

# Representing Regularities in the Metaphoric Lexicon

James H. MARTIN

Berkeley Artificial Intelligence Research  
Computer Science Division  
University of California  
Berkeley, California  
94720

## Abstract

This paper describes a system for representing knowledge about conventional metaphors for use by natural language analysis, generation and acquisition systems. A system of hierarchically related structured associations is used. These associations are implemented as a part of the KODIAK representation language. Particular attention is paid in this paper to representational mechanisms that can capture generalizations over the system of conventional metaphors as a whole.

## 1. Introduction

The most commonly occurring words in English display a wide range of metaphorically motivated meanings. A knowledge representation system capable of representing these meanings must satisfy two competing requirements. The first requirement is that it must be capable of representing the specific details of each lexicalized metaphor. More precisely it must provide enough knowledge to permit the easy recognition and interpretation of a particular use when it is encountered. The second requirement is that the representation must capture the relevant generalizations manifested by the system of metaphors underlying the lexicon taken as a whole. The system therefore must represent both highly specific facts and capture systematic generalizations.

## 2. Conventional Metaphors

A conventional metaphor (Lakoff and Johnson 1980) consists of source and target concepts. The target concept is the concept that is actually under consideration. The source specifies the concepts in terms of which the target concept is being viewed. When we refer to metaphoric knowledge we mean explicit knowledge of the conventional associations between various source and target domains. These associations are a conventional part of the language.

The first requirement therefore is that the representation must be able to capture individual metaphors as concepts consisting of sets of associations between source and target concepts. Consider example (1).

### (1) How can I kill a process?

This example, from the UNIX domain, involves the conventional metaphor that to kill an ongoing process means to terminate it. The target concepts involve computer processes and the actions that terminate them. The source concept is that of the action of causing a living thing to die. The metaphor consists of the source, target and the set of associations linking them. The associations specify the correspondences between the source and target concepts. In this example they must specify that: the concept kill corresponds to terminate, the victim of the killing is the process terminated and that the killer is the agent of the termination.

## 3. Metaphorical Regularities

It is not enough for the representation to simply account for each individual conventional metaphor in the lexicon. The set of metaphorical word senses in the language taken as a

whole exhibits a number of regularities that must be accounted for. This paper will consider two of the most important regularities. The first is the metaphoric use of different words motivated by a single underlying metaphor. The second regularity has to do with the use of the single words with metaphors that exhibit various degrees of similarity.

Consider the following examples.

- (2) Mary has a cold.
- (3) John gave Mary a cold.
- (4) Mary got a cold from John.

The various uses of the verbs have, give and get in these examples are not instances of isolated metaphors. They all arise from a common conventional metaphor centered around the idea of state as possession. This metaphor entails to be in an infected state is to possess a disease. The transfer metaphor underlying (3) and (4) is a natural extension to the core possession metaphor. In the source domain, the concepts giving and getting are events that result in a having. In the same way the concept infecting leads to an infected state. Therefore the metaphor infection-is-transfer follows coherently from the core possession metaphor and the structure of the source and target concepts. The representation should efficiently capture the fact that coherent underlying metaphors give rise to surface metaphors that may involve several closely connected words.

Now consider these examples.

- (5) John gave Mary a cold.
- (6) John gave Mary the flu.
  
- (7) Mary gave John a good idea.
- (8) Mary gave John a cold.

The similarity of examples (5) and (6) is clearly a result of the application of the same metaphor to two closely connected target concepts (infection with either a cold or the flu). The transfer metaphor in this case is at a level of abstraction that subsumes both of these examples. The representation must therefore permit the representation of metaphors at an appropriate level of abstraction.

Examples (7) and (8) are however not in the same kind of close similarity relationship. The metaphors underlying these examples have completely dissimilar target concepts. Nevertheless they do exhibit an underlying structural similarity. In both cases an abstract effect upon a person is viewed as a transfer of an abstract concept to the effected person by the agent responsible for the effect. A fine grained representation involving the hierarchical sharing of subparts will be used to capture this more distant similarity and still permit the representation of the distinct target concepts.

