A Computational Model of Complex Concept Composition

by

Carl Floyd Andersen, B.A.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Arts

The University of Texas at Austin

August 1996

A Computational Model of Complex Concept Composition

Approved by
Supervising Committee:

To my parents

Acknowledgments

I am extremely grateful to Bruce Porter and Peter Clark for their intellectual contributions, guidance, and patient helpfulness.

Carl Floyd Andersen

The University of Texas at Austin
August 1996

A Computational Model of Complex Concept Composition

Carl Floyd Andersen, M.A.
The University of Texas at Austin, 1996

Supervisor: Bruce W. Porter

Complex concept composition is the joining of information from two existing concepts to form a third, more complex concept. Accounts of how concepts are composed are of great theoretical and practical importance. Such compositions are a part of most human cognitive functions, and are therefore of great interest to cognitive scientists. Knowledge engineers study composition for a different reason: to discover efficient techniques for building conceptual knowledge bases. This thesis presents a composition mechanism that draws upon psychological views of concepts as theories, sets of interrelated facts. Such a mechanism more accurately models the interaction between two concepts' features in the composition process and explains psychological data about concept composition which past models cannot account for. The mechanism also serves as a useful framework for efforts to build modular, reuseable concepts.

Contents

List of Figures

Chapter 1

Introduction

Complex concept composition is the joining of information from two existing concepts to form a third, more complex concept. Accounts of how concepts are composed are of great theoretical and practical importance. Such compositions are a part of most human cognitive functions, and are therefore of great interest to cognitive scientists. In constrast, knowledge engineers study composition to discover efficient techniques for building conceptual knowledge bases. This thesis presents a composition mechanism that draws upon views of concepts as theories, sets of interrelated facts. Such a mechanism more accurately models the interaction between two concepts' features in the composition process and explains psychological data about concept composition which past models cannot account for. The mechanism also serves as a useful framework for efforts to build modular, reusable concepts.

Chapter 2

Motivation

Current models of complex concept formation have been influenced by psychological experiments, linguistic evidence, philosophy, and artificial intelligence representation techniques. Our understanding of the issue has evolved over the past thirty-five years as successive theories of concepts have clashed and become more prominent. The issue is a central one for both cognitive scientists' and knowledge engineers' efforts.

2.1 Cognitive Science

For cognitive scientists, the mechanism by which complex concepts are formed has implications for a host of related issues. The most central of these is the characterization of conceptual structure: any account of how concepts are structured or defined must ultimately have the resources to explain how new concepts can be created in the mind. Presenting a concrete composition algorithm that explains some of the wealth of intuitional and empirical data about concept formation would provide support for related accounts of conceptual structure.

Concept composition also has implications for various cognitive processes that seem to involve composition or be similar to it. Such processes include analogy and metaphor, language comprehension, and problem-solving [21]. Each process relies upon a mechanism for concept composition, and is therefore potentially constrained by such a mechanism's explanatory richness.

This thesis presents an algorithm for concept composition that is based on ideas of conceptual structure proposed by Murphy and Medin [21] , who claim that concepts are at least partially composed of theory-like sets of inter-related facts. I hope to show that the algorithm better explains our basic intuitions about new concepts and how they are formed. I will also demonstrate that the algorithm can explain psychological data about concept composition summarized in Murphy and Medin's work. Finally, I will show that the algorithm supports specific research into goal-directed typicality performed by Lawrence Barsalou [5, 8]. My algorithm will serve to concretize each of these reseachers' models of concept structure and composition, bolstering their associated claims.

2.2 Knowledge Engineering

Because of the difficulty of modeling and entering knowledge representations (the "knowledge engineering bottleneck"), traditional construction of expert systems has focused on the task at hand. Historically, knowledge engineers created highly specific KB's suitable for narrowly directed goals, such as diagnosing a small subset of liver ailments. The knowledge contained in these KB's was typically not reusable in new systems because of its specificity, even though a fair proportion of it deals with potentially general knowledge such as causal processes, probabilities, spatial properties, etc.

A related problem with knowledge representations of the past involves the sharability of knowledge. Representations are dependent upon the assumptions of knowledge engineers about the world. Such assumptions often conflict with those of a different knowledge base if knowledge is transferred wholesale from the first, requiring detailed reconciliation of conflicting representations. In an ideal world, representations centered around reusable, composable concepts might provide a standard core of knowledge. Concepts built upon such KB "kernels" would be more easily shared and translated. Representations of completely different problem domains would be compatible if constructed upon such a common foundation.

A final issue encountered in past efforts at knowledge engineering is redundancy. Because knowledge engineers built individual representations without a systematic process of knowledge reuse, representations of distinct yet related phenomena often contained redundancies: the same knowledge modeled twice. At best, such parallel efforts waste time; at worst they introduce contradictions into the KB.

Recent years have witnessed an emphasis on making more general-purpose knowledge bases whose representations are useful over multiple problem domains [14]. Part of this effort involves trying to make concept representations more modular in an attempt to make reuse of their components possible. However, any plan for modularity requires a method of efficiently and accurately composing concept "modules". Such a method must first be general-purpose, able to create any needed concept from the appropriate components. Second, to fully realize the power of modularity, the method must be relatively automatic and accurate, freeing the knowledge engineer from the manual "fleshing-out" and "cleaning-up" activities that dominate systems development. I hope to demonstrate that my algorithm can contribute to such efforts: it helps automate the process of building up complex concepts from basic concepts.

The motivation for providing a more concrete algorithm or mechanism for concept composition is therefore twofold. Such an algorithm can provide support and detail for current psychological claims about the structure of peoples' concepts. Such a mechanism also advances knowledge engineers' efforts to create general-purpose knowledge bases.

Chapter 3

Related Work: Views of Concepts in Cognitive Science

The issue of what constitutes a concept has occupied philosophers for centuries, but in the past thirty-five years work by psychologists, linguists, and computer scientists has brought renewed debate and perhaps, more subtle understanding. As reviewed in numerous papers [12, 3], the classic theory of concepts, derived from Aristotle, views concepts as a set, or collection of entities. Entities are members of the concept if they possess certain necessary and sufficient definitional features. Such a view was progressively challenged by psychological and linguistic research showing that not all features of concepts were defining (neither necessary nor sufficient): there are birds, for example, that cannot fly. Psychological experiments about peoples' judgements of the typicality of concept members were also damaging. Such experiments show no abrupt typicality threshold (and therefore, no clear definitional boundary) between nonmembers and members of a concept. Instead, the results showed a gradual increase in typicality proportional to the resemblance of an instance to the concept definition.

Eventually, views of concepts with more flexible mechanisms determining category membership emerged. One such approach grew out of Zadeh's [27] theories of probabilistic reasoning. Zadeh defined concepts in terms of definitional features, but associated such features with instances probabilistically: a car might have a .9 chance of having a steering wheel, for example. Such an approach successfully addressed the earlier attacks against the classical view, but introduced its own problems. Perhaps the most serious was its inability to provide any standard of coherence for concept instances: an instance of a probabilistically-defined vehicle might contain both an engine, handlebars and a steering wheel [3].

Other accounts of concept structure proposed that concepts were organized around sets of examples of the concept, or prototypes [12]. Prototypes are synthesized from every instance of the concept encountered. In such theories, entities are members of a concept to a partial degree, inasmuch as they conform to a prototype of the concept. Such accounts offer effective explanations not only of typicality, but also of concepts' apparent function of grouping correlated features.

3.1 Murphy and Medin: Concepts as Theories

In recent years Murphy and Medin [21] emphasized the role of theoretical constructs in conceptual coherence. The two psychologists argued that concept definitions are at least partially characterized as theories: sets of interrelated structural and functional explanations or facts. Some of these mental explanations are internal to the concept, relating only concept elements together; others relate concept elements to the larger knowledge base. A theory-based concept definition is different from a concept definition based upon a list of features: the facts making up individual theories are more complex and they are often about the same components. This complexity and interconnectedness lead to concept definitions emphasizing internal relational structure. For example, the concept of an automobile is not simply a feature list, but a structure of related facts about the automobile components and how they relate to each other and the world: the car's tires touch the road, the car's engine makes the car's tires turn, fuel in the tank is consumed by the engine, etc.

Murphy and Medin argued against the primacy of any approach to concept definition based solely upon featural similarity, such as the classical view or even a prototypical stance. Any similarity-based concept definition gauges concept membership by the degree of similarity of the concept instance to the concept; such similarity is defined by the number of concept features possessed by the instance.

Murphy and Medin gave several convincing arguments against similarity-based approaches. They argued, first, that "similarity relationships are not sufficiently constraining to determine which concepts will be coherent or meaningful". Similarity-based views must explain why a concept came to be specified as a set of particular features. The world is full of objects with varying similar features, so how does a particular set of features assume the status of a concept? For example, zebra and a horse are seen as more similar than a zebra and a barber pole, though both pairs contain striking similarities. Furthermore, since psychological experiments demonstrate that people judge similarity by different features in different contexts, there seems to be no principled method of defining a concept by similarity alone.

Another argument questioned similarity-based theorists' ability to effectively model the world with concept definitions based on simple attribute (feature) matching. Murphy and Medin argued that attribute matching is too limited a process to capture the richness of concepts, which may contain elaborate information about possible transformations and operations not easily expressed as simple attributes. Concepts may also contain higher-order relations between attributes. Finally, concepts usually contain information about human concerns and goals and the concept's explanatory relevance to them.

Murphy and Medin advocated theories not as a replacement for similarity, but rather as a necessary addition to it. Theories, they claimed, serve as the essential "glue" that holds concepts together, and operate together with mechanisms using simple features to accomplish most cognitive functions, especially instance classification.

A view of concepts as theories conforms to data from other psychological studies, as we shall see, as well as with new, modular approaches to knowledge engineering. For the knowledge engineer, a concept modeled as a simple set of attributes is useless. To be truly modular, a definition must also express the internal structure of the concept itself: its function and physical structure and how they interrelate. An operational concept definition must contain information about the concept's relevance and possible uses.

One significant element missing from Murphy and Medin's ideas is also essential to knowledge engineers: an explanation of how concepts qua theories can combine to form new concepts. To address this problem, one must move beyond older mechanisms geared to outmoded accounts of concept structure.

3.2 Barsalou: Goal-Derived Concepts

Barsalou's recent work [4, 5, 6] on category typicality uncovered further evidence for complex, theory-like category structure. Barsalou compared typicality effects for common taxonomic concepts (e.g. *bird, plane, clothing*) with those for ad-hoc concepts derived to achieve specific goals (e.g. *foods not to eat on a diet, vacation locations*). As in the experiments described earlier in this thesis, Barsalou's typicality results are evidence for particular underlying structures in their associated categories.

