consider the representation of the meaning of sentences involving the words "buy" and "sell". In Conceptual Dependency, sentences such as "John sold Mary the book" and "Mary bought the book from John" are represented identically. The rationale for this is that they mean the same thing. This seems to be truth-conditionally correct. However, it then becomes impossible to have separate concepts of buying and selling, which are useful for many purposes.

We can combine views with some of the notions that of frame semantics (Fillmore 1982, Kay 1983, Fillmore and Kay 1983) to solve this problem. In particular, we postulate a background frame called "commercial transaction" that both "buying" and "selling" refer to. We can then define buying as being this commercial transaction scenario, but viewed as an action from the point of view of the fellow with the money. Similarly, we can define sell as commercial transaction viewed as an action from the point of view of the fellow with the goods. We can the "viewpoint" via the use the role-play relations. Here is the definition of sell in this analysis:



To simplify this example, the details of **Commercial-Transaction** have not been specified, although a number of aspectuals occurring within its subcomponents are shown.

The representation for **Buy** is defined similarly. Note that, with views, the potential admission of seemingly non-canonical entities like "buy" and "sell" is overcome. These both have distinct representations, although the majority of their representations are identical, as the doctrine of canonical form suggests they should be.

4.4. KODIAK and Representational Principles

The use of hierarchies plus structure mappings allows KODIAK to maintain the representational scope of other systems, while at the same time making it possible for the system to conform to canonical form. Thus KODIAK analyses are meaningful in the way CD analyses are, but do not have the epistemological inadequacies of CD. For example, we can represent such fine distinctions as the difference in understanding between an explicit description of a complex concept, and a reference to that concept. Thus, representing the meaning of the sentence "John caused Mary to die" as an instance of **cause** rather than of **kill** represents an understanding of this sentence at one level. Adding the fact that this instantiates **kill** would represent the additional realization that this was an instance of killing.

The existence of aspectuals and relations means that the idea of being a slot in frame has a clear (or at least, clearer) interpretation in KODIAK than it does in an ordinary frame system. Rather

than stating that the filler of a slot is somehow related to the frame, in KODIAK we state that a rather specific and well-defined relation exists between the two objects. Thus, while **NAME** of **Person** may be undefined in most frame systems, the comparable statement in KODIAK presupposes persons, names, and a naming relation which captures some of the semantics of naming.

In general, KODIAK pays homage to Cognitive Correspondence in a number of ways. Views relate to Cognitive Correspondence in that they enable us to differentiate truth-conditionally equivalent concepts that nevertheless seem to be cognized distinctly. Also, concepts like "property", "part" and "actor" are explicit concepts in KODIAK, having as significant a status as concepts denoting objects or relations per se. As a consequence, facts about aspectuals can be made by simply referring to the node denoting the aspectual. In a system like first-order logic, aspectuals exist only as positions in a predication. Thus, stating a fact about one requires a complicated universal predication. This is both cumbersome and cognitively unappealing.

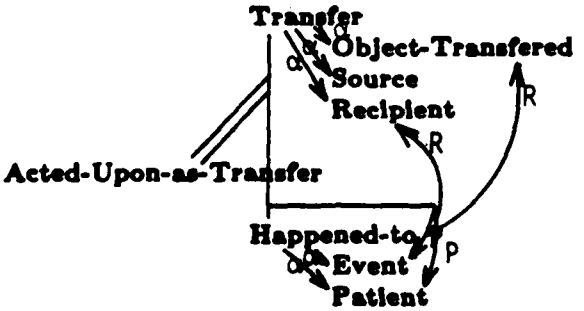
4.5. KODIAK and Language Knowledge Representation

4.5.1. Views and Language

Views appear to be particularly important in representing knowledge about language. In particular, they are useful for representing cognitive structures that do not denote facts so much as how one thinks about the world. For example, in our analysis above, we concluded that buying and selling are factually equivalent, but differ nevertheless in the interpretation of this factual content (i. e., that the same commercial transaction may be thought of as two different actions).

Paul Jacobs (1985) points out that many otherwise unstatable linguistic regularities can be captured using views. For example, he points out that there are many cases like "John took a punch from Bill" and "Bill gave John a punch" in which it appears that punching can be viewed as a transfer, at least for the purposes of linguistic expression. In contrast, considering these two expressions as independent idioms would not capture what would appears to be a substantive regularity.