The following sections will introduce the various mechanisms that will be used to both represent the specific details of individual metaphors and capture the necessary generalizations and regularities.

## 4. Representational Details

The representation language used to capture conventional metaphoric knowledge is KODIAK (Wilensky 1986). KODIAK is an extended semantic net language in the spirit of KL-ONE (Brachman 1985). The use of structured associations

\* This research was sponsored in part by the Defense Advance Research Projects Agency (DOD), Arpa Order No. 4871, Monitored by Space and Naval Warfare Systems Command under Contract No. N00039-84-C-0089.

as described here for representing metaphoric knowledge has been successfully applied to the following tasks: natural language generation (Jacobs 1985), text understanding (Norvig 1986, Jacobs 1987) and the acquisition of new word senses (Martin 1987).

In the following sections, pieces of a KODIAK knowledge base are shown with a graphic representation. For the purposes of this paper it is sufficient to understand that rectangles represent coherent concepts that can stand on their own. These are called absolutes. Concepts in circles are called aspectuals. Aspectuals are concepts that are meaningful only relative to a related absolute. Absolutes are connected to their aspectuals via an S (Slot) link. Aspectuals are given type information via the C (Constrains) link that indicate the kind of concept that can fill this aspectual. Arcs labelled D (Dominate) are a kind of ISA or inheritance link. For a more in-depth description of KODIAK and this graphical representation see Norvig (1987).

#### 4.1. Representing Single Metaphors

Figure 1 shows the KODIAK representation of the source domain from (1). It states that a killing is a kind of action with a result that is a death-event which is in turn an event. The kill-victim of the killing is an inherited role from action indicating that the kill-victim is effected by the action. The kill-victim is constrained to be a living-thing and the killer must be an animate-agent. Finally the equate links require that the kill-victim must be the same as the dier participant of the death-event.

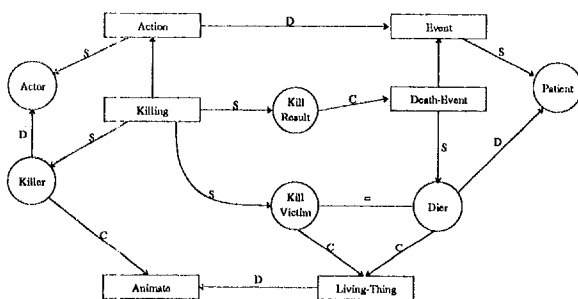


Figure 1: Killing

Figure 2 shows the corresponding concepts from the target domain. It states that a terminate-action-process is a terminate-process which is a kind of action. The terminated-process role is an inherited role specifying the patient of the action. The result of the action is a terminate-process-effect which is a kind of terminate-event. Finally the terminated-process is equated to the terminated-process-event of the terminate-process-effect. This is analogous to the relationship between the kill-victim and the dier shown in Figure 1.

What is needed is a way of associating the appropriate source and target concepts. Such an association is realized in KODIAK by using a relation called a *metaphor-map*. A metaphor-map is a kind of VIEW relation whose aspectuals specify corresponding source and target concepts. A VIEW is a primitive KODIAK relation that permits a limited inheritance between concepts without requiring a strict ISA relationship.

Metaphor maps are needed to link all the core source concepts in Figure 1 to their counterparts in the target domain. In particular the killing maps to the terminate-action, the kill-victim maps to the terminated-process, the killer maps to the actor of the terminate-action and the result of the killing maps to the result of the terminating. It is the co-occurrence of all these maps that constitutes the conventional metaphor that terminating a process can be viewed as

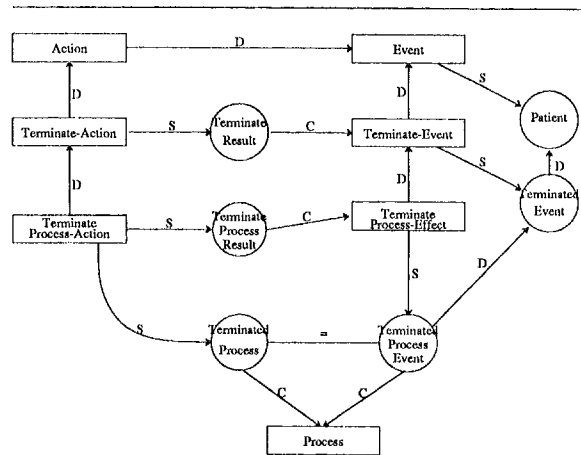


Figure 2: Terminating

a killing.

