Computers Math. Applic. Vol. 23, No. 2–5, pp. 51–73, 1992
Printed in Great Britain.
0097-4943/92 $5.00 + 0.00

CONCEPTUAL DEPENDENCY AND ITS DESCENDANTS

STEVEN L. LYTINEN
Artificial Intelligence Laboratory
The University of Michigan
Ann Arbor, MI 48109, U.S.A.

Abstract — This paper surveys representation and processing theories arising out of conceptual dependency theory. One of the primary characteristics of conceptual dependency was the notion of a canonical form, built out of a small number of primitive representations. Although the notion of primitives has largely been lost in subsequent work, many other of the basic notions of CD have remained. In particular, the idea of building representations around inferential capabilities has prevailed in this family of research. The result is a set of representational structures, all of which are highly knowledge-intensive. The use of these structures in various processing theories has led to knowledge-based theories of language understanding, planning, reasoning and other tasks, which have contrasted sharply with the traditional search-oriented approaches used in other systems.

1. INTRODUCTION

This paper will outline a family of representational theories which have developed from the common ancestor of conceptual dependency (CD) theory created by Roger Schank [1,2]. In addition, we will discuss some of the programs that these representations have been used in, as well as the reasons why these representations have proven to be useful for these programs.

The original goal of conceptual dependency work was to develop a "representation of the conceptual base that underlies all natural languages." [1, p. 554] This rather lofty goal resulted in the proposal of a small set of primitive actions (10–12 or so), and a set of dependencies which connected the primitive actions with each other and with their actors, objects, instruments, etc. The claim was that this small set of representational elements could be used to produce a *canonical form* representation for English sentences (and other natural languages).

Almost twenty years later, no one seems to take seriously the notion that such a small set of representational elements covers the conceptual base that underlies all natural languages. Representation theories that have descended from CD, such as scripts [3], have had a virtually unlimited number of vocabulary items. The notion of a "primitive" seems to have disappeared. However, as we shall see, CD has certainly influenced these subsequent theories. One of the questions that this paper will address, then, is: What did we learn from conceptual dependency? What does conceptual dependency theory have in common with the descendant theories? As we examine these descendants and the tasks that they have been used for, we will try to discover the common threads that connect this family of research.

We will begin the paper by briefly reviewing conceptual dependency theory. For a more thorough review, see [2]. Then we will discuss subsequent theories, including scripts, plan/goal representations, Memory Organization Packets (MOPs), and Thematic Organization Packets (TOPs). Finally, we will examine some of the processing theories that have accompanied these representations. In particular, we will focus on the tasks of language understanding and reasoning. As we will see, the basic assumptions behind the representation theories have a large effect on the processing theories that use them.

2. CONCEPTUAL DEPENDENCY

2.1. A Brief Review of CD Notation

Conceptual dependency theory was based on two assumptions:

1. If two sentences have the same meaning, they should be represented the same, regardless of the particular words used.

2. Information implicitly stated in the sentence should be represented explicitly. That is, any information which can be inferred from what is explicitly stated should be included in the representation.

These assumptions have many implications for what a representation language should look like. The first assumption implies that representations must be *general*; that is, they must capture the similarities in meanings of synonyms. For example, since "get" and "receive" can be used synonymously in many contexts, their conceptual representations in these contexts ought to reflect this fact by consisting of similar, if not identical, predicates.

The assumption that representations ought to explicitly represent implicit information means that inferences must be made in order to produce complete representations. If a sentence implies information without explicitly stating it, then in order to include that information in the representation, machinery must exist which can infer it from what is explicitly stated. This means that representations must support inference. In order to do this efficiently, they must be *canonical*. Whenever a particular inference can be made, it would be desirable if the same inference rule could always be used to make it. Without a canonical representation, several rules would be required, one for each different possible representation form.

Given these representational requirements, then, the vocabulary for conceptual dependency consisted of the following:

- a set of *primitives*, used to represent actions in the world

- a set of *states*, used to represent preconditions and results of actions

- a set of *dependencies*, or possible conceptual relationships which could exist between primitives, states, and the objects involved.

Representations of English sentences could be constructed by piecing together these building blocks to form a *conceptual dependency graph*.

PTRANS: The transfer of location of an object
ATRANS: The transfer of ownership, possession, or control of an object
MTRANS: The transfer of mental information between agents
MBUILD: The construction of a thought or of new information by an agent
ATTEND: The act of focusing attention of a sense organ toward an object
GRASP: The grasping of an object by an actor so that it may be manipulated
PROPEL: The application of a physical force to an object
MOVE: The movement of a bodypart of an agent by that agent
INGEST: The taking in of an object (food, air, water, etc.) by an animal
EXPEL: The expulsion of an object by an animal
SPEAK: The act of producing sound, including non-communicative sounds

Figure 1. The conceptual dependency primitives.