Barsalou found that both types of concepts had strong typicality structures, but that the determinants of such structures were different. For common concepts, typicality was primarily caused by similarity to prototypical features. Though such similarity also was a factor for goal-derived categories, their typicality was more strongly determined by similarity to relevant ideals, properties conducive to fulfilling the associated goal. For example, a determinant of the category *foods not to eat on a diet* is the ideal *zero calories*. Such results clearly demonstrate that many goal-derived categories are not entirely structured around simple sets of prototypical features synthesized from concept examples. Instead, they are constructed on the fly from subordinate concept structures representing goals and ways of achieving them. The composition of such subordinate concepts forms the ideal properties that determine goal-derived category typicality. Interestingly, Barsalou discovered that even common taxonomic categories' typicality is partially determined by ideals: for example, typical *fruits* satisfy the ideal *tastes good.* This indicates that even the most common concepts have a degree of structural complexity involving goals.

Barsalou's second important discovery was that typicality for goal-derived categories varied over different contexts: exemplars were held to be more typical in one contextual use of the category than in another. Results showed that such contexts controlled the degree to which ideals determine typicality. Barsalou interpreted this result as evidence for a more flexible, multi-dimensional model of category structure, both for common and goal-derived categories. In such a model, only portions of a category's structure are activated and used in a particular context.

Barsalou used such results to make further, rather sweeping claims about the roles of common and goal-derived categories in planning tasks. In doing so, he adopted representational ideas (e.g. frames, constraints) of AI researchers , and subscribed to many of Murphy and Medin's ideas about concepts as theories. Barsalou claimed that categories can be modelled as frames, structured representations having attributes and constraints that can be linked together. In the planning process, a frame representing a common taxonomic category (e.g. *vacation* would be activated and its attributes progressively restricted, or instantiated, by one or more goal-derived concepts. For example, the *location* attribute of the vacation might be filled by *maximally beautiful scenery.* This restriction process continues until the structure is sufficiently detailed that specific examples adhering to existing constraints can be chosen (e.g. *The Bahamas*).

Barsalou claimed that the instantiation process is guided by constraint and optimization information built into goal-derived category structure. Such information restricted the possible fillers of attributes. For example, once an *activity* attribute of *snow-skiing* was instantiated, *tropical city* could not be selected to fill the *location* attribute. Barsalou explains contextualization as a product of this process. He claims that prior optimizations and constraints form a contextual structure which combines with possible attribute concepts, thereby creating complex, goal-derived concepts.

Barsalou's model of planning, his accounts of category structure, and his explanations of the differing roles of common and goal-derived categories are compelling, though admittedly speculative. He sketches a model of category composition, emphasizing the role of constraints and other relations between categories. However, his model gives no concrete mechanism by which concepts can be combined; it fails to explain how concepts can exert complex constraints, optimizations, and relations upon one another. I hope to provide such a mechanism and thereby remove one serious objection to Barsalou's various claims.

Chapter 4

Related Work: Knowledge Transfer Mechanisms

The theories of conceptual structure presented above evolved in the past thirty-five years in order to address psychological evidence and intuitive objections. Throughout this period, different representational approaches to modeling both concepts and knowledge transfer between concepts (of which concept composition is but one form) evolved in response. These approaches built upon the concept theories themselves as well as upon representational schemes of increasing sophistication. Accordingly, these methods have increasingly operationalized notions of modularity and transfer of relational structure.

4.1 Sets of Rules

Since the genesis of modern artificial intelligence research, concepts have been modeled using relatively primitive representations such as simple sets of logical rules or facts. Complex concept composition using rules generally involves a method of transferring rules about the two parent concepts to the composed concept. Such an approach suffers from the same problems that generally plague rule systems. Inheritance of properties is not as efficiently implemented as in frame-based systems, so transferring any properties of the modifier concept to the base concept requires great computation. Rules are not naturally encapsulated into distinct concepts as are later semantic nets and frames, so inheritance information cannot be stored modularly. Furthermore, rule systems providing the expressive power of full first-order logic without some way of restricting inference are potentially intractable. Therefore, a rule-based concept composition process must somehow circumscribe any inference performed in a composition process. [19] A final problem concerns knowledge engineering: rules systems are often much more difficult to understand and work with than are semantic nets and frames.

4.2 Frames and Inheritance

A more tractable and intuitive model of concept composition involves frame representations and their associated inheritance methods. Frame-based models are an effective represention for feature-based concepts, and the feature inheritance mechanism between frames serves as a corresponding model of concept composition.

In frame systems, a single frame represents a concept; multiple frames are linked together via relations to form larger models of the world. In its most simple form, concept composition is represented by creating a new frame that is a specialization of two or more prior concepts [9]. For example, one could create the frame *house-boat*, giving it specialization links to two prior concepts *house* and *boat*. Because specialization links entail the inheritance of properties, *house-boat* gains all the properties of *house* and *boat*, such as *inhabited by humans* and *floats on water*. However, such an account loses coherence when describing even many simple compositions. For example, a *dog-house* may possess some properties of a *house*, but certainly not those of a *dog* : it does not have fur or a tail.

More complex accounts of composition involve the joining of only two basic concepts (frames): a base concept and modifer. As described by Murphy [22], composition essentially is a process of specialization of the base concept. The modifier frame is inserted into a particular slot in the base concept's frame; this slot corresponds to a "mediating relation" that specifies how the two old concepts are related in the new composed concept. The resulting composition is therefore a specialization of the base concept, a specialization having the additional property that its value of SLOT is MODIFIER. For example, the concepts *dog* and *apartment* can be combined by filling the *habitat* slot of *dog* with *apartment*. The mediating relation is therefore *habitat*, as "the dog's habitat is an apartment". The resulting complex concept *apartment-dog* is therefore a specialization of *dog* with slot *habitat* = *apartment*.

A composition model based upon inheritance addresses some of the problems of rule-based systems: it deals with inheritance (and therefore composition) efficiently and modularly, it achieves composition with very little inference, and it provides an intuitive model for knowledge engineers. However, inheritance fails to address a range of problems that parallel those of similarity-based accounts in general. The most important concern is that there seems to be no way that the features of two composing concepts interrelate or modify each other, though such a mechanism is clearly at work in many real-life compositions. An adequate conception of *apartment-dog*, for example, involves more than a restriction of habitat. An *apartment-dog* is fed certain food, probably gets less exercise than its non-urban counterparts, and probably is physically smaller. All these features are derived from interactions between the original features of *dog* and *apartment*.

A related problem concerns the resulting concept's coherence: can any two concepts be composed using any slot, or are there cognitive restrictions of the kinds of concepts that can be formed? More generally, it is not clear how an inheritance mechanism might deal with conflicting properties of the two parents.

Several theorists (see [12]) have emphasized the role in the composition process of world-knowledge, knowledge in the knowledge base other than that embodied by the two concept frames. These theorists persuasively argue that world knowledge plays a role in resolving each of the problems raised above. For example, people use contextual information to deduce the proper mediating relation for ambiguous concepts such as *finger cup*. Although it is apparent that world knowledge is essential to successful composition, an inheritance-based model of concept composition provides no easy hooks for world knowledge mechanisms. Because inheritance is based upon a similarity-based notion of concepts, it cannot model complex interactions between concept features and outside knowledge. Clearly, adequate accounts of features' interaction with both world knowledge and each other require more elaborate models of concepts and composition.

4.3 Analogy

Analogical reasoning (see [15, 17]) is another method for transfer of knowledge about one concept or schema to another. Analogy involves transferring an entire set of relations relating one group of objects to another, so that the second group then has the same relations between them. Probably the canonical example in the literature of analogical reasoning is the analogical transfer of our knowledge about the solar system to the atom ("The atom is like a solar system"). In the example, physical relations between heavenly bodies such as "the planets orbit the sun" are transferred to the atom ("electrons orbit the nucleus").

Analogy is noteworthy as a transfer method for two reasons. One is that the transferred knowledge is always in the form of relations between entities, not the entities themselves or their attributes. Thus, analogy, when applied to a concept, does not introduce any new components to the concept; it only further relates those components already in the concept. This mechanism cannot fully model concept composition: many concepts, like *house-boat*, inherit features of both their parents, not just a relational structure. However, such relational-structure transfer is potentially applicable to the composition of Murphy and Medin's structure-emphasizing concepts.

Research on analogy is also relevant because researchers have made initial attempts to explain how the correct knowledge transfer between an object and its analogue is achieved. Some writers present the problem as one of properly matching the structures of two systems of relations. Since complex concept composition seems to involve linking similar components of parent concepts (e.g. *habitat* and *apartment*), such techniques might be applied to find the proper composition of two concepts.

4.4 Compositional Modeling

Compositional modeling is yet another method of building up complex concepts; it involves techniques of building formal logical models of physical systems via modular, composable model "fragments" [13]. A fragment models only a portion of the many relations, operations, or transformations that describe a particular physical phenomenon. For example, one fragment might model the voltage of a battery under certain assumptions. Limiting fragments' scope to modular, discrete phenomena promotes their composability, making reuse and large-scale system modeling possible. Furthermore, the explicit packaging of each fragment's modeling assumptions with it makes feasible the construction of minimally complex models under varying assumptions and levels of detail [20].

Because of their emphasis on reuse and formal notions of relevance and simplicity, compositional modeling techniques are potentially quite useful to knowledge engineers. However, such methods have only been applied to physical systems; it remains to be seen whether domains historically described less formally can be easily broken up into discrete fragments. Furthermore, though the modularity of model fragments makes efficient knowledge base construction possible, techniques for efficient construction of low-level fragments have not been aggressively explored.

Chapter 5

A Model of Concept Composition

Each of the preceding models of concept composition is insufficiently general. Some models (sets of rules and analogy) conflict with our intuitions about composition, while some (inheritance) cannot model important examples. Compositional modeling, though potentially general-purpose, appears to rely too completely on concepts' physical structures for composing them correctly. In this section, I describe a new concept composition algorithm based upon Murphy and Medin's views of concepts as theories. I hope to demonstrate that my algorithm addresses limitations of each of the preceding composition models.

5.1 Representational Assumptions

This algorithm uses representational assumptions found in recent theoretical accounts of concept structure. First, a conceptual graph representation similar to that found in Sowa [25] is used to represent concept definitions. This scheme can represent first-order and more complex logics; it also supports important features such as embedded units, default inheritance, value restrictions, token-type distinc-

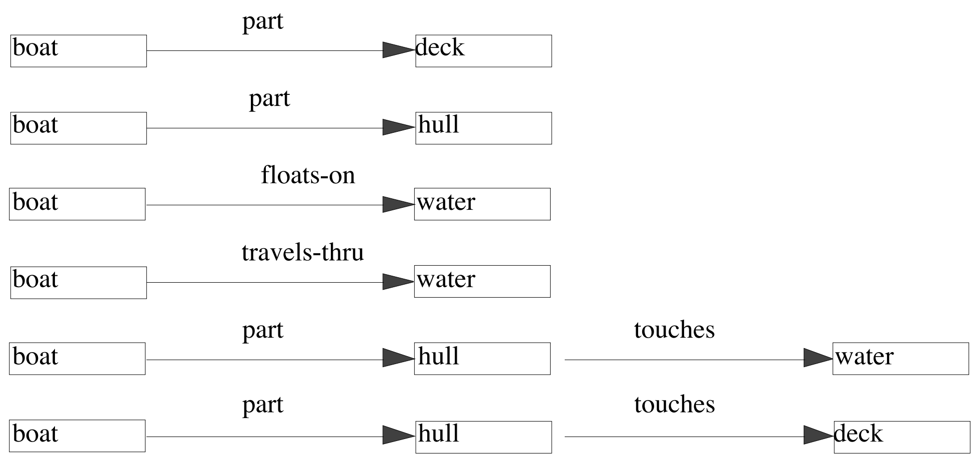


Figure 5.1: The individual concept facts defining *boat*.

tions, etc. The representations themselves model concepts as collections of distinct yet related facts, where a fact is defined as a collection of component concepts linked to the original concept by a set of relations (Figs. 5.1 and 5.3). Such concept representations have many similarities to geometric graphs if we view concepts as vertices or nodes and relations between them as edges. Graph-like concept representations are generally reasoned with by performing a unification process that combines all concept definition facts into a single graph, or set of concept nodes and linking relations (Figs. 5.2 and 5.4).