This conventional co-occurrence of a set of metaphorical associations is called a *metaphor-sense*. A metaphor-sense is simply a KODIAK concept that ties together a coherent set of metaphorical maps. A metaphor-sense can participate in an unspecified number of component-map relations. A component-map relation is a kind of component-association relation that holds between a metaphor-sense and a metaphor-map. Figure 3 shows the metaphor-sense kill-terminate-metaphor that ties together all the metaphor-maps underlying example (1).

To a significant extent metaphor-senses are the minimal meaning-bearing unit of conventional metaphors. Metaphor-maps represent the building blocks out of which meaningful metaphor-senses are constructed. The metaphor-sense represents the level at which one would say that there is a conventional metaphor that to terminate something is to kill it. This level of representation will frequently correspond to a single metaphorical word sense.

#### 4.2. Representing Extended Metaphors

As illustrated above with examples (2) through (4) it is frequently the case that a single core metaphor is the basis for the extended metaphorical use of a number of related words. Consider the following examples.

- (9) John gave Mary a cold.
- (10) Mary has a cold.

A metaphor-sense accounting for (9) with the concepts underlying the word *give* is shown in Figure 4. This metaphor-sense represents the conventional metaphor that to infect can be viewed as a transfer. The relevant maps to this discussion are the *give-infected*, *give-cold*, and the *give-res-inf-result* mappings. These mappings represent the ideas that recipient of the giving is viewed as the newly infected person, the given thing is viewed as the cold and the result of the giving is the result of the infecting action.

Now consider the schema underlying (7) as shown in Figure 5. In this case the haver is the person with an infection, the cold is the thing had and the having corresponds to the state of being infected.

Figure 6 shows the combined source concepts from the *have-a-cold* and *give-a-cold* schemas. This diagram is adapted from Norvig (1987). In it we can see how the source concepts from these two schemas are closely connected. In particular: the *giving-result* is constrained to be a kind of

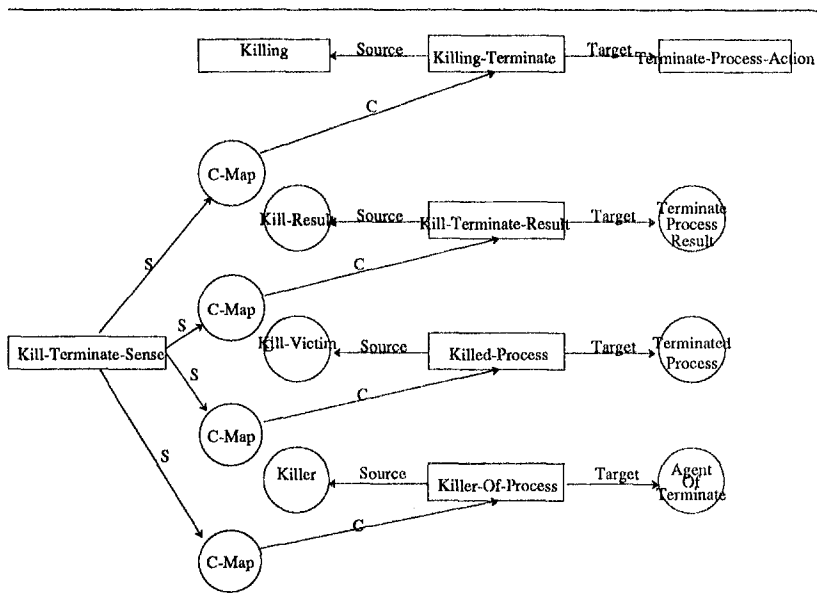


Figure 3: Kill-Terminate-Sense

having (one that results from a giving), the *givee* of the giving is equated to the *haver* of the *givee-has-given* having. Finally the *given-thing* is equated to the *had* of this same having. Therefore all of the source concepts of the *have-cold* schema are either equated to or constrain concepts in the *give-cold* schema. A similar analysis follows for the target domain of infection.