The set of primitives varied somewhat during the course of conceptual dependency theory, but it consisted of approximately 10–12 predicates, each of which represented a type of action. The primitives are shown in Figure 1.

Each primitive had a set of *slots* associated with it, from the set of conceptual dependencies. Associated with each slot were restrictions as to what sorts of objects could appear in that slot. For example, the slots for PTRANS were the following:

> ACTOR: a HUMAN (or animate object), that initiates the PTRANS
> OBJECT: a PHYSICAL OBJECT, that is PTRANSed (moved)
> FROM: a LOCATION, at which the PTRANS begins
> TO: a LOCATION, at which the PTRANS ends.

In order to make explicit the information implicitly presented in the text, inference rules were written based on the primitives. For example, from the primitive PTRANS, the inference could be made that the OBJECT which was PTRANSed was initially in the FROM location, and after the PTRANS was in the TO location.

The need to make inferences in order to explicitly represent implicit information suggested a criterion for what made a good primitive: it should support a cluster of reliable inferences. Thus, PTRANS constituted a good primitive because all PTRANS's shared several common inferences: from the representation one could infer the prior location of an object being PTRANSed, the location subsequent to the PTRANS, that the source and destination of the PTRANS must be locations, etc. These same inferences could be made no matter what type of PTRANS: flying, driving, walking, falling, and so on.

Conceptual dependency representations were written graphically as shown in Figure 2. The actor of a primitive action was connected to the primitive using a double arrow; the object appeared to the right with a single arrow connection, and the source and destination (TO and FROM) appeared to the right of the object. Thus, the sentence, "John gave Mary a book" would be represented as shown in Figure 3. John was explicitly represented as both the ACTOR and FROM slots of this action, since it was assumed that John had control/possession of the book originally.

Although individual primitives grouped together a set of similar actions, it was important that distinctions between these actions could also be captured in CD notation. For example, reading and talking are both types of MTRANS's, but there are obvious differences between them: reading involves using one's eyes while talking involves using one's mouth and ears; the source of information in reading is an inanimate object (such as a newspaper) while the source in talking is a human, etc. In order to express differences between similar actions, several primitive actions could be connected to each other, using one or more of the conceptual dependencies. For example, the CD graph in Figure 4 represents the action of reading. The "i" link in this graph stands for an *instrumental* connection between the MTRANS and the MOVE.

Talking, on the other hand, might be represented as shown in Figure 5. Again, an instrumental connection expresses the relationship between the MTRANS and John's MOVEment of his mouth and Mary's ATTENDing with her ears.

Sometimes connections also involved explicit mention of one or more intermediate states or actions. Consider this example, from [3]:

> John cried because Mary said she loved Bill.

The causal connection between John crying and Mary loving Bill is not direct: most readers infer that John loves Mary, realizes that Mary does not love him back since she loves Bill, and therefore is sad. This might be represented as shown in Figure 6. In this graph, "I" stands for an *initiate* link, connecting together an event and a mental state; and "I/R" stands for *initiate/reason*, a complex connection between two mental events.

Representations could be arbitrarily complex, sometimes making explicit a whole series of inferences that could be made. These complex sets of primitive actions, states, and conceptual dependencies were called *causal chains*. For example, consider the following sentence:

> John went to Sears and found a TV for $100.

Figure 2. Basic form of a conceptual dependency graph.

Figure 3. Representation of "John gave Mary a book."

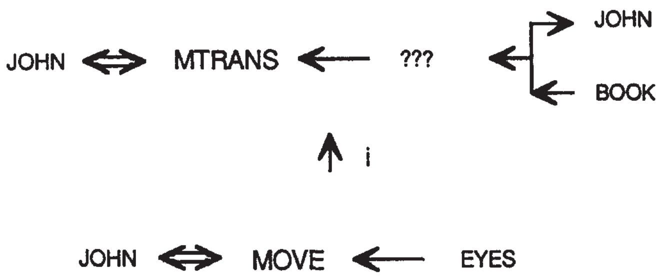

Figure 4. Representation of the act of reading.

Most readers infer that John bought the TV and paid $100 for it. This implies a rather complex chain of events, consisting of John going into the store, looking around, seeing a TV on sale for $100, taking it to the cashier, and paying $100 to the cashier. This causal chain might be represented as shown in Figure 7.

2.2. Conceptual Dependency and Semantic Nets

A distinction has been made in the literature between *content* and *structure* theories. In some ways, it is ironic that conceptual dependency and its derivatives are often grouped into the family of semantic net representations, because in general, semantic nets are a structure theory, whereas CD is a content theory.