Jacobs' solution is to represent a "being acted upon as transfer" view. That is, acting upon an object can be viewed as transferring the action to that object. This could be represented as follows:



Here **Happen-to** denotes the notion of being acted upon. This is technically a relation between an event, and an object that event may be directed toward (called the **Patient** above). Since punching involves a **Happen-to**, a language generator could use this view during generation to map an instance of punching into an instance of transferring*. Then knowledge about how to

Actually punching would be related to **Happen-to** via an additional view that relates the actor of the event to the source of the transfer. This complexity is omitted to simplify the exposition.

express transfers can be used to produce the actual linguistic expressions. This sort of language knowledge representation, as well as its role in generation, is explored in detail in Jacobs (1985).

Note that during understanding, it is plausible that a language analyzer seeing an expression like "Bill gave John a punch" would first produce a representation of this as an instance of "giving". However, noticing that transferring may be a view of certain kinds of other events, the analyzer would know not to stop here. Rather, it would eventually recognize that this fits the "acting upon as transfer" view. The analyzer could "unmap" this view to produce the more literally correct interpretation of the input, namely, as an instance of "punching".

Additionally, we could have a "relation as possession" view, which we use to express relations in terms of possessive expressions in English. Thus expressions such as "John's girlfriend", "John's apartment", "The girlfriend John has" or "The girlfriend of John" would all be analyzed initially into expressions involving possession. By the presence of the "relation as possession" view, the analyzer would know not to take literally statements referring directly to possession. Rather, it would attempt to "unmap" such representations to produce the particular relation as its content.

This use of view for linguistic representation seems in accordance with Lakoff and Johnson's (1980) description of the role of metaphor in ordinary language, and Lakoff's (1986) observations about the role of metonymy. In fact, we suspect that there are two kinds of views, metaphoric and metonymic. For example, the view of "commercial transaction as action" is metonymic, because there is an action component to commercial transaction. The "being acted upon as transfer" view is metaphoric in nature, in that one idea is being thought of in terms of another.

As another example, consider Fillmore's (1985) description on the constraints of the use of the word "on" in forms like "on the bus". As pointed out by Fillmore, such usage is correct only when the bus is actually in service. For example, it would be appropriate to say that "John was on the bus during the earthquake" if the earthquake occurred when John was taking the bus to work. However, it would not be proper to say this if the bus had been abandoned and John had taken shelter in it. In that situation, the use of the preposition "in" would appear to be more satisfactory.

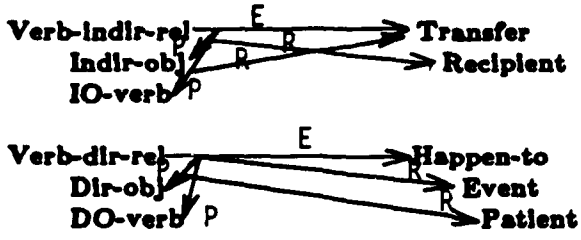
Fillmore points out is that it seems inappropriate to ask the question "Is it *true* that John was on the bus during the earthquake if he took shelter in an abandoned bus?" Rather, whether the correct qustion is whether making such a statement is *appropriate* in the situation. This observation fits in nicely to the scheme suggested here. For example, one can assert a linguistic fact that the preposition "on" can be used to describe the relationship between a conveyance and the object it conveys. In addition, one can suppose a "mass transit medium in operation is a conveyance" view. One would then be able to speak of "being on the bus" as encoding a (metaphoric) view of a bus in service as a conveyance. This would explain Fillmore's observation that it is inappropriate to refer to the "truth" of this assertion, because it is not clear in general what it means for a metaphor or a metonymy to be true, as opposed to appropriate.

4.5.2. EXPRESS

Structured mappings are useful for declaratively representing linguistic-conceptual relations. In particular, Jacobs (1985) uses the relation **REF** to represent word-to-meaning relationships. Actually, this is something of a misnomer, and we use the term **EXPRESS** to more adequate capture the sense of this relation.