- The source and target concepts of the additional associations must be closely connected to the source and target concepts specified by the core associations.
- The new source and target concepts must be connected to the core source and targets in a coherent manner.

Figure 7 shows the extension relationship between the core metaphor *have-cold* and the extended sense *give-cold*. In this diagram we see that the component metaphor maps from the core sense are shared and specialized by the extended sense. This section will consider exactly how this instance of an extended sense meets each of the given extension requirements. This example will be used to illustrate in general the mechanisms that are used to accomplish metaphor extension.

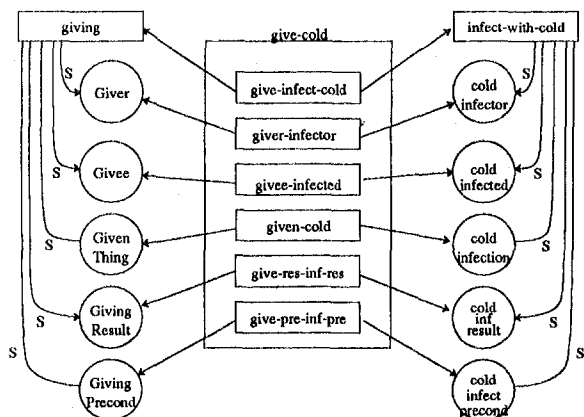


Figure 4: Give-A-Cold

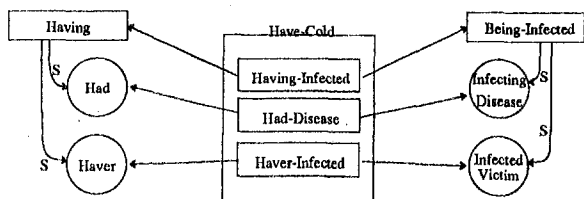


Figure 5: Have-A-Cold

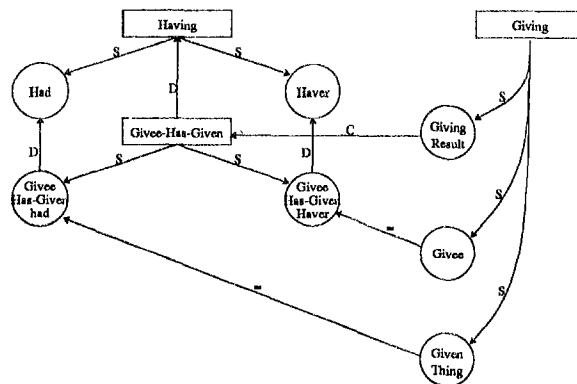


Figure 6: Giving and Having

A given metaphorical use must meet the following requirements to be considered an extended use of a more fundamental core metaphor.

- The extended sense must contain all the core metaphorical associations as a subset of its own associations.

The metaphor-map representing the fact that a haver plays the role of the infected, the *haver-infected* map, appears twice in the extended sense, once as the *giver-infectior* and again as the *givee-infected*. The giver of the giving maps to the infectior; this matches the core map since the giver is equated to be a kind of haver. This core map is also used to represent the fact that the givee is also this kind of haver as a result of the giving. The next map to consider from the core is the *having-infected* map. This map also appears twice in the extended sense. The first use is in the *give-res-inf-res* map. This map indicates that the result of the giving, a having by the givee, plays the role of the result of the infecting, an infection of the infected person. It is therefore a specialized map from a having to an infected state. It appears again in the