Such a representation can be notationally confusing, as one can write of a "concept" and refer either to the node labeled with the concept name or to the larger conceptual graph. Therefore, when I use the term "concept" in the context of conceptual graphs, I refer to a graph containing facts that define the concept. When I refer to the term "main concept node", I refer to the node in such a graph bearing the name of the concept. Finally, when I use the term "component" or "concept node", I refer to some individual concept node in such a graph.

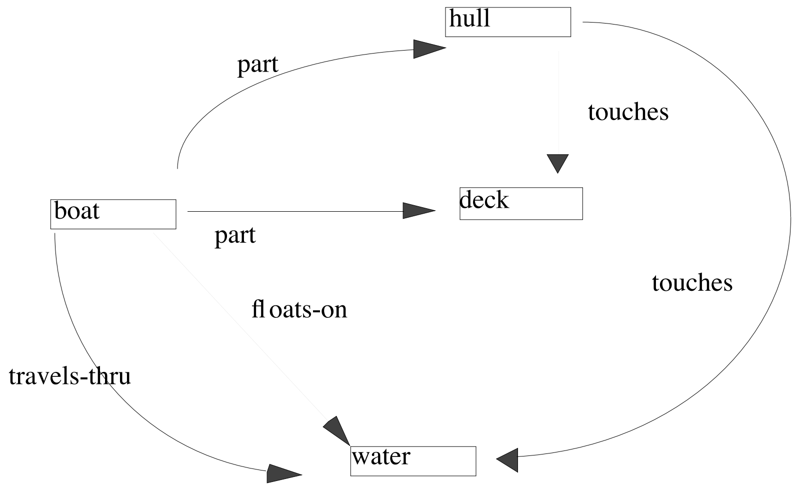


Figure 5.2: Individual facts can be unified to form one graph.

5.2 An Overview

The process of complex concept formation consists of the creation of a new concept definition from the information contained in two initial concept definitions. The resulting definition may itself be used in conjunction with another concept to form a new, even more complex concept; in this way, concepts of arbitrary complexity can be composed.

My description of concept composition is role-based: it mirrors our intuitive notions of one entity acting "in the role" of another. A house boat, for example, can be viewed as a case of a house acting in the role of a boat. This intuitive notion is operationalized by transferring all properties of *boat* to *house*. The resulting entity has properties of both concepts: it is a *house* that is also a *boat*. Not all compositions involve the concept itself acting in a role; sometimes a component of the concept does so. For example, in the concept *boathouse* (a place to store boats), the occupant or contents of the concept *house* act in the role of a *boat*.

Such a role-based model of concept composition starts with two concepts, a Base and a Modifier, as well as a hint about how the two concepts relate to each other. Each concept is defined by a set of distinct facts represented via frames. The composition algorithm first uses a graph unification process [25] to combine each set of facts into single graphs. Facts from the Modifier concept are transferred to the Base concept by mapping all nodes and relations from the Modifier graph onto the Base graph. Finally, the resulting Base graph is deconstructed back into individual facts, which become the facts defining the new concept. For instance, the concept *boat* can be used to modify the concept *house* by transferring facts about *boat* to *house*. The resulting composition of the two concepts is a new concept, *house-boat* (a house that is also a boat), which is distinct from either of the two initial concepts and shares properties of each.

5.3 The Graph Construction Phase: A Construction Algorithm

The first part of the concept composition algorithm therefore begins with the two sets of facts defining the Modifer and Base concepts; each fact is represented as a graph. The algorithm converts each set of facts into a unified graph via the use of conceptual join operations [25]. Join operations are graph versions of logical unification: two graphs, each representing a logical sentence, are combined by finding appropriate variable substitutions. The construction algorithm begins with one fact and progressively joins the remaining facts with it (Figs. 5.1 and 5.3). Provided that the concept facts are non-contradictory, such an algorithm will return a single unified graph. The output for this stage is therefore two graphs corresponding to the Modifier and Base concept definitions (Figs. 5.2 and 5.4).

5.4 The Graph Mapping Phase

The second phase of concept composition uses a Graph Mapping algorithm in which the Modifier graph is mapped onto the Base graph. This algorithm begins by using

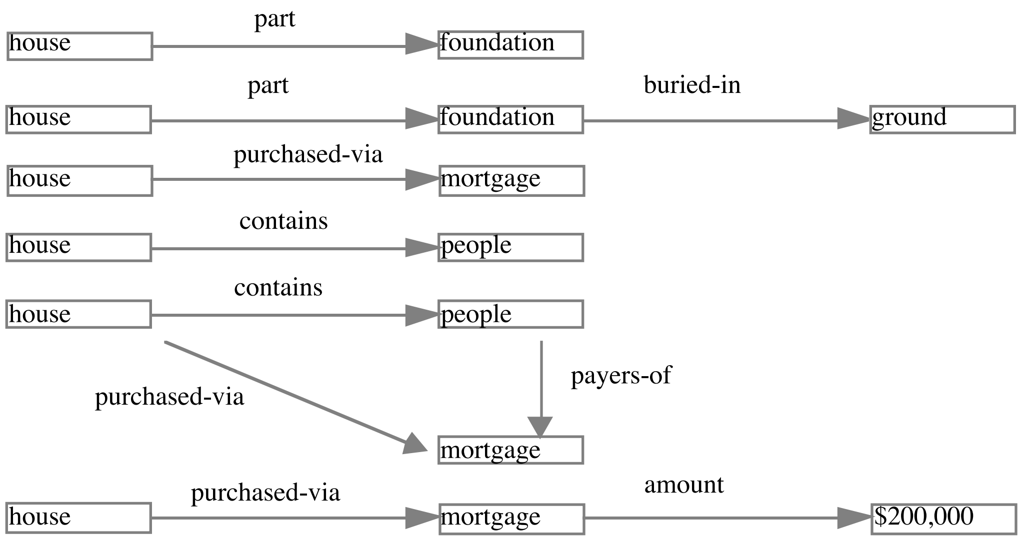


Figure 5.3: The individual facts defining *house*.

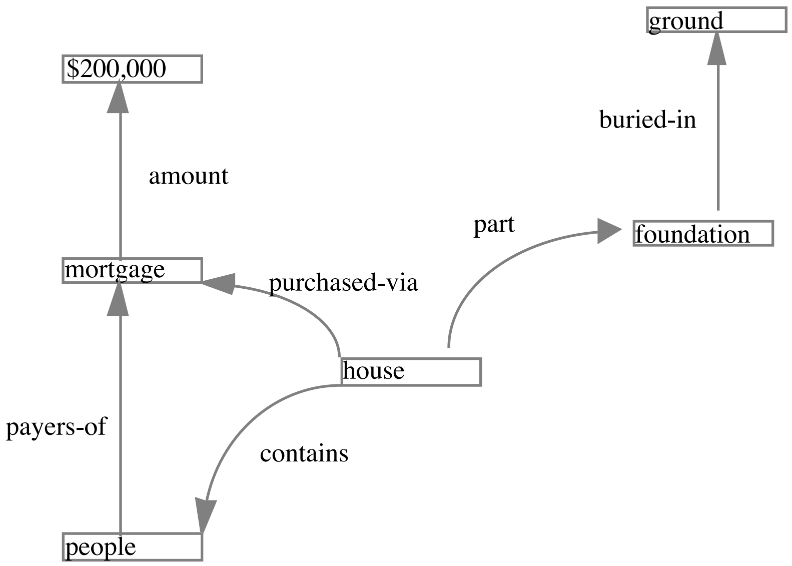


Figure 5.4: The Graph Construction Phase of the algorithm unifies the individual facts to form one graph.

a given Mediating Relation to determine where the Modifier and Base graphs are to be initially matched.

5.5 Establishing a Mediating Relation Using Outside Knowledge

Cognitive scientists have long argued that a "mediating relation" is necessary to connect the two concepts so that information may be properly transferred. For example, in the example of *ocean drive* the relation might be that the drive *is-beside* the ocean. Various authors also argue that world knowledge, knowledge not contained in either the modifer or base concepts, is necessary for the specification of the proper mediating relation. Such an account is proven by examples such as *finger cup*, in which the two concepts could conceivably relate in many different ways. Our knowledge of the world and of the context *finger cup*, is encountered in enable us to specify the proper relation. However, there has been little in the way of details of how such a specification might be achieved. This question is beyond the scope of this model, so we assume that a mediating relation is predetermined.

5.6 A Mapping Algorithm

The graph mapping algorithm begins with Modifier and Base graphs and a given mediating relation between them (Fig. 5.5). The mediating relation between the modifier and base concepts enables at least one mapping, or match, to be made between corresponding Modifier/Base components. There are two possible cases:

1. The mediating relation is *isa* (e.g. a houseboat is a boat that *isa* house). In this case, the Base concept itself fills the role of the Modifier (as opposed to a subcomponent of the Base doing so). Therefore, a match is made between the main concept nodes of Modifer and Base (Fig. 5.5, part *A*).

2. The mediating relation is some other relation X (e.g. *contains, part, function*). Then we say that the Base concept is in relation X to the Modifier. In such a case, the Base concept will already contain relation X, which links the main concept node and some component node N. We then say that N fills the role of the Modifier and therefore match the main concept node of Modifier with node N. (Fig. 5.5, part *B*)

A more complete mapping between the two concepts can be achieved by spreading the mapping outward from the first pair of nodes. To do this, the algorithm traverses the Modifier graph, attempting to match each successive node with a counterpart in the Base graph.

5.7 Mappings Between Components

Mappings, or matches are established between corresponding nodes in the modifier and base concepts to reflect that when a Base concept acts in the role of the Modifier, components of the Base can be said to act in the role of counterparts in the Modifier. By making this correspondence explicit, we can transfer properties from modifier components to base components.

New matches are found by beginning with a pair of matched Modifier-Base nodes M and B. Each node in the pair may be linked to other nodes via relations; if M is linked to a node M' by relation R and B is linked to a node B' by the same relation R, then M' and B' are candidates for matching.

