KODIAK - A Knowledge Representation Language

Robert Wilensky

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

1. Introduction

A new theory of 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. The most important characteristic of the theory is the elimination of the frame/slot distinction made in frame-based languages (alternatively, case/slot distinction found in semantic network-based systems). In its place is a new notion called the "Absolute/Aspectual" distinction.

The theory described here provides a means of representation that has the following characteristics: It is broad and uniform, applying to any number of semantic domains; it is object-oriented; it contains a finite set of primitive epistimological relationships; it has the ability to create new relationships; it is cognitively plausible (i. e., it may reflect how things are represented in the mind); it conforms to other desirata for representations, such as canonical form and usefulness as a memory organizer.

2. The Problem with Frames

As has been pointed out by Woods (1975) and Brachman (1979), the interpretation of most semantic network formalisms is at best non-uniform. Attempts to address these inadequacies has led to the development of systems such as KL-ONE (Brachman et al. 1979). The theory proposed here similarly begins with a dissatisfaction with a number of existing formalisms. It ends up with a new formalism that is not unlike KL-ONE and its descendents in spirit. In detail, the formalism described below makes some different distinctions, and in some cases directly opposes the particular decisions made in KL-ONE and other recent attempts at knowledge representation.

We begin with frame-based systems (Minsky 1975) rather than semantic networks as the starting point. Research on frame-based systems have produced a number of interesting products, arguably, Conceptual Dependency* (Schank 1975) and scripts (Schank and Abelson 1977), which were specific to particular types of knowledge, and KRL (Bobrow and Winograd 1977), FRL (Roberts and Goldstein 1977) and FRAIL (Charniak et al. 1983), which were intended as general frameworks for representation.

In all the general frame languages, it is possible to define frames, and include in the definitions

*Conceptual Dependency preceeded frames historically, but was based on "case frames" that were frame-like in the Minsky sense.

assertions about what slots the frame has. It is also possible to write down arbitrary constraints on what may fill these slots, and to specify defaults for subclasses or instances of that frame. For example, one can define a "Person" frame, and specify that it takes slots for "Age", "Name", and "Address". In effect, such a frame system would be quite similar to a semantic network with a node for Person from which emanated links for Age, Name and Address.

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 just as ill-defined as what it means to be a link in a network*. In particular, the meaning of a slot appears only procedurally, if at all. For example, if we fill the Address slot for some Person with "393 Foxon Road", this presumably means that that person's place of residence is at the location so designated. However, filling in the Name slot with "John Smith" means that the person is called by this name. Unfortunately, this difference in meaning appears only in the way various routines happen to manipulate those slots, i. e., it is encoded procedurally, and therefore, outside of the formal system of representation.

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

There also appears to be no "in principle" answer to the question of which frames can support which slots. For example, if we allow Age to be a slot in Person, and Father (to be filled by the Person's father) to be a slot in Person, could we allow Father's-Age to be a slot? How about Person's best friends between the ages of 25 and 35? Regardless of our own intuitions, the frame languages do not distinguish the suitability one from another.

In actual practice, frame systems users appear to represent such knowledge outside the frame system. For example, "best friends between the ages of 25 and 35" might 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.

Problem 3: Many Concepts Do Not Get Defined

Most importantly, what we have been calling "slots" seem to be perfectly good structured 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 perfectly 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 to some other moment in time. Similarly, Address is a "referring object" for a building; Name is a "referring object" for a person.

In sum, frame systems 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.

3. KODIAK

KODIAK (Keystone to Overall Design for Integration and Application of Knowledge) is a knowledge representation language being created at the Berkeley Artificial Intellgience Research

*Charniak, Riesbeck and McDermott (1980) talk about these languages as "form languages" This nomenclature suggests, I think correctly, that the formalism is more of a form to collect knowledge than a representation of that knowledge.

Project that attempts to redress the above grievences. We view KODIAK as an extention of frames. However, the system is actually no more frame-like than semantic-network-like (which also appears to be the case for the more advanced semantic-network derived languages like KL-ONE).

Like KL-ONE, the primary structure of KODIAK is the *Concept*. However, there is no notion of role, slot or case. Instead, the idea of have a slot or role is replaced by one of a set of primitive epistemological relations. This relation is called **MANIFEST**. A Concept is in a **MANIFEST** relation to another *Concept* when, intuitively, the first *Concept* "has" the second *Concept* as a property. For example, if we want to indicate that physical objects have ages, we could assert that the *Concept* **Physical-Object MANIFESTs** the **Age** *Concept**. Furthermore, once the **MANIFEST** relation has been asserted to exist between two *Concepts*, a new relation comes into existence. This relation lets us assert that particular kinds (or instances) of one *Concept* can **MANIFEST** particular kinds (or instances) of the other. If *Concept1* **MANIFESTs** *Concept2,* say, then we name this relationship *"Concept2-of-Concept1"*. We call such a relationship an **aspectual.** In contrast, we call all other *Concepts*, such as **Age** and **Physical-Object**, **absolutes.**

For example, if we assert that **Physical-Object MANIFESTs Age**, then the aspectual relation **Age-of-Physical-Object** comes into existence. We can use this relation to assert the age of some particular physical object, among other things.