For example, a particular word might be in a **EXPRESS** relation with a particular idea. Moreover, a language construct might be in a **EXPRESS** relation with a concept, with the ROLE/PLAYs associating the parts of one with the parts of the other. For example, actions might be expressed as sentences, with the conceptual ACTOR playing the role of the syntactic

subject, and the ACT playing the role of the verb root; acting upon an object might be expressed by the verb-direct object relation, with the particular action being expressed by the verb and the particular object by the direct object; and being the recipient of a transfer can be expressed by the verb-indirect object relation, with a similar associated mapping. We can diagram the verb-object and verb-indirect object as follows:



The various grammatical relations thus identified, verb-subject, verb-direct object, verb-indirect object, etc., are associated with grammatical *templates* (not shown above). Among other things, these templates include grammatical *patterns*, which express the word order of the constituents. Thus an analysis mechanism can use word order to suggest a template, and thereby infer a grammatical relation; **EXPRESS** mappings from these relations to meanings can then be used to suggest semantic interpretations of the utterance. Similarly, the **EXPRESS** links might be traversed from meaning to grammatical relation as part of a natural language expression mechanism.

Among other things, this formalism helps facilitate declarative representations of linguistic knowledge. See Jacobs (1985) for an extensive treatment of these and related issues.

4.6. Reification

It appears that many object concepts can be derived from the aspectuals of relations. For example, consider the **Contains** relation, which has aspectuals **Container** and **Contents**. Of course, **Container** here is the aspectual, so it refers to the idea of being in a containing relation to something, rather than the more typical English usage of the words, to specify an object used for containing. However, the latter concept is still a legitimate one. One would like to derive it from the **Container** aspectual by saying that there is a concept **object-container** that is an object whose function is playing the role of a **Container** aspectual.

This appears to be a rather general type of derivation. Therefore, we would like to be able to derive object concepts from aspectuals by applying some sort of operator to an aspectual, and having it produce an object representation of some object intended for playing the role of that aspectual. We call this process of producing an object representation from an aspectual *reification*.

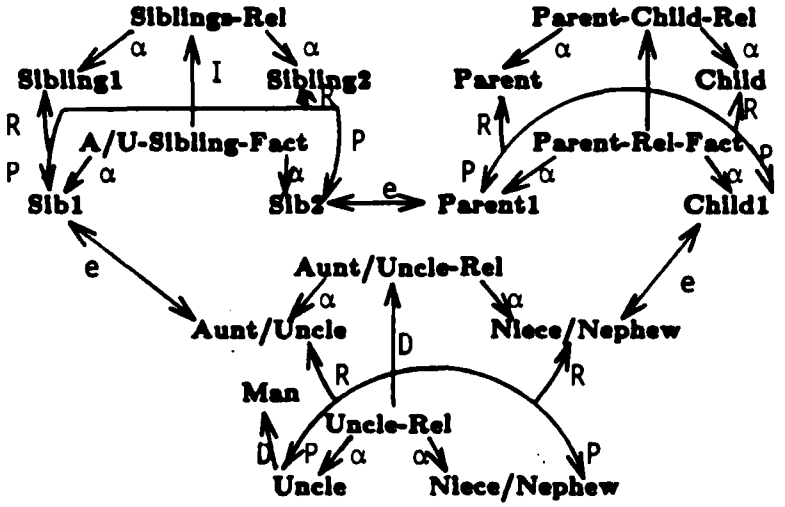
There are several types of reification we have encountered. In addition to "intended for playing the role of", there are "has a proclivity of playing the role of" and "has played the role of". For example, the word "killer", as in "John is a killer" seems to means that John has a proclivity toward killing. The word "murder", as in "John is a murder" seems to mean that John has once played the murder role.

In addition, the notion of a "sign" could be represented as a reification of the "signifier" aspectual, presented in the representation of names in the section on examples. Here the reification is one of "conventionally used for palying the role of". All these are instances of the general KODIAK idea of trying to define objects in terms of relations as much as is possible.

4.7. Other KODIAK Notions

4.7.1. EQUATE

This relation is used to show that two descriptions are constrained to be the same thing. This is used primarily in situations in which it is not possible to use role-play links to connect to roles. For example, consider the following definition of the concept "uncle":



This diagram states that the aspectual **Aunt/Uncle** (intuitively, either an aunt or an uncle) is the sibling of a parent of the **Niece/Nephew** aspectual. This presumes a parent-child relationship, and a sibling relationship, which are not further defined above. The peculiar **A/U-Sibling-Fact** and **Parent-Rel-Fact** and their aspectuals are a technical device needed to provide objects that mean "sibling of Aunt/Uncle" and "parent of Niece/Nephew".