The algorithm attempts to match them by checking their similarity (Fig. 5.5, part *C*). If the pair are sufficiently similar, the match succeeds; the node B' is deemed able to fill the role of the node M' and therefore remains as part of the final concept.

```
GRAPH MAPPING ALGORITHM AND VARIABLES
Note: X.Y is the object for which X bears relation Y

GLOBAL VARS
Modifier is the unified modifier concept definition, a graph of nodes and relations
Base is the unified base concept definition, a graph of nodes and relations
M is the modifier concept node [e.g. the node boat]
B is the base concept node [e.g. the node house]

LOCAL VARS
MNode is the current node from Modifier being examined
BNode is the current node from Base being examined
RM is the mediating relation between Modifier and Base
R is the current relation from Modifier being examined

Compose_Concepts(Modifier,Base, RM)
    MNode := M
    IF (RM = isa)
        BNode := B                                                  *A*
    ELSE
        BNode := B.RM                                               *B*
    Apply_Role(MNode,BNode)

Apply_Role (MNode,BNode)
    mark MNode
    IF Bnode = B OR Similar(MNode,BNode)                            *C*
        add <MNode,BNode> to matched nodes list                     *D*
    ELSE
        destroy all unmarked relations in Base leading from BNode   *E*
        replace BNode with a copy of MNode
    WHILE (there are unmarked relations leading from MNode)
        choose such a relation R and mark it
        IF (MNode.R is unmarked AND BNode.R exists)                 *F*
            Apply_Role(MNode.R, BNode.R)
        ELSE
            copy relation R to BNode                                *G*
            Apply_Role(MNode.R,MNode.R)
```

Figure 5.5: A graph mapping algorithm.

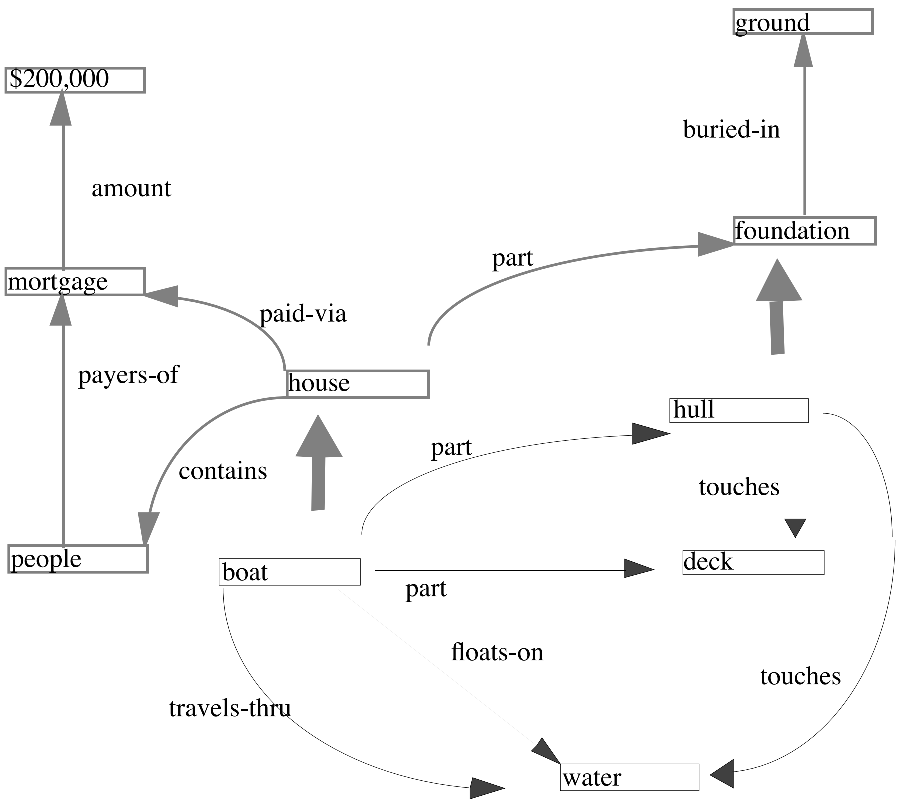


Figure 5.6: In the Graph Mapping Phase of the algorithm, the Modifier concept is overlaid upon the Base concept graph via matches made between nodes in each graph. Here, the *boat* concept acts as a modifier to *house*- this forms the complex concept *house-boat*, a house that is also a boat. In this example, the only match is between *boat* and *house*. The algorithm attempts to match *hull* and *foundation*, but *foundation* is too dissimilar (by virtue of information not shown in this graph, such as being rooted in the ground). So, *foundation* is overridden by *hull*

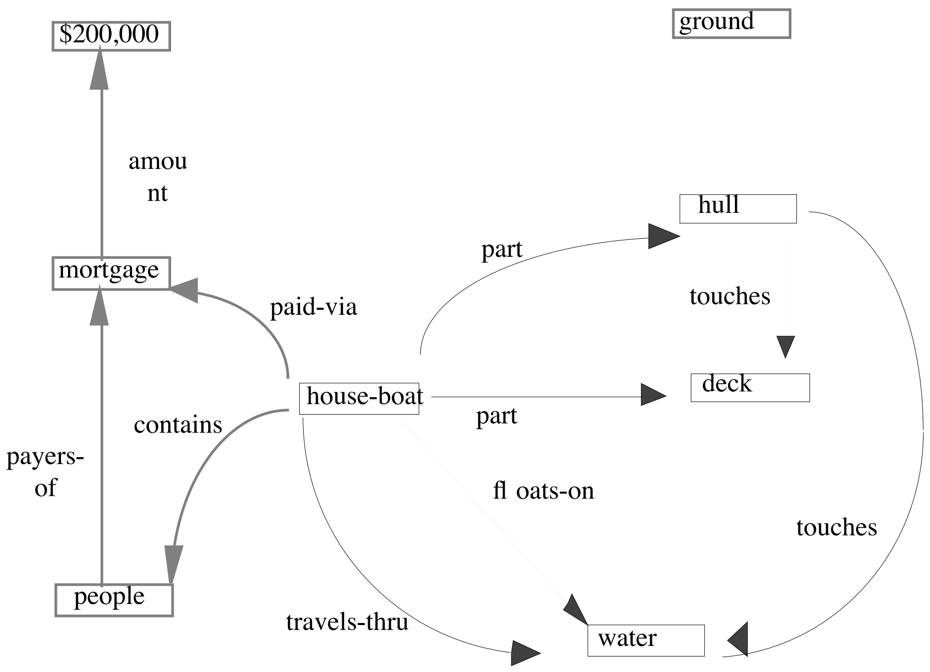


Figure 5.7: The combined graph. When *foundation* is overridden by *hull*, the *buried-in* relation is destroyed. This leaves the node *ground* disconnected from the combined graph. As a result, in the Graph Deconstruction Phase of the algorithm, no fact involving *ground* is included in the new concept definition.

5.8 Similarity Checking

When a pair of nodes are candidates for matching, the Base node is potentially able to fill the role of the Modifier node within the concept. Semantically, this amounts to the Base node being capable of having all the Modifier's properties. Therefore, a subprocess must perform some form of check on the pair for similarity. The mechanism for such a check is also somewhat beyond the scope of this thesis, but several possibilities can be touched upon, proceeding from those most conservative in confirming similarity:

1. reified specialization: if a specialization relation links the Base to the Modifier node, then the Base automatically inherits any properties of the Modifier and can definitely fill its role. For instance, a *baseball* could definitely fill the role of a *ball*;

2. comparison of properties: if there exists no specialization link between the nodes, yet they share most or all of their properties, then the Base node may fill the role of the Modifier node . For example, a *balloon* might share enough properties with a *ball* to fill its role;

3. failure to find contradiction: if the Base node can be given the Modifier node's properties without causing a contradiction, then the Base node might fulfill the role of the Modifier node;

4. world knowledge strategy: probably the most likely candidate for humans' judgements of similarity, this strategy draws upon knowledge from the outside world to assess the ability of the Base to fill the role of the Modifier.

From a practical standpoint, a similarity judgement could be performed manually by a knowledge engineer while still maintaining efficiency. Most concepts have far fewer components than facts, so approving a single match between Base and Modifier components would be tantamount to approving quite a few facts. This mirrors our intuitions of roles: if an A fills the role of a B, we can often make many inferences about A.

5.9 Match Outcomes

A similarity check either approves or rejects the match. If the match is approved (Fig. 5.5, part *D*), the Base node successfully fills the role of the Modifier node, and is retained in the graph. Because the algorithm recursively maps all relations of the Modifier node onto the Base node, the Base node will receive all properties of the Modifier.

If the match is rejected (Fig. 5.5, part *E*), then the Base node is incapable of filling the role of the Modifier. The Modifier node overrides the Base node, replacing it in the graph (Fig. 5.6). Overriding a Base node automatically invalidates all unexplored Base relations touching it; such relations are removed from the graph. This outcome's semantic interpretation is that Modifier and Base facts conflicted, causing removal of the conflicting Base facts. As a result, other Base nodes may lose their links to the original Base graph (Fig. 5.7). All Base nodes and relations not linked to the Base concept at the conclusion of the Graph Mapping Algorithm are dropped; they are not relevant to the modified concept anymore. Such an event will end the recursive traversal of the Base structure, though not that of the Modifier (Fig. 5.5, part *F*)

The formation of the construct *container-boat* is one example of a positive similarity judgment. In it, the boat's *hull* is matched with the *container*'s *wall*. Though *hull* is not a known specialization of *wall*, numerous similarities exist between the two concepts: they are both physical objects; they both have the *part-of* relationship with the larger concepts containing them; and, *wall* and *hull* contain the outer surface of container and boat, respectively. Therefore their match succeeds, and *hull* receives the properties of *wall* (Fig. 5.7).

5.10 Adding Component Nodes to the Base Concept

Not all component nodes of the Modifier concept may have counterparts in the Base and vice versa. In the first instance, unmapped components and relations from the Modifier are simply added to the base concept in order to effect a complete "overlay" of the modifier onto the base (Fig. 5.5, part *G*). For example, when *log* is modified to become a *container-log* (i.e. a log containing something), the *contents* component of *container* has no counterpart in *log*; logs do not commonly contain things in the classic sense of containership. Therefore, the *contains* relation and the *contents* component is simply added to *log* so that *log* can properly act in the role of a container. Extending the conceptual structure of the base concept with that of the modifier is one of the most obvious methods of knowledge transfer. Of course, one has difficulty viewing a completely solid log as a container, so some other structural properties of log may be overridden in the composition process.

Overall, these mapping processes are best characterized together as a quasi-unification of the base and modifier concept in which the base concept's objects are either made specializations of their counterparts in the modifier concept or replaced by them. Although these specialization relationships may or may not be recorded in the KB, their defining properties must hold: the base component inherits all facts about the modifier concept. This is in contrast to analogical reasoning, in which the base concept simply inherits a pattern of relations between its components.

5.11 Coherent Complex Concepts

The requirement that all matched components of the base must act in the role of their counterparts does lead to potential constraints of the roles a particular base concept may fill. This in turn provides a coherency constraint upon the kinds of concepts that can be constructed. For example, it is hard to imagine a cocktail party filling the role of an apple, as a modified base concept would have to resemble a cocktail party yet have seeds and be a plant. An appropriate modifier concept must not contain too many facts that contradict (fail to unify with) those of the base concept. This restriction explains why most real-world concepts have modifier and base concepts of the same high-level type: e.g. both are scripts, or both are physical-objects, relations, etc.