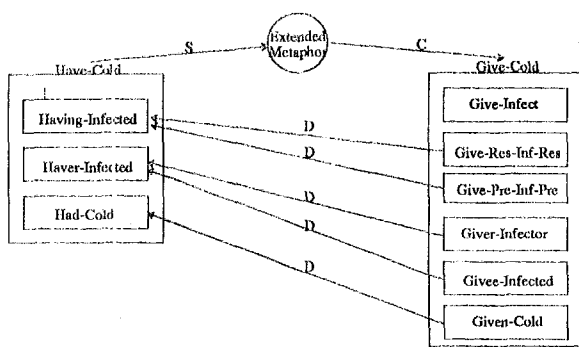


Figure 7: Having Extending to Giving

give-pre-inf-pre metaphor map. This map indicates that you have to have a cold before you can give it to anyone. This is again basically a map from a having to an infected state. The final map to consider is the core had-cold metaphor-map. This map appears once in the extended sense as the given-cold metaphor.

The links between the shared maps in the extended sense and the maps in the core sense are shown simply as individual D links. In fact the specialization of a metaphor-map is a more complex structured association involving the specialization of each of the components. The following section on metaphor hierarchies will go into the details of how this is actually accomplished. At this point it is sufficient to say that the first extension requirement is satisfied if all of the metaphor-maps in the core exist in the extension as specializations.

The second requirement for extension is that the source and target components of the extended sense must be closely connected to the concepts specified by the core. Figure 6 shows that this is clearly fulfilled in this case in the source domain. The concepts in the source domain of the extended sense are all either equated to, immediately dominated by or related by a single relation to the concepts in the source of the core domain. The same is true in the target domain. In general the second requirement will be satisfied if the extended concepts are related to the core by these types of close association.

The final requirement specifies that the extended concepts be consistent extensions from both the source and target domain of the core metaphor. This is met in this case by the result relations in both the source and target. On the source side the giving is related to the having from the core by a result relation. This is shown in the abbreviated slot form in Figure 6 by the giving-result slot. This relation is inherited from the more general category of action and indicates that a result of a giving is a having. The same analysis holds for the target domain. The result of an infecting is a state called being-infected. These states are shown as the target concepts in Figures 4 and 5. The last requirement is satisfied if the same relationship holds between the core and extended concepts in both the source and target domains.

### 4.3. Metaphorical Hierarchies

Metaphor-maps and metaphor-senses are both full-fledged KODIAK concepts and can therefore be arranged in abstraction hierarchies. The use of hierarchies will be the primary mechanism used to account for the similarities and differences among conventional metaphors. This use of abstraction hierarchies to capture regularities in the lexicon is similar to that described in Fass (1988). The following section will show how the various similarities and differences among conventional metaphors follow directly from the relations among metaphor-maps.

Consider again the kill-process map from (1). This is a mapping from a concept constrained to be a living thing to a

target concept that is not a living thing. This is a manifestation of a more general metaphor that allows us to view non-living things in terms of living things for the purpose of explaining or understanding their behavior in terms of living things. Examples (11) through (14) from Lakoff and Johnson (1980) all contain specialized instances of this general metaphor.

- (11) Inflation is eating up our savings.
- (12) Those ideas died a long time ago.
- (13) He is the father of modern quantum mechanics.
- (14) Those ideas won't yield any fruit.

Example (11) is motivated by the metaphor that the reduction in savings caused by inflation can be viewed as inflation consuming the savings. Inflation is therefore viewed as an animal that can consume things. Example (12) contains a metaphor dealing with the duration of ideas. When an idea is no longer held or believed it has died. At a more specific level we have an idea filling the role of the dier of a death-event as in Figure 3. This role has the similar constraint as the kill-victim of (1) that it be a living-thing. In a similar way as (1) a set of interrelated metaphor-maps link the source domain of death to the target domain of ideas and their duration. In particular there is a metaphor-map from the dier to a concept that is not alive.