The distinction between these two types of theories lies in their emphasis. Semantic net theory is about how knowledge should be organized: there will be nodes, with arcs connecting the nodes together. There is also some general (although, in the case of semantic nets, rarely formalized) notion of the structure's semantics; i.e., how to interpret a particular semantic net structure. Finally, there is usually the general notion of inheritence, etc. This is all structural information. That is, it says nothing about *what* will be represented, it simply says something about the (structural) form that the representation will take. Putting this another way, semantic nets tell us to use nodes and arcs, but they don't tell us what labels to put on the nodes, or what arcs to use and where. It is up to the user to decide these details. Without the details, however, semantic nets do not represent anything.

Conceptual dependency theory, on the other hand, was an attempt to enumerate the types of nodes and arcs which could be used to build representations. Rather than specifying the structure of representations, CD theory specified the *content*. True, the conventions used in drawing conceptual dependency graphs also specified structure, but this was not really the essence of the theory. The essence was the primitives, and the *names* of the dependencies which could be used to hook primitives together.

This distinction can be made clearer if one imagines trying to implement conceptual dependencies and semantic nets in first order predicate calculus. In the case of conceptual dependencies, it is not hard to imagine: the primitive acts, dependencies, and states would specify a set of predicates to use when writing predicate calculus statements to represent sentences.[1] We might have

[1]The sort of inferences proposed by those who have used CD are not typically of an exclusively deductive nature, as would be true in the predicate calculus, but this really has more to do with the *use* of the representations, rather than the representations themselves.

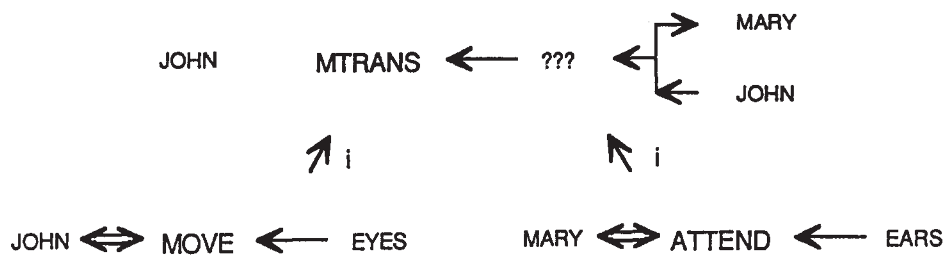

Figure 5. Representation of the act of talking.

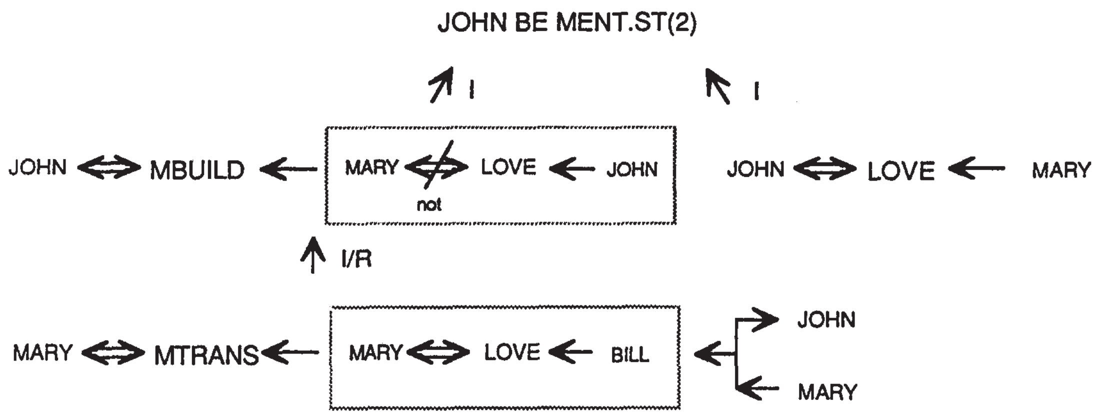

Figure 6. Representation of "John cried because Mary said she loved Bill."

to adopt some translation conventions in order to make all assertions first-order, but these would be quite straightforward. On the other hand, it does not make as much sense to "implement" semantic nets in predicate calculus. The two representations are in competition with each other: each provides a different syntax for distinguishing between predicates, arguments, and relations. There would be nothing left of semantic nets if we translated them to predicate calculus. Putting this another way, semantic nets would not add anything to first-order predicate calculus. This is in contrast to conceptual dependency, which adds the CD primitives as recommended predicates to be used.

2.3. Conceptual Dependency and Inferences

As we stated earlier, one motivation for representing the meaning of a text in a canonical form is to facilitate inferencing. Canonical form allows us to write inference rules as generally as possible: if representations did not always capture similarities in meaning, then rules about what can be inferred from a text would need to be duplicated, one rule required for every form of representation which had the same meaning relevant to the inference.

To illustrate that inferencing was facilitated by conceptual dependency, Rieger wrote the MEMORY program, which made inferences from text [4]. Inferences were used to build a "causal chain" to connect events in a story. For example:

> John hit Mary. Mary's mother took Mary to the hospital. Mary's mother called John's mother. John's mother spanked John.