5.12 Graph Deconstruction Phase

Often, the factual information in a conceptual graph must be converted back into individual facts to be useful. Question-answering, for example, might require a justification for an answer to be given: such a justification should be in the form of individual rules. Using viewpoints [1] to extract relevant sets of facts about a concept also requires some sort of deconstruction of the concept graph. Therefore, the following deconstruction algorithm is used to convert the modified Base graph back into a set of facts that shall serve as the new concept definition. Intuitively, such facts are those contained in the graph resulting from the Graph Mapping phase. To separate them into individual facts, we implement an algorithm of unification and substitution involving the graph and the original Modifier and Base facts.

5.13 A Deconstruction Algorithm

Because the new concept is intuitively a subclass of the Modifier, the new concept should have all properties (i.e. facts) of the Modifier. However, the new concept still contains components from the Base concept, and these components may replace the original components of Modifier facts. Therefore, all facts from Modifier undergo a process of match substitution (Fig. 5.8, part *A*). If a Modifier fact contains a node that was matched with a Base node in the Graph Mapping phase, the Base node is substituted for the Modifier node in the fact. After any match substitutions are made, the deconstruction algorithm attempts to unify each (altered) Modifier

GRAPH DECONSTRUCTION ALGORITHM

```
Select_Modifier_Facts(MFacts)
    FOR each MFact                                                        *A*
        FOR each MNode in MFact that is matched with a BNode
            replace all occurrences of MNode with BNode
        IF MFact unifies with Graph                                       *B*
            label of MFact's main concept node := new concept name
            include MFact in new defn

Select_Base_Facts(BFacts)
    FOR each BFact
        IF BFact unifies with Graph                                       *C*
            label of BFact's main concept node := new concept name
            include BFact in new defn
```

Figure 5.8: A graph deconstruction algorithm.

fact with the graph (Fig. 5.8, part *B*); any facts that succeed are included in the final concept definition.

The new concept can also contain facts from the base concept, though some of them may have been invalidated if Modifier nodes overrode Base nodes in the Graph Mapping phase. The deconstruction algorithm therefore attempts to unify each Base fact with the graph (Fig. 5.8, part *C*); any that succeed are included in the final concept definition.

The final step of the deconstruction algorithm is to rename the main concept node in all of the selected Modifier and Base facts. All these facts now concern the new concept, so their main concept nodes (i.e. the node marked Modifier or Base) is renamed with the new concept name (Fig. 5.9).

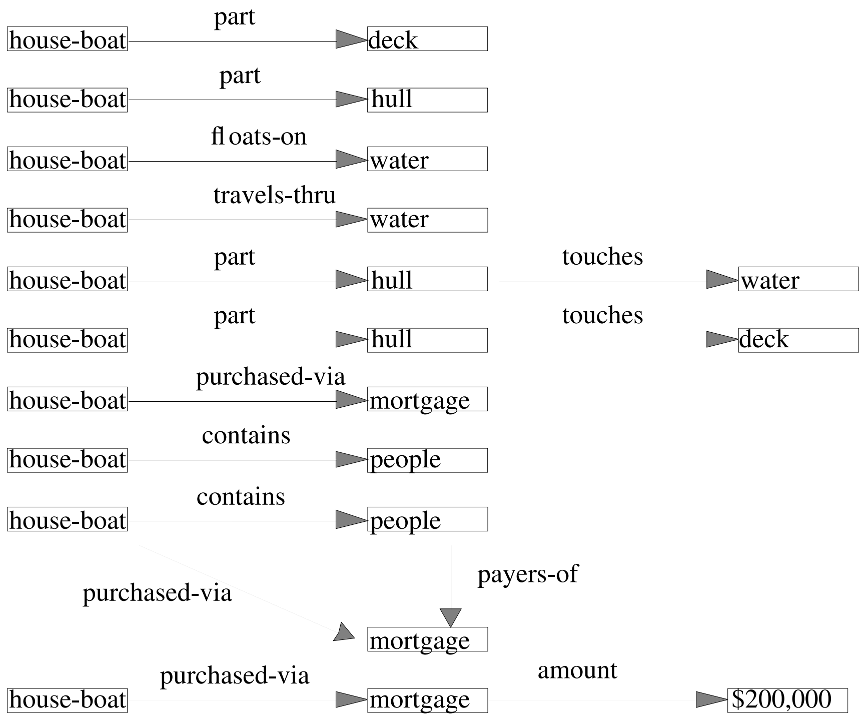


Figure 5.9: The new concept facts for *house-boat*, the result of deconstructing the graph formed by *house* and *boat*.

Chapter 6

Commentary

6.1 The Power of Role-Based Composition

The role-based algorithm outlined above explains several details associated with concept composition that prior approaches cannot. It shares several of these representational advantages with inheritance-based algorithms. For example, a role-based algorithm implements inheritance efficiently and is easy to understand. In contrast to analogical reasoning, a role-based algorithm enables simple attribute transfer (as opposed to transfer of relational-structure). And unlike compositional modeling, a role-based approach does not require physical structure information for proper composition.

Role-based composition goes beyond inheritance in its commitment to modeling the complex, theory-like structure of concepts. Such a commitment enables it to adress several issues that inheritance cannot. The most important such issue is the complex interaction of features from the parent concepts. By representing composition as a process of mapping components and relational structure of the modifier onto the base, related features (or facts, as represented here) are automatically mapped onto each other. Viewing mappings as instances of attempted role-filling intuitively models a number of different featural interactions.

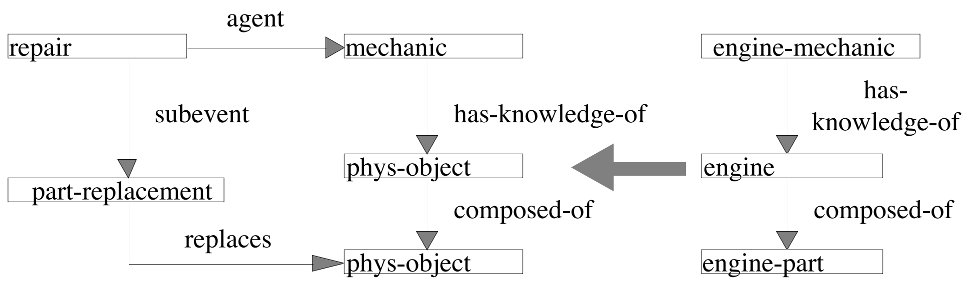


Figure 6.1: The composition of *engine-repair*.

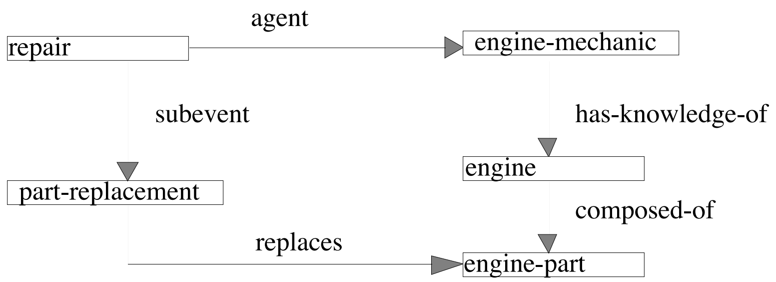


Figure 6.2: The resulting composed concept: *engine-repair*.

In contrast to an inheritance-based composition method, which simply links a modifier onto a base via one mediating relation, role-mapping engenders more global interaction. For example, in the composition of *engine-repair*, role-mapping (Fig. 6.1) enables additional inference beyond that of simply identifying *engine* as the *object* of the *repair*. Facts were inferred about the type of mechanic involved and his/her knowledge, as well as the sort of parts used (Fig. 6.2). This additional inference is a direct result of viewing concepts as factual, theoretical structures instead of mere attribute lists.

A second advantage of role-mapping over inheritance-based methods is that the notion of role-filling enables a richer account of how concept structures interrelate. Instead of simply specializing one another, concept features can be overridden, enabling the larger structure to successfully fill a role. This models examples such as *house-boat*, in which *house* must be modified (by losing its connection to the *ground*) in order to fill the role of a *boat*. An inheritance-based account cannot model such interactions.

A final, related advantage of role-mapping is that it provides a framework for understanding how world knowledge can be used in the composition process. World knowledge's primary function is to help make compositions coherent. This function is modeled via world knowledge participation in similarity judgements of component role-mappings. The judgement process provides a first, rough explanation of examples like the one above, in which our outside knowledge of land, water, and flotation dictate the modification of *house*. World knowledge can be similarly invoked to explain the rejection of sufficiently incoherent compositions such as *green ideas* or *stone-water*: in these cases, the two concept structures are sufficiently different as to be effectively unmappable.

World knowledge promotes coherence even more fundamentally by helping determine the proper mediating relation between the base and modifier. Although this issue is not explored in my thesis, it can be viewed as a problem of finding a maximally coherent mapping between two concepts. As mentioned before, this formulation makes several mechanisms from analogical reasoning applicable. Modeling composition as mapping therefore leads to promising possibilities for explanation, in contrast to inheritance mechanisms.

Theory-based notions of concepts address problems and examples which stymie similarity-based accounts. It is therefore not surprising that a model of composition based on such notions is likewise more potent than inheritance and other mechanisms.

6.2 Applications to Compositional Modeling

A role-based model of concept composiion also has a practical application to knowledge engineering techniques such as compositional modeling. In compositional modeling, models of a phenomenon are created by composing model fragments. However, the knowledge engineer is still confronted by the bottleneck of creating these basic fragments.

This problem is made more acute by the modular approach taken by compositional modeling. To maximize fragment modularity, model fragments do not always correspond to common concepts (e.g. radio, animal, circuit). Instead, they often model physical components of such concepts (the radio knob), or operations and transformations upon a common concept (electrical voltage in a circuit). Therefore, to adequately model a domain, the knowledge engineer must still enter a massive number of concepts, possibly more than in conventional knowledge bases. The problem is actually still worse, because the knowledge engineer must also create multiple models of a phenomenon to account for differing modeling and operating conditions (e.g. constant battery voltage vs. temperature-sensitive battery voltage).

Therefore, a general-purpose algorithm to efficiently and accurately compose model fragments would be quite useful. I claim that role-based composition is a natural solution. It is a composition process for concept definitions emphasizing complex theoretical structure, which model fragments are. Because it is capable of mapping systemic structural information from one fragment to another, it would be especially useful assisting with the creation of multiple models under varying modeling/operating conditions.

A particular assumption class (a set of fragments modeling the same entity under differing modeling conditions) might be applied to a fragment modeling another entity to create a new assumption class modeling information from both entities. For example, a *container* might have an assumption class specifying its properties at different levels of abstraction. One fragment in the class might describe the container qualitatively, as a set of parts related by simple relations. Another fragment might represent the container parts spatially with a more complicated coordinate system. Still another might represent that same container at an atomic level.