Probably this representation is too abstract — separate and more redundant representations for "aunt" and "uncle", defined in terms of "sister" and "brother" rather than "sibling" may be cognitively more accurate. The same use of **equate** is required, of course.

4.7.2. GENERIC-INDIVIDUAL

This relation is used to define a concept that acts as an exemplar of another concept. Properties that are typically true of a concept but not strictly necessary may be asserted about a concept that is in a **GENERIC-INDIVIDUAL** relation to another concept. Information about "prototypes" can be accommodated in this manner. **GENERIC-INDIVIDUAL** is similar to the ***TYPE** feature of Fahlman's NETL system (Fahlman 1979).

4.7.3. Minial Aspectual Sets

One problem with KODIAK as currently presented is that there is no obvious way to determine what is necessary to do to instantiate a concept. For example, suppose we want to encode the fact that John was killed. Looking at the representation for "kill" propose above, it might appear that a copy of the entire structure constraining **Kill-effect** would have to be copied, in order to assert that some particular individual died. This seems wasteful, since specifying the **Kill-victim** is all the information that is necessary.

Those aspectuals from which the rest of the aspectuals of a concept may be computed are called the *minimal aspectual set*. This set represents the essential unknown information about a particular concept. The minimal aspectual set for **Kill** would include **Kill-victim**, but not **Kill-effect**, say, since latter is determined by the former.

Only those aspectuals contained in the minimal aspectual set of a concept ever need to be specified. For example, in the same representation for "kill", the **Dier** of the **Death-event** concept is specified, but the **Death-initial** and **Death-final** are not, as they are completely determined from the given information.

5. Advantages of the Proposal

Greater Representational Scope

Most of the power of KODIAK comes from the fact that it is a relation-based system. KODIAK is not unique in this respect; it just tends to provide more objects than other systems do. E. g., what are "slots" in other systems are objects (aspestuals) in KODIAK, as is the fact that some concept has a "slot". So not only can we define the concepts implicit in "slot" names, but we can assert information about the kind of "slot", for example, whether a given concept can manifest only one such relation, or several. KODIAK attempts to provide a way to express *how* people think about certain kinds of concepts, in addition to the expressionf of facts.

Uniformity with Canonical Form

KODIAK is rather uniform, making fewer unmotivated and unnecessary distinctions between kinds of concepts, levels of representation, etc. This is true not just in comparison to systems like CD, but in contrast to more uniform formalisms like semantic networks, FRAIL and KL-ONE. However, unlike some of these systems, KODIAK makes an explicit commitment to cognitive correspondence.

Undefined and Partially-Defined Concepts

While some concepts do have definitions, many apparently do not. For example, the concept "Jew" has many things that we know to be true about it, but it is not clear which if any of these are definitional. One may argue that there is a formal definition, e. g., a Jew is someone whose mother is a Jew. But this fact is not known to everyone who has a functional use of the concept, and therefore seems to be beside the point. Furthermore, we have the classic example of Putnam, who claims that ordinary folk have a functional knowledge of the concept "gold", but must appeal to some expert, who really knows what gold is, in order to determine if some item is truly gold or not. In addition, it has been claimed that most natural kind concepts have no definitional information predicated about them. Instead, they are clusters of generally true information.

While KODIAK support concepts with real definitions, it permits concepts that have none. For example, natural kind concepts can be represented by definitionless objects that have many assertions about their "generic-individual." In this manner, any degree of definition is allowable.

Processing Appeal

KODIAK allows for a full and deep meaning representation, but, at the same time, has the property that simple linguistic forms (i. e., one's that seem to be easily understood) can be easily represented. For example, to represent the fact "Bill was killed", we need only create a new symbol designating the particular event, and a new symbol designating the person and then grow the appropriate links. To represent "John killed Bill", we could add further links indicating that the new event is also an **Action**, with "John" being the **Actor**.