MEMORY made several inferences from this story, including that John's mother spanked John because she was angry at him for hitting Mary, and that Mary went to the hospital because she was hurt.

Inferences were made in MEMORY by a set of rules, which were organized around inference categories. There were a total of 16 of these categories, some of which were:

1. Specification inferences, which filled in missing "slots" in a CD primitive, such as the ACTOR or INSTRUMENT of an action.

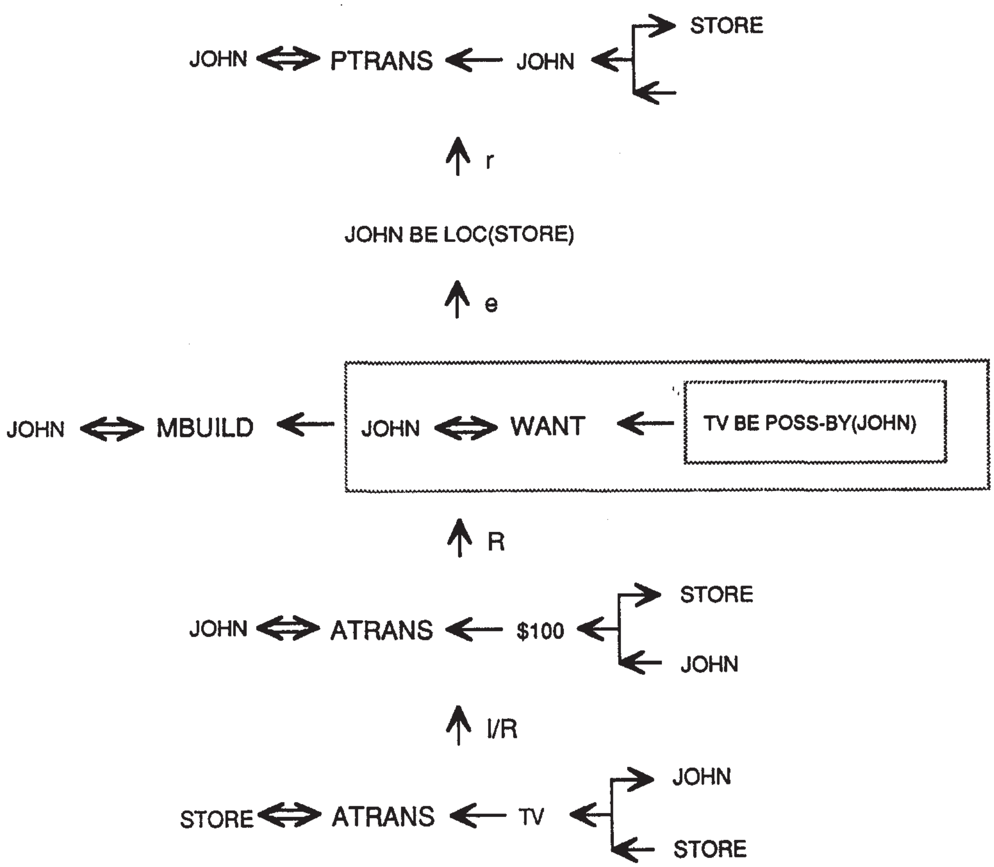

Figure 7. Representation of "John went to Sears and found a TV for $100."

2. Causative inferences, which hypothesized possible causes or preconditions of actions.
3. Resultative inferences, which inferred likely results of actions.
4. Function inferences, which inferred likely functions of objects.

Inference rules were applied in an *undirected* fashion. That is, when MEMORY read a sentence, its inference rules were automatically applied, without any particular goal in mind, such as building a causal chain to connect events together. This bridging often happened, but when it did it was fortuitous in some sense: inferences were applied to a new representation in an undirected fashion, sometimes resulting in the *confirmation* of another representation.

The undirectedness of MEMORY's inferences was meant to reflect the spontaneous nature of inferences that people make. These inferences seem uncontrollable for people: one cannot learn a new fact without inferring things about it. However, this undirected behavior led to problems: when processing a story, MEMORY did not know which inferences were most likely to lead to building a coherent causal chain to represent the story. This led to a combinatorial explosion in the number of inferences that the system had to consider in order to build causal chains. In some sense, Rieger's system was lacking common sense knowledge about what inferences were most likely to be relevant in a situation. For example:

John picked up the menu. He decided on the fish.

There are many conceivable inferences that can be made from someone picking up an object. People pick objects up for many reasons, as is illustrated by the following examples:

Use-as-instrument: John picked up the menu. He swatted a fly.

Subgoal-to-move: John picked up the menu. He found his fork.

Subgoal-to-read: John picked up the menu. He read it.