The intuition behind the idea of aspectuals to capture the dual use of terms like "name" and "color". When we talk of the "name of" someone or the "color of" an object, the claim is, we are referring to color as an aspectual (more properly, we are referring to the **Color-of-Physical-Object** aspectual). When we say "red is a color", we are talking about both **Color** and **Red** as *Concept*ual categories. Similarly, **Age** is the *Concept* of age, but **Age-of-Physical-Object** is the "age" implicitly referred to in "John is twelve years old".

In effect, we have split the idea of slot into several parts. One is the idea that a "frame" can have a slot of a certain type (this is expressed by the **MANIFEST** assertion); another is the *Concept* that is the slot (this is represented as another, in principle independent, *Concept*); finally, there is the fact that particulars or subtypes of the "frame" and **MANIFESTed** *Concept* can be in a relation of this sort to one another (this is enabled by the semantics of **MANIFEST**, and expressed by a particular derivative aspectual relation assertion).

It is awkward to talk about the assertion of a relation between two *Concepts*. Therefore, I shall loosely refer to such an assertion as a *link*, and depict it graphically as such.

The advantage of this formulation is that we can provide explicit definitions for and assertions about *Concepts* such as **Age**. In a traditional frame based system, such *Concepts* could not be predicated about explicitly.

For example, we would like to assert that the *Concept* **Age** is the difference between the creation time of an object and some other time (usually **Now**). To do so, we need to introduce some additional epistemological relations.

---

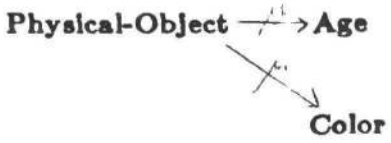
*Of course, we may want to assert this fact about some category more general than **Physical-Object**, so it would be meaningful to talk about the age of an idea, for example. In this paper, I shall not be terribly concerned about the correctness any such assertion. Instead, I will use categories that are familiar rather than those that may be technically necessary to describe properly a conception of the world.

4. Primitive Epistemological Relations

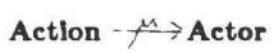
In KODIAK, the following set of epistemological relations is supposed:

MANIFEST

The semantics of **MANIFEST** is described above. We indicate this relation graphically by a directed arrow labelled "μ". Formally (i. e., in non-pictorial language) we indicate this by the form **(MANIFEST** *Concept Property-Concept*). For example, to indicate that a **Physical-Object** has an **Age** and a **Color**, we can draw the following:



Similar, we can indicate that an Action has an Actor:



These examples illustrate several different kinds of **MANIFEST**ation. Maida (1984) has suggested that *Concept*s like **Action MANIFEST Actor** *definitionally* (i. e., the *Concept* **Actor** is defined in terms of the *Concept* **Action**), whereas *Concept*s like **Physical-Object MANIFEST Color** *assertionally* (i. e., this asserts a true but non-definitional fact about the world). In addition, we suggest that **Physical-Object MANIFEST**s **Age** *derivatively* (i. e., the definition of **Age** entails this particular **MANIFEST** relation). See Maida (1984) for a further exploration of these ideas.

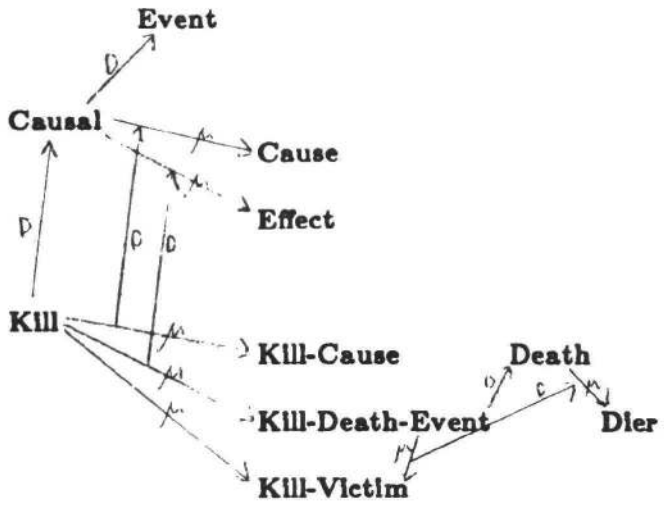
DOMINATE

This is a "structured inheritence" relation between *Concept*s. Its semantics is essentially ISA. We indicate it graphically by a link labelled "D" and formally by an expression of the form **(DOMINATE** *general-concept specific-concept*).