Now, if we wished to represent "John killed Bill intentionally", we would first have to have represented the concept **Intended-Action**. This could be represented as a kind of **Action** in which the **Actor Willing** something is the actual **Cause** of that thing. Then the representation of the sentence just entails an additional **DOMINATE** link to this concept.

The advantage here is that we capture the full semantics of these sentences, but do not require processing that seems out of line with the ease with which these sentences can be understood.

6. Comments

6.1. KODIAK and KL-ONE

KODIAK is probably most similar to KL-ONE. Certainly, some goals are shared by both systems, e. g., the desire to ovecome epistemological weaknesses. Also, both promote a proliferation of objects. The systems have other similarities, such as treating slots as objects; the structured mappings of KODIAK are generalized from KL-ONE's notion of cables. In addition, while KL-ONE does not explicitly address all the criteria of representation suggested here, it is not necessarily in violation of them either.

A few minor differences between KL-ONE and KODIAK were noted above. For example, objects in KL-ONE are marked as being generic or individual, while this is a relative distinction in KODIAK. But probably the most significant differences are the following: As suggested above, KL-ONE is subject to the "belonging fallacy", and so it does not meet the criterion of interpretability. In addition, there is nothing corresponding to views in KL-ONE. Hence, the kinds of things represented by them in KODIAK could not be readily represented in KL-ONE.

6.2. Experience with the System

KODIAK implementations have been created and used in a number of tasks. In particular, Peter Norvig has implemented a version of KODIAK that has been used extensively in his FAUSTUS text understanding system, and as the basis for a UC (UNIX Consultant) system. The details of this implementation, and of its application to UC, appear under separate cover.

6.3. Problems

An outstanding feature of both KODIAK and KL-ONE is the proliferation of concepts. Rather than a small set of semantic notions from which all meaning is derived, there will end up being many more concepts in KODIAK than there are words of a given language. This does not appear to be problematic, because, as was argued above, more reductionistic systems seem to end up with such concepts one way or another. What we have provided is a uniform means to represent these

notions, independent of their particular semantic content.

There are many details to be worked out in KODIAK. We have not yet determined how best to assert information about aspectuals. For example, we may want to talk about a α relation as being "one-many", etc., or as being transitive, meaning the concept αs another, which truly αs the one in question.

We are attempting to do without any notion of a variable, and assume that quantification can be accommodated by assertions about general categories to reflect a commonsense "scoping" capability. For example, phrases such as "most books" and "all books" are represented as KODIAK concepts which partition the class of books, and which might be predicated upon, rather than by introducing quantifiers over expressions. At this stage we do not fully understand all the consequences of this assumption.

In general, these issues which appear to be problematic for KODIAK are also problematic for all systems. We are hopeful that the framework established in KODIAK will be able to accommodate solutions to these problems without radical changes, although we have not had enough experience with the system to support such a claim.

7. References

(1) Bartlett, Frederick. **Remembering.** Cambridge University Press, Cambridge. 1932.

(2) Bobrow D. G. and Winograd, T. An Overview of KRL, a Knowledge Representation Language. In **Cognitive Science** Vol. 1, No. 1, pp. 3-46. 1977.

(3) Brachman, R. J. On the Epistemological Status of Semantic Networks. In **Associative Networks: Representation and Use of Knowledge by Computers.** N. V. Findler (ed.). New York: Academic Press, 1979.

(4) Brachman, R. J. et al. Research in Natural Language Understanding. BBN report No. 4274, Cambridge, Mass. 1979

(5) Brachman, R. J., Fikes, R. E., and Levesque, H. J. Krypton: A Functional Approach to Knowledge Representation. IEEE Computer 16(10):67-73, October, 1983.

(6) Brachman, R. J., and Schmolze, J. G. An overview of the KL-ONE knowledge representation system. In **Cognitive Science** Vol. 9, No. 2, pp. 171-216. 1985.

(7) Charniak, E. A common representation for problem solving and natural language comprehension information. *Artificial Intelligence,* 1981, 16, 225-255.

(8) Charniak, E., Gavin, M. K., and Hendler, J. A. The FRAIL/NASL reference manual. Technical reprot CS-83-06, Department of Computer Science, Brown University. 1983.

(9) Charniak, E., Riesbeck, C. K., McDermott, D. **Artificial Intelligence Programming Techniques.** Lawrence Erlbaum Associates: Hillsdale, New Jersey. 1980.