Example (13) contains the notion that the creation of an idea is a birth-event and that the originator of the idea plays the role of the father in the birth event with the created idea playing role of the child. Once again in this metaphor there is a metaphor-map from a concept that is not a living thing (the created idea) to a role that must be one (the child being born). This metaphor-map, however, is more specific since the constraint is not only to be a living thing but to be human. Finally example (14) contains the notion that an idea can produce new ideas. This is metaphorically structured as a plant producing new fruit. In this case an idea is viewed as a specific kind of living thing, a plant.

What (1) and (11) through (14) all have in common is the idea that an abstract concept like a process or idea can be viewed as a living thing to explain some aspect of its nature. They differ in the particular kind of living-thing that is used and in the role that it plays. These similarities and differences result in specific metaphor-maps in each particular case. What is needed is a mechanism that can capture the commonalities and differences among these various metaphor-maps.

This mechanism is provided by the general inheritance mechanisms provided by KODIAK. Metaphor-maps that share properties are dominated by more abstract parent maps that capture the commonalities among the children. The source and target aspectuals of the parent map are constrained by concepts that are more abstract than and dominate the constrainters on the children's source and target aspectuals. Figure 8 illustrates this situation with the maps from examples from (1) and (11) through (14).

The top half of Figure 8 shows the hierarchical relationships among the maps underlying the above examples. They all converge on the abstract metaphor-map representing the idea of viewing a non-living-thing as a living-thing. The two metaphor-maps in the dotted box are expanded in the bottom half of the diagram to show the exact details of the inheritance links. In this expansion we can see that the idea-as-living-thing metaphor-map dominates the forgotten-idea-as-dier map. In the parent map the idea-as-target aspectual is constrained to be an idea. The living-thing-as-source aspectual is constrained to be a living-thing. In the forgotten-idea-as-dier we see that the inherited source aspectual is specialized by being constrained by the dier aspectual of the death-event. The inherited target aspectual is further specialized by being constrained by the

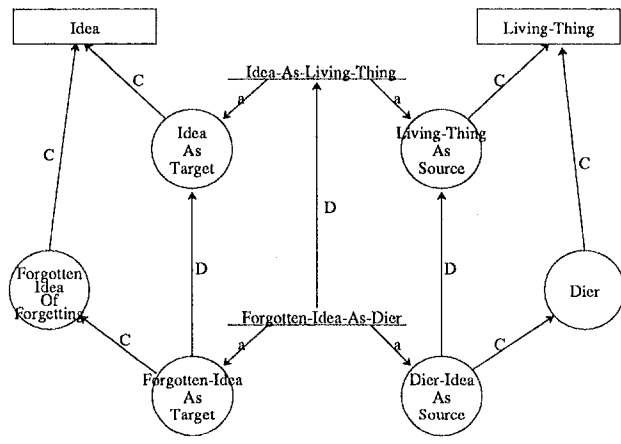
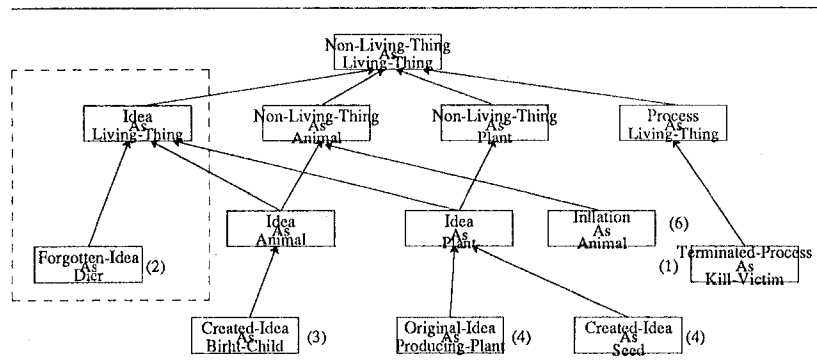


Figure 8: Metaphor-map Hierarchy

forgotten-idea aspectual. The constraining aspectuals, forgotten-idea and dier satisfy the appropriate inheritance restrictions since they in turn are constrained by idea and living-thing respectively. It is typically the case that the source and target aspectuals of metaphor-maps at the bottom of the metaphor hierarchy are constrained by aspectuals of some specific context. More abstract metaphors tend to have their aspectuals constrained by less specific absolutes.

Careful examination of Figure 8 reveals that the metaphor-maps are actually being specialized in different ways. Consider the process-as-living-thing specialization of the non-living-thing-as-living-thing. In this case the metaphor-map is specialized on the basis of the target concept. The generic category non-living-thing is replaced by process. In the case of the non-living-thing-as-plant mapping the specialization is done on the basis of the source concept. The metaphor-map hierarchy may therefore contain specializations based either the source or target aspectuals or both.

Metaphor-maps may also participate in multiple inheritance. Consider the idea-as-animal and idea-as-plant maps. These maps inherit their respective specializations from multiple parents. The details for the idea-as-animal map are shown in Figure 9. In this diagram we can see that this map has two parents the idea-as-living-thing map and the non-living-thing-as-animal map. Its source and target aspectuals are each dominated by the both of the corresponding aspectuals from each parent. The multiple-inheritance mechanism assigns only the most specific constrainer to the lower child aspectuals. In this case the source aspectual is constrained to be an animal, which is the more specific than living-thing. In the same way the target is constrained to be

an idea rather than the more abstract non-living-thing. In this way the idea-as-animal map is created by specializing different parts of the two parent maps.

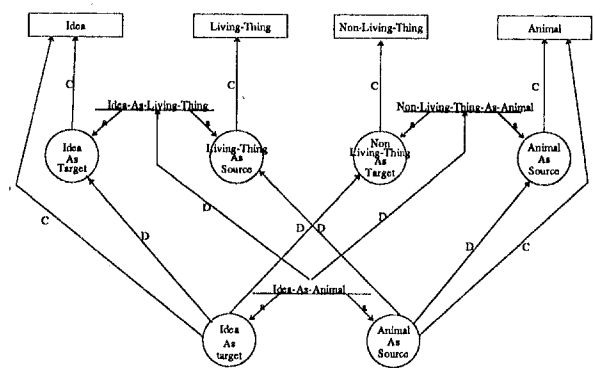


Figure 9: Multiple Inheritance

4.4. Representing Metaphorical Similarities

Consider the relationship between the metaphors underlying (15) and (16).

- (15) Mary gave John a good idea.
- (16) Mary gave John a cold.

These examples do not involve metaphors with overlapping scope nor are they related in a strictly hierarchical fashion. Nevertheless they do to have a structural similarity that should be reflected in the representation.

In particular note that both contain a map that represents

the notion that something that is not an object can be treated as one, (a disease and an idea.) Both contain the notion that transfer represents some change of state with respect to the metaphorical object. The point of these examples is that the metaphor schemas underlying examples (15) and (16) contain individual component maps that are similar. This similarity is accounted for by the hierarchical relationship among the component maps in each schema. Figure 10 shows some of the relevant relationships between these schemas that accounts for their similarity. (Note that for the sake of readability most of the maps in each schema are omitted from this figure.)

This figure illustrates the building block relationship between metaphor-maps and schemas. In the give-an-idea schema two maps are shown. The give-tell map is dominated by the communicate-as-transfer map which in turn is dominated by the act-upon-as-transfer (Jacobs 1985) map. The act-upon-as-transfer map is a map representing a general metaphor that allows abstract events with participants to be viewed as transfer-events where some participant receives or donates some metaphorical object. The given-idea map is dominated by the idea-as-object map which is in turn dominated by the non-object-as-object map. This metaphor map reflects the widespread use of reification in our conceptual system.

Similarly in the give-a-disease schema two maps are shown. The given-disease map is dominated by the disease-as-object map. This in turn is a sibling of the idea-as-object map since they both have the non-object-as-object mapping as a parent. Finally the give-infect map is dominated by the infect-as-transfer map which like the communicate-as-transfer map is dominated by the act-upon-as-transfer mapping.

Figure 10 illustrates the somewhat more distant relationship between the schemas underlying (15) and (16). They involve the same source domain with completely separate target domains. The domains of communication and infections are not related in any meaningful way in the hierarchy. They do not share any close meaningful parent categories. For this reason there can be no abstract metaphor schema dominating both of the schemas in question. However the similarity of the two schemas is accounted for by the hierarchical structure of the component maps and the sharing of maps with common parents.

### 5. Summary

An adequate representation of knowledge about the con-

ventional metaphors in a language must meet two requirements. It must capture the exact details of individual metaphors to permit efficient recognition and interpretation. In addition it should reflect the regularities of the system of conventional metaphors as a whole. In particular the extension of core metaphors to extended senses should arise naturally from the structure of the metaphor and the structure of the source and target domains. This paper has described a system of representing conventional metaphors that efficiently meets these requirements.

### References

Brachman, J. & Schmolze, J. G. (1985) An overview of the KL-ONE representation system. *Cognitive Science*, 9, 171-216.

Fass, D. (1988). *Collative Semantics: A semantics for natural language processing.*, Report No. MCCS-88-118, Computing Research Laboratory, New Mexico State University, Las Cruces.

Jacobs, P. (1985). *A knowledge-based approach to language production.* Doctoral Dissertation, Computer Science Division Report No. UCB/CSD 86/254, University of California, Berkeley, CA.

Jacobs, P. (1987). A knowledge framework for natural language analysis. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence.* Milan, Italy: Morgan Kaufmann.

Lakoff, G. & Johnson, M. (1980). *Metaphors We Live By.* Chicago, IL: University of Chicago Press.

Martin, J. (1987). The acquisition of polysemy. *Proceedings of the Fourth International Workshop on Machine Learning.* Irvine, CA: Morgan Kaufmann

Martin, J. (1987). Understanding new metaphors. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence.* Milan, Italy: Morgan Kaufmann.

Norvig, P. (1986). *A unified theory of inference for text understanding.* Doctoral Dissertation, Computer Science Division Report No. UCB/CSD 87/339, University of California, Berkeley, CA.

Wilensky, R. (1986). *Some problems and proposals for knowledge representation.* Technical Report No. UCB/CSD 86/294. Berkeley, CA: University of California, Berkeley, Computer Science Division.

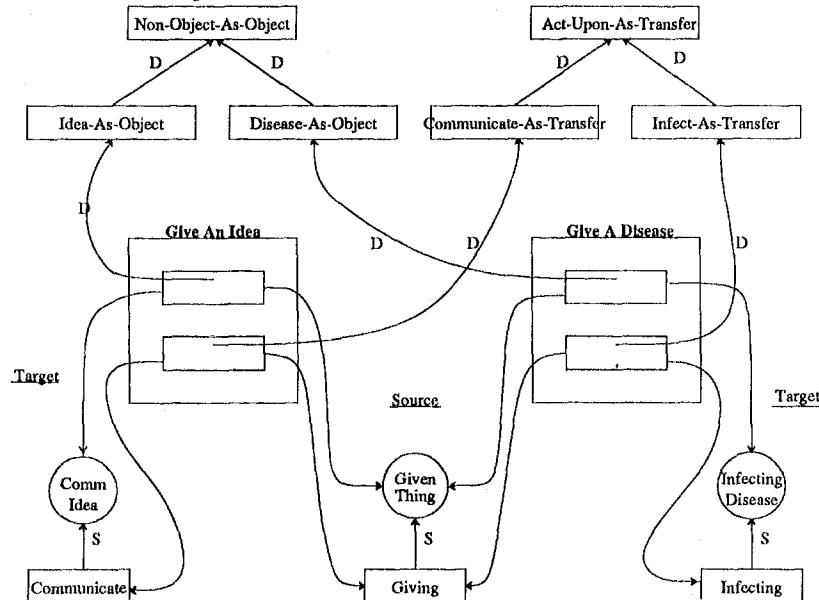


Figure 10: Giving Metaphors