The information from this assumption class could be transferred to another fragment, such as *kayak*, in order to easily represent the *kayak*'s container-like properties. This type of composition cannot be done via compositional modeling, which relies on exact matches between variables representing each fragment's physical structure. Although structural elements of *kayak* and *container* need to be matched, these elements use different completely different variable names (e.g. *container-wall* vs. *hull*), and therefore cannot be properly linked.

Role-based composition would also need matching guidance, provided by the knowledge engineer via similarity judgements. However, such a process might only have to take place once, to match *kayak* with the first *container* fragment in the assumption class. Thereafter, the mapping between *kayak* and *container*'s structure might be reused when *kayak* is matched with the other fragments in the *container* assumption class. In this way, an entirely new assumption class modeling *kayak*'s *container*-like properties is created efficiently and accurately.

The following example demonstrates the usefulness of a role-mapping process in composing multiple fragments from an assumption class and a single fragment. In this example, various possible models of a container are presented. (Figs. 6.3, 6.4, 6.5)

The first represents the container as a space which physical objects can pass into or out of. Among its invocation conditions is the requirement that it have a portal larger in size than the container object. The second models the container as a space that liquids can pass into or out of. In this model, the portal's size is not relevant. The third models a container with a sealed portal; among its implied relations is the fact that a contained object cannot touch entities outside of the container.

Each of these models uses different representational or invocational assumptions, yet they share structural similarities that can be exploited to easily transfer information to new fragments. A simple model fragment of a kayak is presented as an example. In order to create new fragments of kayak which model its container-like properties, we initiate a role-mapping process between the first container fragment and the kayak fragment. (Fig. 6.6)

Such a mapping process would likely rely on a knowledge engineer to supply the proper mappings between *container-object-exit* and *kayak*, producing the following graph. The resulting knowledge fragment models the object-containing properties of *kayak*.

The set of mappings can be easily reused to compose additional fragments modeling different *container*-like aspects of *kayak*. (Fig. 6.7) This technique effectively transfers the alternative conceptual structures of *container* to *kayak*, achieving a degree of modularity and reuse unavailable to compositional modeling.

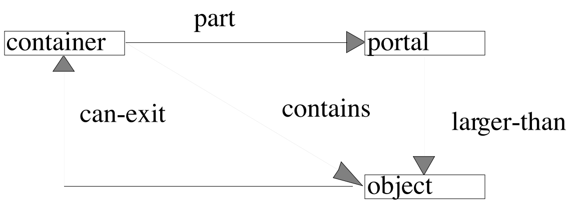
6.3 A Concrete Model of the Instantiation of Goals in Planning

Aside from its potential usefulness to knowledge engineers, the role-based composition model can concretize psychological theories which rely on composition yet offer vague accounts of it. Barsalou's model of planning is one instance of such a theory: his account relies on general-purpose compostion algorithm which is never described. I shall examine an example from his work and demonstrate how role-mapping can help achieve the progressive restriction effects Barsalou outlines.

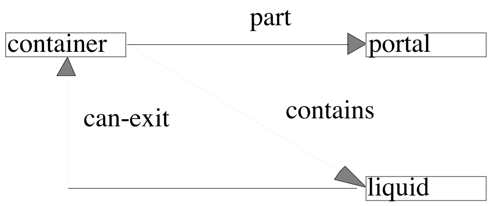
The initial concept is a *vacation*, which has the following attributes: *activity, time, location, and actor*. We begin with a group of facts concerning each of these entities and others (Fig. 6.8). Initially, only the facts about the vacation and its attributes are activated, but as the process progresses, choices are made, and additional facts about each choice are added (Fig. 6.9 - 6.13). For example, the decision to spend the vacation *snow-skiing* causes various facts about *snow-skiing* to be added, constraining the choice of *location* (Fig. 6.12). Optimizations are also used: the desire for *privacy* requires a *location* that is *minimally popular* (Fig.

```
(defModel (container-object-exit ?c)
    individuals         ((?c)
                         (?p :conditions        (portal ?p)
                                                (part-of ?p ?c))
                         (?o :conditions        (phys-obj ?o)
                                                (contains ?c ?o)
                                                (larger ?p ?o)))
    assumptions         ((CONSIDER (exists ?c))
                         (CONSIDER (geometric-properties ?c))
                         (CONSIDER (geometric-properties ?p))
                         (CONSIDER (geometric-properties ?o)))
    relations           ((can-exit ?o ?c)))
(defModel (container-fluid-exit ?c)
    individuals         ((?c)
                         (?p :conditions        (portal ?p)
                                                (part-of ?p ?c))
                         (?o :conditions        (liquid ?l)
                                                (contains ?c ?l)))
    assumptions         ((CONSIDER (exists ?c))
                         (CONSIDER (geometric-properties ?c))
                         (CONSIDER (geometric-properties ?p))
                         (CONSIDER (contained-liquid-behavior ?l)))
    relations           ((can-exit ?o ?c)))
(defModel (sealed-portal-container ?c)
    individuals         ((?c)
                         (?o :conditions        (contains ?c ?o)
                                                (larger ?p ?o)
                                                (sealed ?p)
                                                (?o2 :conditions (not (contains ?c ?o2)))))
    assumptions         ((CONSIDER (exists ?c)))
    relations           ((not (touches ?o ?o2))))
(defModel (kayak ?k)
    individuals         ((?k)
                         (?s :conditions (seat ?s)
                                                (part-of ?s ?k))
                         (?c :conditions (cockpit ?c)
                                                (part-of ?c ?k))))
```

Figure 6.3: Three model fragments of *container* and one fragment of *kayak*, represented using Falkenhainer and Forbus' [13] conventions.

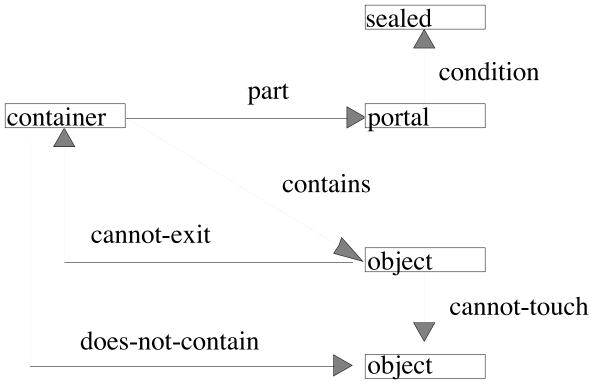


A representation of a conceptual fragment:
a container with an object inside.

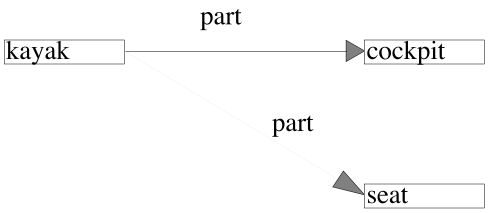


A representation of a conceptual fragment:
a container with a liquid inside.

Figure 6.4: Two model fragments of *container*, represented using conceptual graphs.



A representation of a conceptual fragment:
a container with a sealed portal.



A representation of a proposed conceptual fragment
specified only by its physical structure: a kayak.

Figure 6.5: A third model fragment of *container* and one fragment of *kayak*, represented using conceptual graphs.

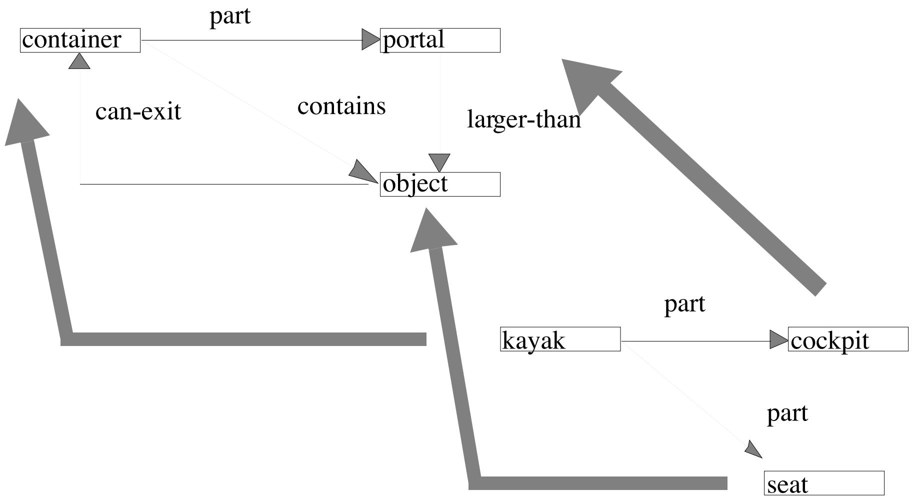


Figure 6.6: Composing the two knowledge fragments with the help of a knowledge engineer.

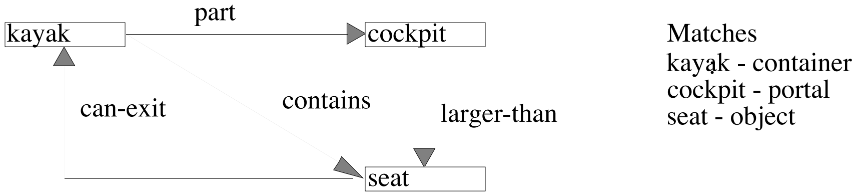


Figure 6.7: The finished composition. The matches created (shown at right) can be reused to compose *kayak* with different fragments automatically. In this way, fragments for kayak under different modeling assumptions can be created efficiently.

6.10). Ultimately, the vacation concept is sufficiently constrained that actual location candidates can be considered: *Peru, Ecuador*, and *Chile* (Fig. 6.14). This last step involves similarity/typicality calculations that are beyond the scope of the composition process.

By demonstrating that role-based composition can model important aspects of Barsalou's example, we lend credence to his psychological claims. These claims, with their emphasis on a conceptual structure of interrelated constraints and optimizations, in turn support Murphy and Medin's ideas by showing their power to explain experimental data.