(10) Coleman, Linda and Kay, Paul. Prototype semantics: The English verb *lie*. *Language* 57:1.

(11) Cullingford, R. E. Script Application: Computer understanding of newspaper stories. Yale University Computer Science Research Report #116.

(12) Fahlman, S. E. **NETL: A System for Representing and Using Real-World Knowledge.** MIT Press, Cambridge, MA. 1979.

(13) Feldman, J. A., and Ballard, D. H. Connectionist models and their properties. *Cognitive Science,* 6, pp. 205-254. 1982.

(14) Fillmore, C. The case for case reopened. In P. Cole and J. M. Saddock (eds.) **Syntax and Semantics 8: Grammatical Relations.** Academic Press, New York. 1977

(15) Fillmore, C. Frame Semantics. In **Linguistics in the Morning Calm** Hanshin Press. Seoul, Korea, 1982.

(16) Fillmore C. Frames and the Semantics of Understanding. Unpublisher manuscript, 1985.

(17) Fillmore, C. and Kay, P. Final Report to NIE: Text Semantic Analysis of reading Comprehension Tests. 1983.

(18) Gentner, D. Structure-mapping: A theoretical framework for analogy. In **Cognitive Science,** 7, pp.95-119. 1983.

(19) Hinton, G. E. Implementing semantic networks in parallel hardware. In G. E. Hinton and J. A. Anderson (eds.) **Parallel Models of Associative Memory.** Hillsdale, N. J.: Lawrence Erlbaum Associates.

(20) Israel, D. J., and Brachman, R. J. Some remarks on the semantics of representation langauges. In M. L. Brodie, J. Myopolous, J. W. Schmidt (eds.) **On conceptual modeling: Perspectives from artificial intelligence, databases, and programming languages.** New York: Springer Verlag.

Jacobs, P. S. A knowledge-based approach to language production. Report no. UCB/CSD 86/254. Computer Science Division (EECS), UC Berkeley. August, 1985.

(21) Kay, P. Linguistic Competence and Folk Theories of Language: Two English Hedges. In Proceedings of the Ninth Annual Meeting of the Berkeley Linguistics Society, Berkeley, California, 1983.

(22) Lakoff, G. Linguistic Gestalts. In *Proceedings of the Thirteenth Regional Meeting of the Chicaog Linguistics Society.* 1977.

(23) Lakoff, G. Categories and Cognitive Models. Berkeley Cognitive Science Report No. 2, Institute for Human Learning, University of California, Berkeley. 1982.

(24) Lakoff, G. **Women, Fire, and Dangerous Things.** forthcoming, 1986.

(25) Lakoff, G. and Johnson, M. **Metaphors We Live By.** University of Chicago Press, Chicago, Ill. 1980.

(26) McCarthy, J. and Hayes, P. J. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer and Michie (eds.) *Machine Intelligence,* Vol. 4. New York: American Elsevier, 1969, pp.463-502.

(27) Maida, A. Conceptual Coherence (working paper) 1984.

(28) Minsky, Marvin. A framework for representing knowledge. In P. H. Winston, (ed.) **The Psychology of Computer Vision.** McGraw-Hill, New York. 1975.

(29) Moore, J. and Newell, A. How can MERLIN understand? In **Knowledge and cognition.** Lawrence Erlbaum Associates, Hillsdale, N. J. 1973.

(30) Quillian, M. R. Semantic memory. In M. Minsky (ed.) **Semantic Information Processing.** MIT Press. Cambridge, 1968.

(31) Rieger, C. J. Conceptual Memory and Inference. In R. C. Schank **Conceptual Information Processing.** North Holland, Amsterdam. 1975.

(32) Roberts, R. B. and Goldstein, I. P. The FRL Manual. Technical Report AIM-408, MIT Artificial Intelligence Laboratory. 1977.

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

(34) Schank, R. C. and Abelson, R. P. **Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures.** Lawrence Erlbaum Associates: Hillsdale, New Jersey. 1977.

(35) Woods, William A. What's in a Link: Foundations for Semantic Networks. In **Representation and Understanding: Studies in Cognitive Science.** D. G. Bobrow and A. Collins (eds.). New York: Academic Press, 1975.

END

11-86

DTIC