To indicate the relations between the parts of one *Concept* and those of a *Concept* that **DOMINATE**s it, we use an informal relation called "role-play". For technical reasons, this relation is implemented in terms of another, so it is not a true relation of the system. Nevertheless, it is convenient for expositional purposes.

As an example, we propose that there exists a type of **Event** called **Causal**, which **MANIFEST**s a **Cause** and an **Effect**. If we accept the interpretation that **Kill** means "cause to die", this can be represented by specifying a *Concept* **Kill** which **MANIFEST**s, among other things, a **Kill-Victim** and a **Kill-Death-Event**. The latter *Concept* is represented as meaning that the **Kill-Victim** died. We want to establish the meaning of **Kill** now by saying, intuitively, that **Kill-Death-Event** *plays* the *role* of the **Cause**, when **Kill** is viewed as a **Causal** event.

Rather than introduce an explicit role-play relation, however, we take advantage of the fact that the **MANIFEST** relations between **Causal** and **Effect** and between **Kill** and **Kill-Death-Event** give rise to aspectuals. In particular, they create the aspectuals **Effect-of-Causal** and **Kill-Death-event-of-Kill**. Since aspectuals are full-fledged *Concept*s in KODIAK, we can represent the role-play relation simply by asserting that the latter aspecutal is **DOMINATE**d by the former. Thus we have the following graphic depiction:



First, note that these terms refer to the actual *Concept*s. For example, the term **Cause** refers to the idea of "cause", and the term a **Effect** to the idea of "effect". These are not meaningless placeholders in a form.

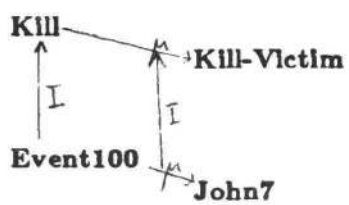
Second, much has been omitted in this diagram, for example, the semantics of **Cause**, **Effect** and **Death**. These are of course a crucial part of the overall system, and are omitted here for simplicity's sake.

Third, note that some *Concept*s, for example, **Kill-Cause**, have no additional semantics associated with them. That is, this is an "empty" *Concept*. **Kill** could have inherited the general **Cause** from **Causal**, so in this case the new name is not strictly necessary. However, it would become necessary if we wanted to make an assertion about the **Cause** of a **Kill** event. In contrast, the *Concept* **Kill-Death-Event** has an explicit definition as a kind of **Death** event.

INSTANTIATE

This relation holds when one *Concept* is to be considered as an instance of another. Its depiction is similar to that for **DOMINATE**s. For example, the fact that some *Concept* represents an individual human being would be represented by an **INSTANTIATE** link between that *Concept* and the *Concept* **Person**. Similarly, a particular killing event would be represented by an **INSTANTIATE** link between the particular event *Concept* and the *Concept* **Kill**.

Like **DOMINATE**, **INSTANTIATE** allows for "role-play" relations between the respective **MANIFEST**ed Concepts. For example, to represent the event in which John was killed, we create a new *Concept*. We call this Concept **Event100**, to suggest mneumonically that it is an event, and to indicate that such *Concept*s are rather numerous. Similarly, **John7** denotes the *Concept* of the person named "John." We then indicate that **Event100** **INSTANTIATE**s **Kill**, and that **John7** plays the role of the **Kill-Victim:**



Again, the representation shown here is abbreviated. For example, the link between **John7** and **Person** is not shown, nor is the information that the first name of **John7** is "John."

Note that in KODIAK, there is no such thing as an individual 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. The properties of some individuals 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 intrinsic to individuals.

As a further example, consider the **War and Peace** problem. The book **War and Peace** is an individual of the *Concept* **Book**. However, the particular copy of **War and Peace** sitting on my shelf is appears to be in the same relationship to the *Concept* **War and Peace** as that *Concept* is to the *Concept* **Book**. This situation can be represented in KODIAK by asserting that the *Concept* **War and Peace INSTANTIATE**s the *Concept* **Book**, and that the particular copy of a book **INSTANTIATE**s the *Concept* **War and Peace.**