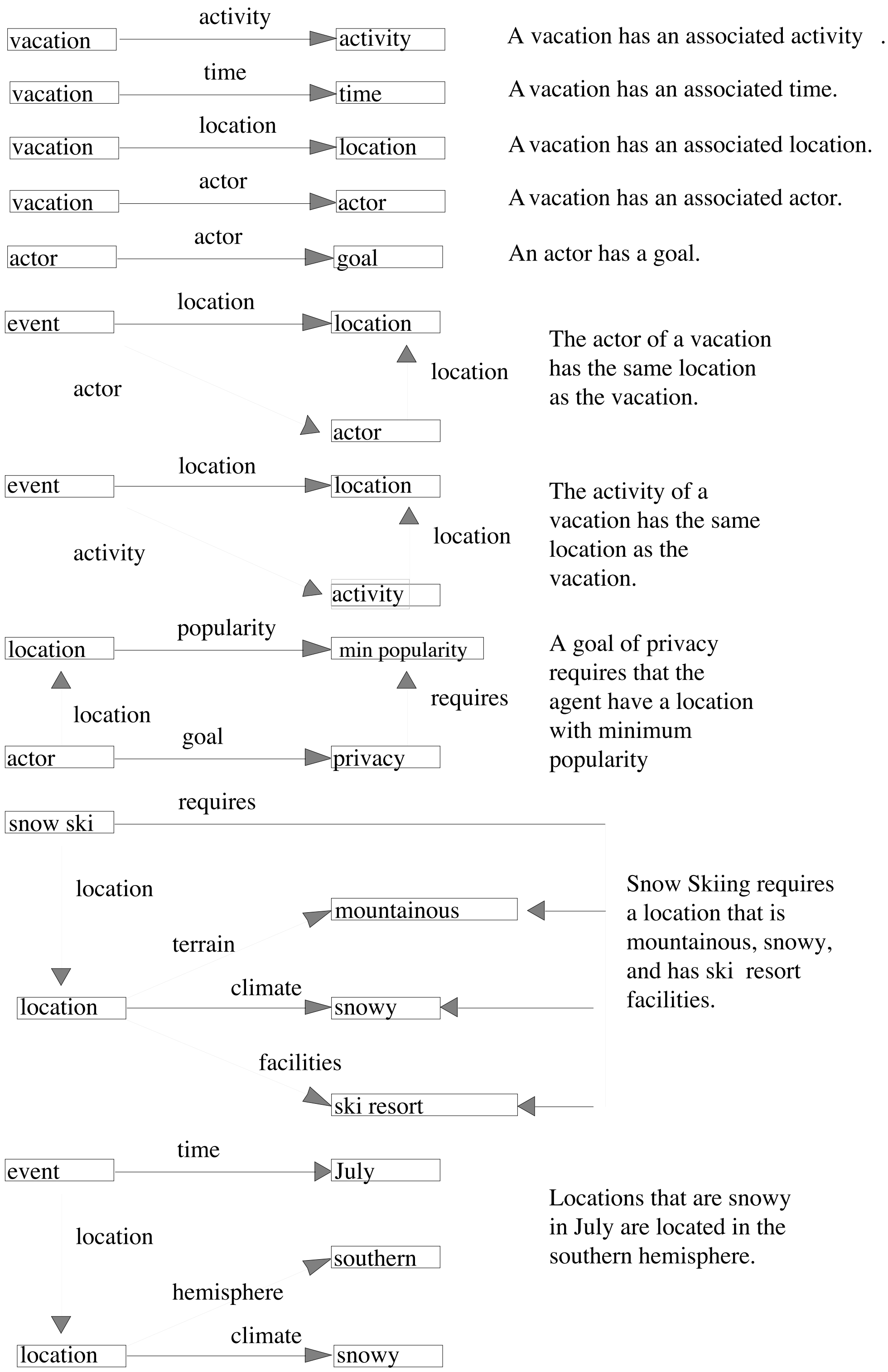


Figure 6.8: The concept facts for Barsalou's [4] vacation example.

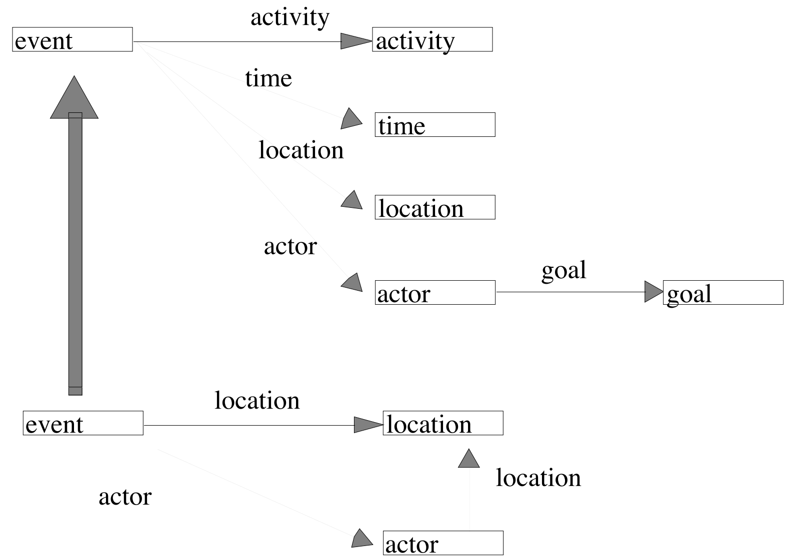


Figure 6.9: Adding a constraint about events: an event takes place where its actor is located.

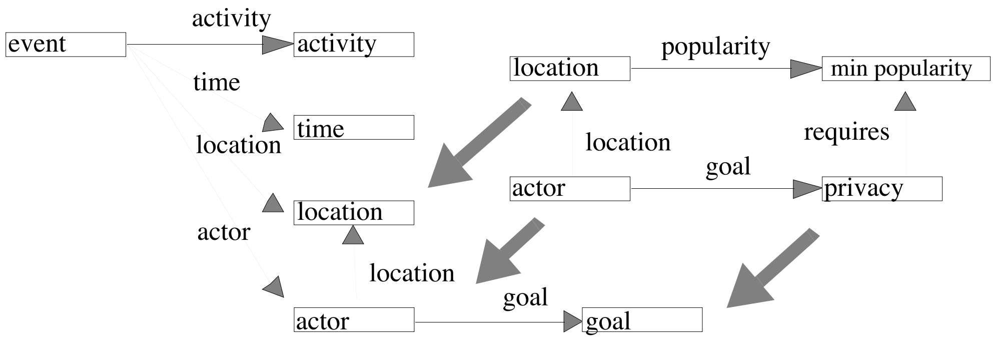


Figure 6.10: Choosing a possible goal: privacy. Its constraints are added.

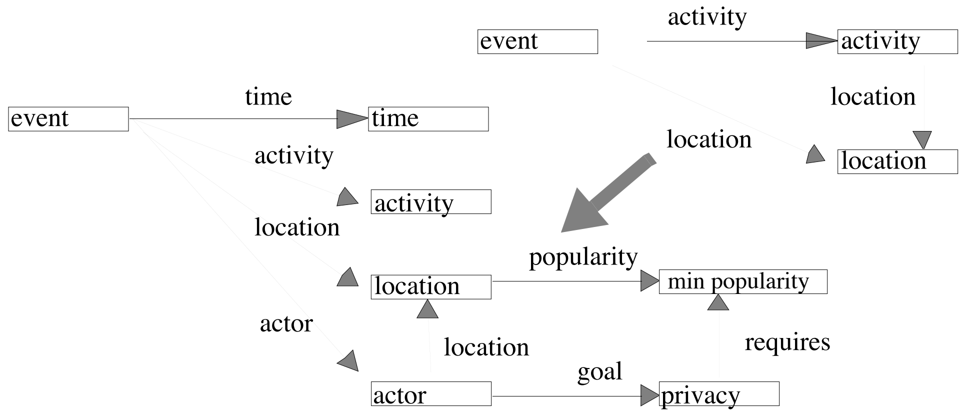


Figure 6.11: Adding a constraint about activities.

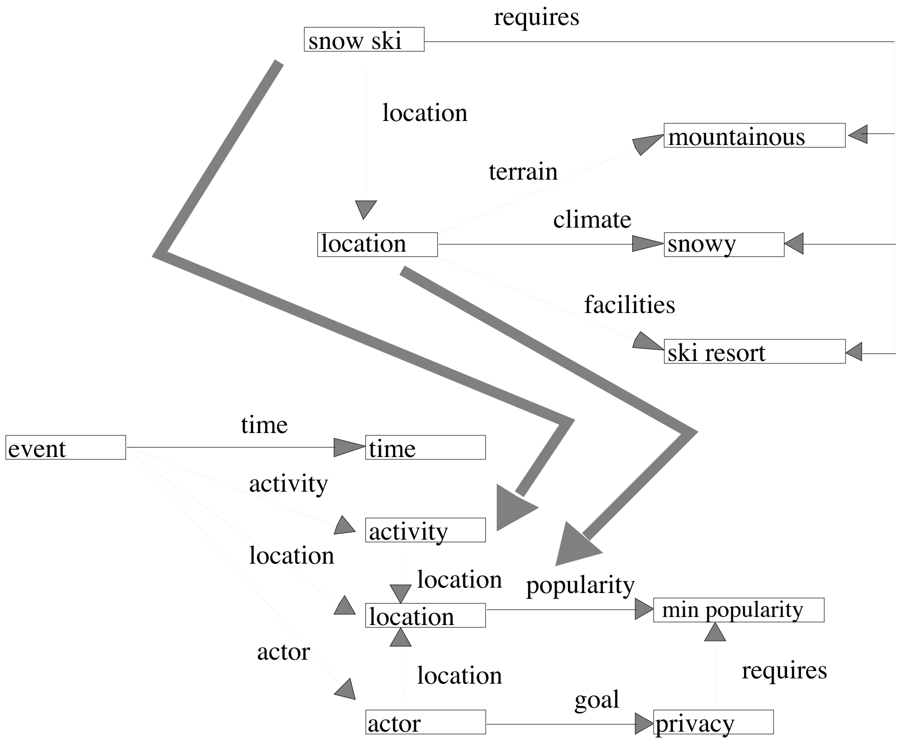


Figure 6.12: Choosing a possible vacation activity: snow skiing and its associated constraints.

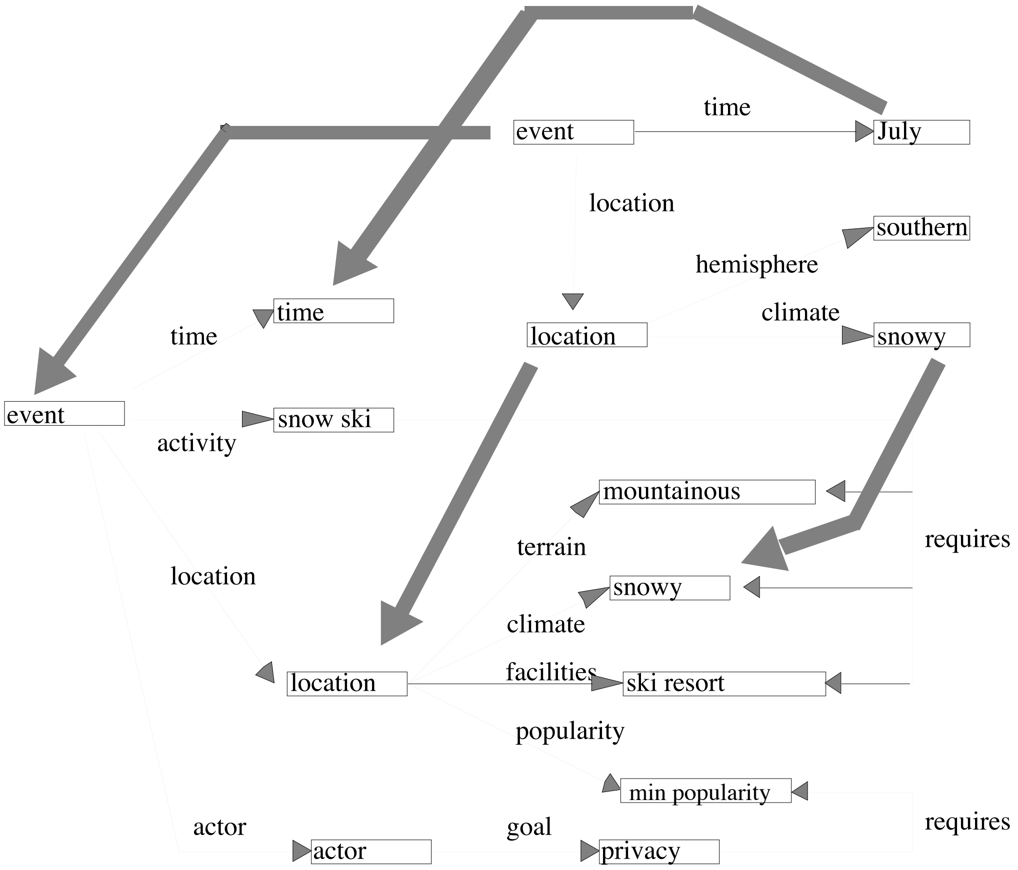


Figure 6.13: Choosing a possible vacation time: July and its associated constraints.