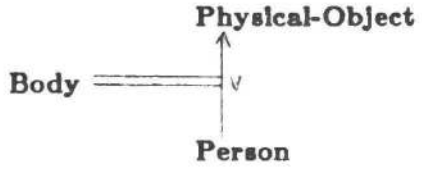
VIEW

An important aspect of the theory underlying KODIAK is that conceptual structure is not monolithic or static. In particular, we want to be able to talking about viewing one *Concept* in terms of another. This idea was first suggested as a representational technique 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**. **VIEW** is similar to **DOMINATE**, except that it does not imply a primary or definitional status to the relation. For example, in KODIAK, we can assert that **Person** is **DOMINATE**d by **Living-Thing**, or some such *Concept*, and also assert that we can **VIEW Person** as a **Physical-Object.**

VIEW is more complicated than the other relations we have seen. This is the case because the **VIEW** of one object as another is itself a full-fledge *Concept*. For example, the **VIEW** of a **Person** as a **Physical-Object** is itself the *Concept* **Body**. Thus we represent **VIEW**s as three-part relations. We depict this graphically as follows:

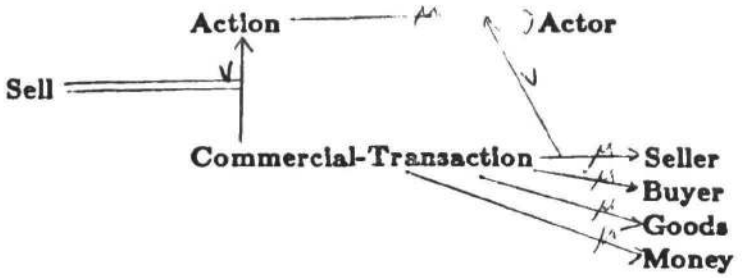


Formally, we can say that **(VIEW** *viewed-concepted viewed-as-concept view-concept*), meaning that *view-concept* is *viewed-concept* viewed as *viewed-as-concept*.

As is the case with **DOMINATE**, we can elaborate on a view by specifying additional **VIEW**s between the derived aspectuals of the *Concepts* participating in the relation.

One application of **VIEW** is to express some of the notions that arise in frame semantics (Fillmore and Kay 1980). In this system, some concepts are defined in reference to a background

frame. For example, "buying" and "selling" are defined in reference to the frame for "commercial transaction". We can represent this with **VIEW** as follows:



Buy is defined similarly.

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).

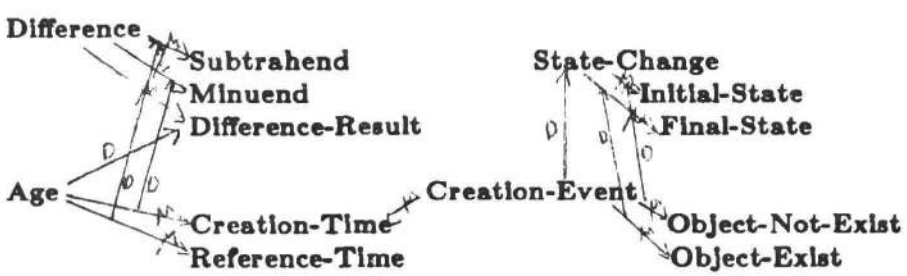
EQUATE

This relation is used to show that two descriptions are co-referential. We shall not elaborate on its use here.

5. Examples

Age

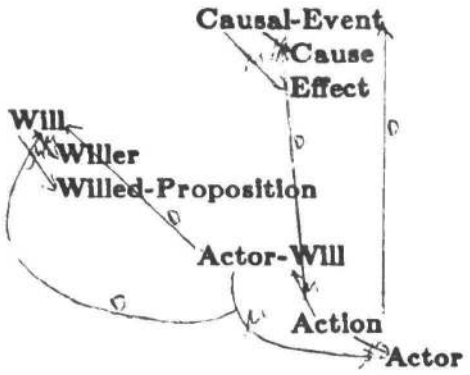
As mentioned about, a strong motivation for KODIAK was to be able to represent the semantics of concepts like "age". Given the above relations, we can define an **Age** *Concept* which is the difference between the creation of a thing and some other time:



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 representation of **Object-Exist**, etc., is 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.**

6. Processing and Representation

One advantage of this representation is that it allows for the 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 symbol designating the new event is also an **Action**, with the symbol designating "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 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.

7. Conclusions

An outstanding feature of KODIAK 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 even more reductionistic systems seem to end up with such concepts. For example, the various knowledge structures of proposed by Schank seem to posit the existence of a large class of elements similar to those explicitly acknowledged in KODIAK. What we have attempted to provide is a uniform means to represent these notions, independent of their particular semantic concept.

Of course, there are many representational problems which the current system does not address. However, most of these appear to be 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.

8. References

(1) 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.

(2) 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.

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

(4) 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.

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

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

(7) Fillmore, C. and Kay, P. Progress Report: Text Semantic Analysis of reading Comprehension Tests. Manuscript. 1980.

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

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

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

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

(12) 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.

(13) 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.