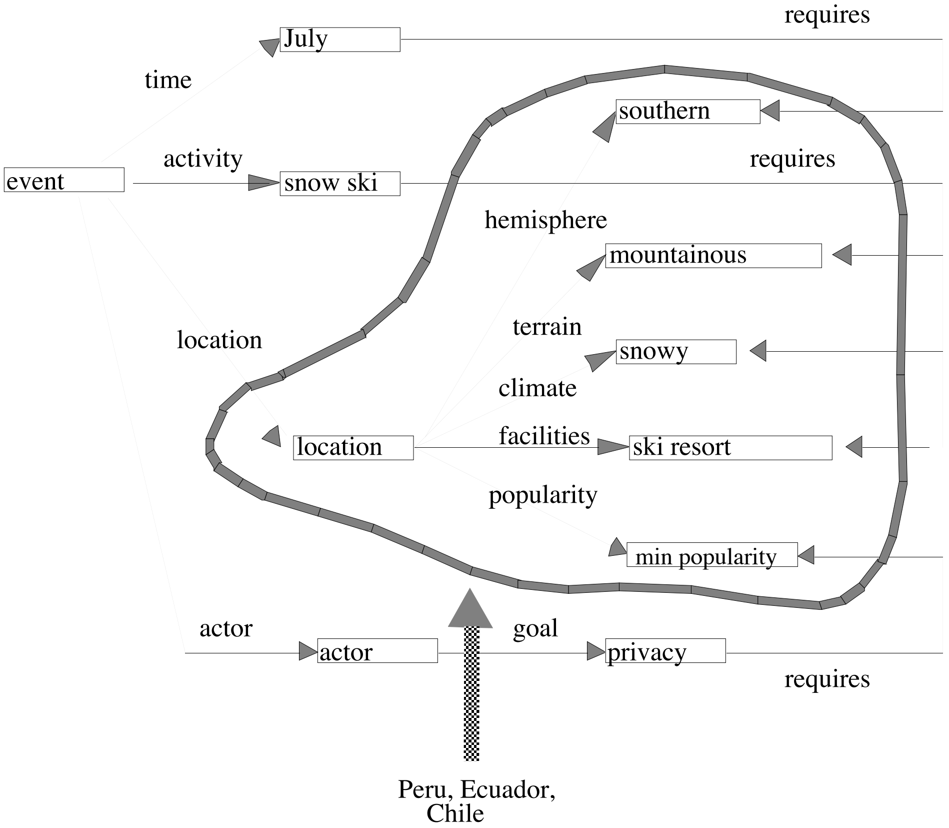


Figure 6.14: The final vacation concept: it is now constrained enough so that specific location candidates can be chosen: Peru, Ecuador, and Chile.

Chapter 7

Conclusions and Future Work

This thesis provides a concrete model for the composition of complex concepts; its role-based process offers a richer explanation of composition than previously available. In particular, it provides a powerful explanation of how concepts, which clearly have complex structures relating together many different facts, can be combined in a coherent way. However, my model leaves many aspects of the process unexamined. For example, it neglects the mechanism by which two concepts are initially linked at the correct node. Similar mechanisms have been examined to some degree by theorists of analogical reasoning (Gentner 1983) and these might be applicable to the problem. In any case, it seems clear that this area is one of the most dependent on contextual knowledge for success.

Another concern involving context has implications for concepts' very definitions. Barsalou [4] found evidence that only portions of concepts' full definitions are activated and used in particular contexts; a concrete explanation of how this occurs could be quite useful in understanding the tractability of human reasoning. This research direction also might examine whether, in fact, there are some concepts which override portions of each other's structure, as opposed to my model, in which only the Modifier can override the Base. If such mutual override is possible (which is still an open question in my mind), then it might occur because only portions of a full concept are actually being used in the composition process.

Related but more complex issues deal with how conflicting pressures can determine the various possible outcomes of the composition process. One mechanism probably subject to these pressures is the similarity checking process I have tried to sketch. This process is still somewhat mysterious, but presumably uses contextual knowledge to make local and global judgements of structural compatibility between concepts. What may be difficult to explain is how mediation between positive and negative pressures is achieved; we experience such pressures every time we ponder a questionable concept such as *artificial intelligence*. A related conundrum concerns the possiblity of composition involving more than two concepts, as seems to occur when we humans experience a sudden crystallization of jumbled thoughts into one orderly structure.

Remaining issues concern the representational (and therefore structural) assumptions I have incorporated. For example, a wealth of psychological evidence indicates that concept structure is not just theoretical, but also oriented around simple perceptual features. I have to a great degree neglected these aspects of conceptual structure in my model. Although they are perhaps not as important to issues of reasoning with complex concepts, they are still vitally important to the categorization of exemplars, and need further explanation.

These and many other issues remain to be examined. However, the model presented in this thesis captures important aspects of the composition process. Besides modelling concepts as theories, it captures an important intuition: that combining concepts (and perhaps all thought, really) involves a process of matching these complex, interrelated factual structures together. It also captures a second intuition: that such matching is only possible if these structures are sufficiently similar to form a coherent concept.

Bibliography

[1] Acker, L. and Porter, B. Extracting viewpoints from multifunctional knowledge bases. Tech Report AI-TR92-180, AI Laboratory, University of Texas at Austin, TX, 1992.

[2] Ait-Kaci, H. *The Wild LIFE Handbook (prepublication edition)*. 1994, Digital Equipment Corporation.

[3] Bareiss, E. R. and Porter, B. W. A survey of psychological models of concept representation. Tech Report AI-TR87-50, AI Laboratory, University of Texas at Austin, TX, 1987.

[4] Barsalou, L.W. and Hale, C. R. Components of conceptual representaiton: From feature lists to recursive frames. In Van Mechelen I., Hampton J., Michalski R.S., and Theuns P. (eds.) *Categories and Concepts: Theoretical Views and Inductive Data Analysis*. Academic Press, San Diego, 1991. 97-144.

[5] Barsalou, L. W. Deriving categories to achieve goals. *The Psychology of Learning and Motivation*, 1991, 27, 1-64.

[6] Barsalou, L. W. The instability of graded structure: Implications for the nature of concepts. In Neisser U. (ed.) *Concepts and Conceptual Development: Ecological and Intellectual Factors in Categorization*. Cambridge University Press, New York, 1987. 101-140.

[7] Barsalou, L. W. Ideals, central tendency, and frequency of instantiation as determinants of graded structure in categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition.* 1985, 11(4), 629-654.

[8] Barsalou, L. W. Ad hoc categories, *Memory and Cognition*, 1983, 11(3) 211-227.

[9] Bobrow, D. and Winograd, T. An Overview of KRL, a knowledge representation language. *Cognitive Science*, 1977, 1.

[10] Brachman, R., McGuinness, D., Patel-Schneider, P., Alperin-Resnick, L., and Borgia, A. Living with CLASSIC: When and how to use a KL-ONE-like language. In Sowa J.F. (ed.) *Principles of Semantic Networks*, Morgan Kaufmann, Los Altos, 1991.

[11] Chapman, D. Cognitive cliches. Working Paper 286, AI Lab, MIT, 1986.

[12] Cohen, B. and Murphy, G. L. Models of concepts. *Cognitive Science*, 1984, 8, 27-58.

[13] Falkenhainer, B. and Forbus, K. D. Compositional modeling: finding the right model for the job. *Artificial Intelligence*, 1991, 51, 95-143.

[14] Feigenbaum, E. A. The art of artificial intelligence: Themes and case studies of knowledge engineering. In *IJCAI-77*, Cambridge, MA, 1977, 1014-1029.

[15] Gentner, D. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 1983, 7, 155-170.

[16] Greiner, R. Abstraction-based logical inference. In Helman D.H. (ed.) *Analogical Reasoning*, Kluwer Academic Publishing, Boston, 1988.

[17] Holyoak, K. J. and Thagard, P. Analogical mapping by constraint satisfaction. *Cognitive Science*, 1989, 13, 295-355.

[18] Kedar-Cabelli, S.T. Toward a computational model of purpose-directed analogy. In Prieditis (ed.) *Analogica: Proceedings of the First Workshop on Analogical Reasoning*, Morgan Kaufmann, Los Altos, 1987.

[19] Levesque, H. J. and Brachman, R. J. A fundamental tradeoff in knowledge representation and reasoning (Revised Version). In Brachman R.J. and Levesque H.J. (eds.), *Readings in Knowledge Representation*, Morgan Kaufmann, Los Altos, 1985.

[20] Levy, A. Y. Irrelevance reasoning in knowledge based systems. Tech. Report STAN-CS-93-1482, Dept of Computer Science, Stanford University, 1993.

[21] Murphy, G. L. and Medin, D. L. The role of theories in conceptual coherence. *Psychological Review*, 1985, 92, 289-316.

[22] Murphy, G. L. Comprehending complex concepts. *Cognitive Science*, 1988, 12, 563-586.

[23] Murray, K. KI: An experiment in automating knowledge integration. Tech Report AI-TR88-90, AI Laboratory, University of Texas at Austin, TX, 1988.

[24] Smith, E. E. and Osherson, D. N. Conceptual combination with prototype concepts. *Cognitve Science*, 1984, 8, 337-361.

[25] Sowa, J. F. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA, 1983.

[26] Winston, P.H. Learning and reasoning by Analogy. *Communications of The ACM*, 1980, 23, 689-703.

[27] Zadeh, L. Fuzzy sets. *Information and Control*, 1965, 8, 338-53.

Vita

Carl Floyd Andersen was born in Columbus, Ohio on July 27, 1967, the son of Lucille Brooke Andersen and Kenneth Lee Andersen. He graduated from Stanford University in 1989 with the Degree of Bachelor of Arts in Philosophy. In September, 1993, he entered The Graduate School at The University of Texas at Austin.

Permanent Address: RR1, Box 317B
Newcastle, ME 04348

This thesis was typeset with LaTeX 2_ε[1] by the author.

[1]LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin.