Semantic Networks and the Generation of Context

by

John Mylopoulos, Philip Cohen, Alexander Borgida and Laszlo Sugar

Department of Computer Science
University of Toronto
Toronto, Canada
416-928-5180

Abstract

This paper outlines a representation of knowledge based on semantic networks and organized in terms of semantic axes and "scenarios". A context mechanism for such nets is also discussed, along with an algorithm for integrating new information to a semantic net. Finally, an example is given to illustrate the representation and the context mechanism.

0. Introduction

The representation of knowledge is one of the most important problems of AI today. Early approaches to AI attempted to simplify the problems tackled by proposing syntactic solutions to inherently semantic problems, thus understating the importance of the representation of knowledge. The history of AI has taught us, however, that there are no shortcuts to the process of "understanding". This process can only be accomplished by a system that possesses knowledge about the topic it is expected to understand. Moreover, the degree of understanding is proportional to the degree to which this knowledge can be used for recognition and prediction. The importance of the representation of knowledge stems from these premises.

In this paper we outline a model of semantic representation that includes a semantic net(work) and a context mechanism, and illustrate its use with several examples.

Our semantic net is organized around the concept of a "scenario", i.e. a semantic unit that may encompass more than just isolated propositions. This is in agreement with current trends in this area of research (Minsky [1], Schank [2], Abelson [3] etc.)* Scenarios as well as more elementary semantic entities are organized on the semantic net along two axes we have called the SUB and DEF axes. The SUB axis has been used elsewhere (Norman and Rumelhart [4], Martin I 5]) and a form of the DEF axis has also been used in [4I.

Given a semantic net, we consider the problem of defining and using a context mechanism for referent determination, (semantic) disambiguation and inference. The problem of integrating new information to the semantic net is also treated through a process we call "graph fitting" and the relationship of context and graph fitting is discussed. Finally, a long example is presented to illustrate both the suitability of the representation and the use of the context mechanism.

The knowledge we have represented and used for most of the examples is derived from the "educational world", i.e. the world of students, courses and grades, applications to universities and decisions by universities regarding admission, etc. We consider the educational world an example of an institutional world and regard further work on modelling particular institutional worlds quite important since they subsume many situations which arise in daily life, e.g. business. With the exception of work by Martin [5] and Malhotra [6 , we are not aware of other major efforts to represent and use knowledge that can be classified as institutional.

We must emphasize from the beginning that this work has as goal the creation of "intelligent" computer systems rather than cognitive simulation of humans. The work was carried out as part of the TORUS (TORonto Understanding System) project whose aim is to provide natural language front ends to (relational) data base management systems. A prototype version of TORUS has been implemented, (Mylopoulos et al [7], and uses a simpler version of the semantic memory described here. The universe of discourse for the prototype is, again, the educational world.

1. World Modelling

By "world modelling" we mean here the process of representing the knowledge relevant to a topic of discourse. In this section we introduce the TORUS representation and give several examples of how knowledge about a domain of discourse can be represented.

1.1 The Semantic Network - Basic Building Blocks

The semantic network is simply a labelled directed graph, where both edges and nodes may be labelled. The labels of nodes will only be used for reference purposes and will usually be mnemonic names. The labels of edges, on the other hand, will have a number of associated semantic properties and inferences.

It must be noted that our use of the term "semantic network" is different from that of Quillian [8], Simmons [9], etc. However, we feel that the term captures the essence of the object it names and propose to continue to use it.

We recognize three basic types of nodes: concepts, events and characteristics, which are used to represent the ideas making up our universe.

Concepts are the essential constants or parameters of our world and specify objects or abstractions. There is a loose connection between concepts and a subset of the class of nouns.

Events are used to represent the actions which can occur in the world that we are modelling. Their representation ased on a case-grammar model, with which we assume the reader is familiar, cases (or roles) will be represented by case-labelled edges radiating outward from the event node and pointing to the nodes which fill the respective cases.

For example,

john <---agent---apply---dest---> department

represents an instance of the event "apply" with the agent case filled by "john" and the destination case filled by "department".

The following is the list of the cases presently in use: agent (a), affected (aff), topic (t), instrument (i), result (r), source (s), destination (d) and object (o). Their names are intended to be approximately self-explanatory.

For a complete understanding of an event, all of its cases must be known. We will not insist, however, this always be done at the time the event is first presented for integration with the semantic net and the context. Many semantic cases which are omitted from the input are filled in by default and contextually inferred cases when this input is integrated into the net.

The cases of an event can often be related to linguistic syntactic phenomena (e.g. prepositions, inflectional affixes, etc.) of a corresponding verb and this correlation can be very helpful in obtaining the mapping between actual sentences and their semantic representation.

A more important function of cases is to aid in inference and other logical operations which must be performed when isolated events are put into context. This is an important topic and will be discussed in section 2.2.

Characteristics are used to represent states and to modify concepts, events or other characteristics. For concepts, characteristics can be qualities, traits or properties which are inherent in, or ascribed to, the concept. For events, characteristics describe general properties which are less closely associated with the event than are cases and which have a more general, uniform meaning at all times (e.g. location, time, duration, etc.). A characteristic may be considered to be a binary relation, mapping elements from its domain - those nodes to which the characteristic may apply, to the range - those values which the characteristic may take. For example, SEX maps ANIMAL into the set containing MALE, FEMALE. Graphically, the characteristic is represented as a node labelled by the name of the characteristic, with a "ch" ("characterize") edge pointing to an element of the domain and a "v" ("value") edge pointing to the corresponding value:

john <----ch---- sex ----v----> male

"True" characteristics are usually natural attributes of concepts, but characteristics can also be used as abbreviations of more complicated situations, where we want to omit unnecessary detail. Thus ADDRESS may be thought of as a characteristic of a PERSON, although, if considered in more detail, it could be represented as the LOCATION of the DWELLING which the PERSON usually INHABITS. In some circumstances such abbreviations are mappings from a cross-product domain to a range and we use a WRT (with-respect-to) edge to indicate the second argument. For example, GRADE characterizes STUDENTS with respect to COURSE, producing a GRADE.VALUE:

```
john <----ch----grade----v----> a-
                  |
                  |  wrt
                  |
                  v
               csc.248
```

In addition to these types of entities, we will sometimes use mathematical predicates such as MEMBER, DIFFERENCE, etc.

The nodes that constitute the semantic net will be divided into two classes: one, relating to generic concepts, events and characteristics, describes the possible or allowable state of affairs in our domain of discourse. This class we shall informally call the "upstairs" of the semantic network in contrast to the second class, t's "downstairs", where we keep the instantiations and particular occurences of various ideas. "Upstairs" nodes will have their names given in capital letters, whereas nodes "downstairs" will be in lower-case.

For example,

PHYS.OBJECT <---ch--- WEIGHT ---v---> WEIGHT.VALUE

specifies that all physical objects may have a weight, whereas

peter <----ch---- weight ----v----> 120kg

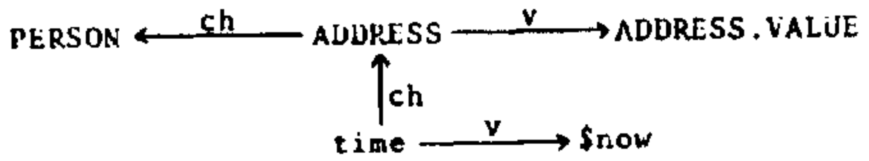
specifies that Peter's weight is 120kgs.

1.2 Scenarios

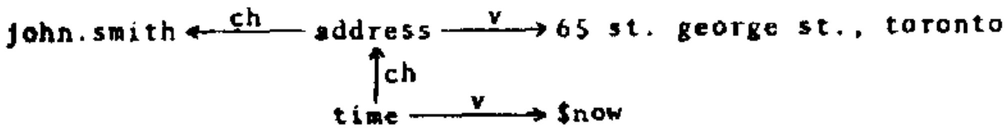
The apparatus described in the previous section is sufficient for the representation of most isolated phenomena, but we need the ability to construct larger units of knowledge which will help in inference, referent determination, prediction, etc. Such structures have been proposed in various forms by many others, including Minsky's frames [1], Schank's scripts [2], Riesbeck's story patterns [10], etc. We call our version of such structures scenarios.

A scenario is a collection of events, characteristics and mathematical predicates related through their associated edges and the concepts filling their cases, or through temporal and/or causal connectives, such as "before", "after", "same.time", "while", "prerequisite" and "effect".

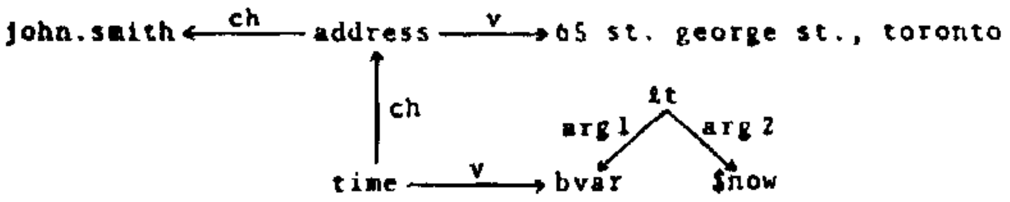
One can regard a scenario as a pattern which, when matched by a structure, enables the system to make inferences and predictions. Consider, for example, the notion of "a person's current address", which may be represented as



where $now stands for the value of the variable named "now", the current date, say. This trivial scenario will match the structure that represents the meaning of "john smith's address is 65 st. george st., toronto", shown below,

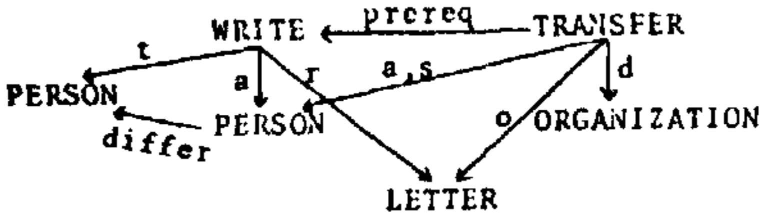


On the other hand, the structure for "a person's current address" will not match the structure that represents the meaning of "john smith's address was 65 st. george st., toronto", shown below, because the values of the "time" characteristic do not match.



Here, "lt" stands for the numerical predicate "less than", and "bvar" stands for a bound variable whose value we do not know. Clearly this is a very simple scenario since it does not allow any inferences or any (non-trivial) predictions.

Consider now a scenario for recommending a person to an organization by writing a letter:



This scenario describes a recommendation action in terms of WRITE and TRANSFER actions, the first being the prerequisite of the second. The relationship between the cases of the two actions is also defined. Note that the recommending person is both the agent of WRITE and the AGENT-SOURCE OF TRANSFER. The names of the nodes here indicate the types of instantiations that can fill the cases of this scenario. The "differ" edge specifies that a person cannot recommend himself.

The reader may wonder how this scenario is related to RECOMMEND. We will discuss overall organization of the semantic net in the next section and we only wish to mention here that this scenario may be thought of as an abstract procedure defining RECOMMEND. There may be other scenarios which define WRITE and TRANSFER in terms of other, more primitive, events.

1.3 Semantic Net Organization

So far we have discussed the entities that serve as building blocks for our representation. In this section we describe how they are put together to form the semantic net. This organization will be defined in terms of "axes" or "dimensions" and several other types of edges.

1.3.1 The SUB axis

We will say that the node X is a SUBnode of node Y, if the set of instantiations of X is a subset of the set of instantiations of Y. The SUB relation between X and Y will be denoted by **Y----SUB---->X, or simply Y--------->X. If X is** "downstairs", the relation between Y and X is one of instantiation or example-of. We will continue to use an unlabelled edge to denote such relations, since the fact that X is "downstairs" is already specified by its name. Thus

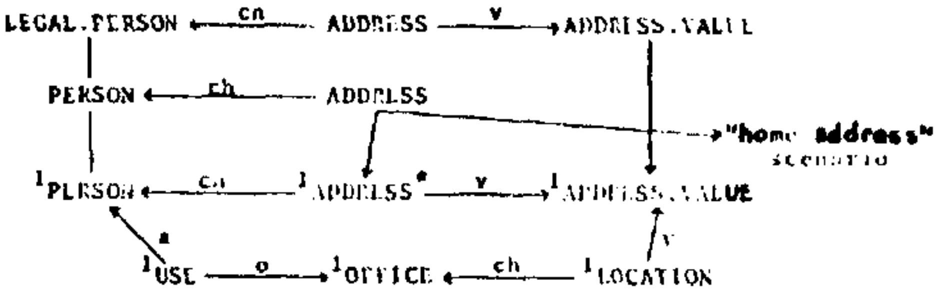
specifies that STUDENT is a subconcept of PERSON, "John.smith" is an instantiation of PERSON (i.e. a person) and "jim.brown" is a student. The SUB and instantiation edges have been used by many other representations of knowledge.

We can now organize (partially order) the concepts occuring in our domain of discourse into a hierarchy, representable by its Hasse diagram. It is important to note that (semantic) properties of concepts are inherited along the SUB axis. Thus, for example, since STUDENTS are PERSONS and PERSONS are PHYSICAL.OBJECTS, STUDENTS may have a WEIGHT associated with them. This property of the SUB axis provides us with a very important memory-saving device.

Scenarios are also organized on the SUB axis. Consider the generic characteristic for ADDRESS.

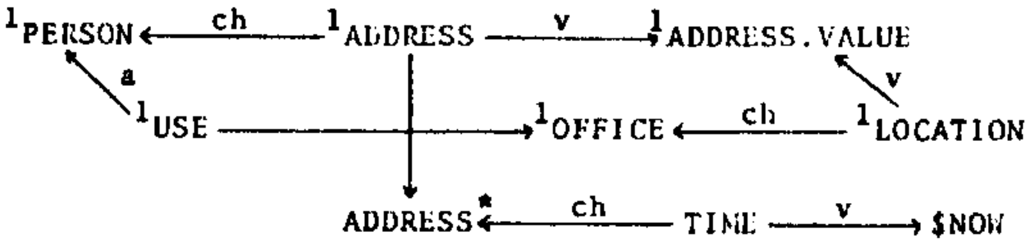
LEGAL.PERSON <---ch--- **ADDRESS**---v---> ADDRESS.VALUE

LEGAL.PERSON stands for a person or an organization (anything you can sue ...). Below it on the SUB axis we can place scenarios for office address and home address, for persons only:



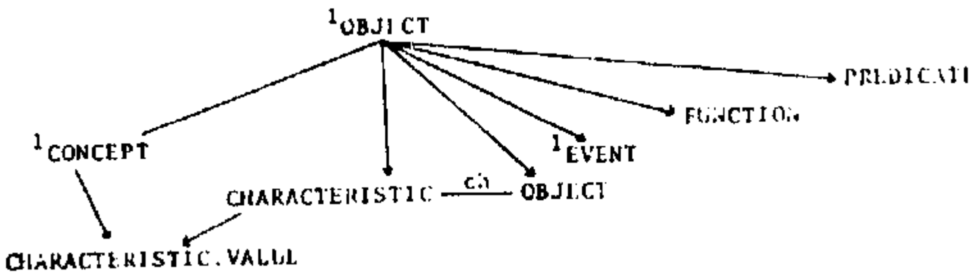
Thus a PERSON'S office ADDRESS, defined by the node marked with *, is the location value (address values are also location values in this representation) of the OFFICE USEd by the PERSON. The representation of USE and OFFICE is rather superficial, but legal, and will do to drive the point across.

We could now fit the current address below the office address scenario, in order to define the notion of "current office address"

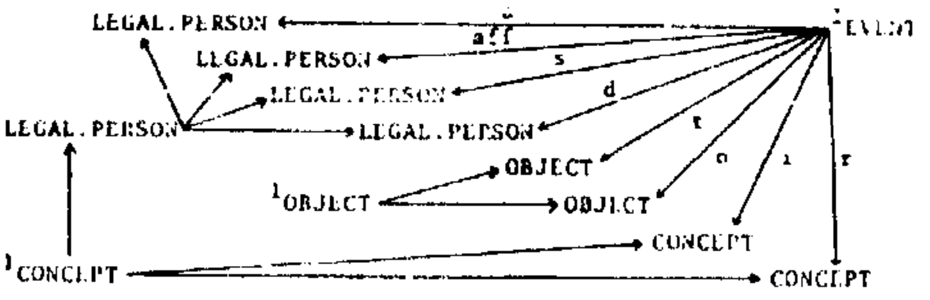


The left superscript 1's indicate the correspondence of nodes between the last two figures. Note that the ADDRESS node marked with * does not need a "ch" or "v" edge, since these are inherited from its SUPernode.

Before we proceed to discuss other axes used to organize the semantic net, it is worthwhile to mention a few things about the top of the SUB axis, Every node on the semantic net fits below the node OBJECT,



the CHARACTERISTIC and EVENT nodes being part of very general scenarios that will fit any characteristic or event



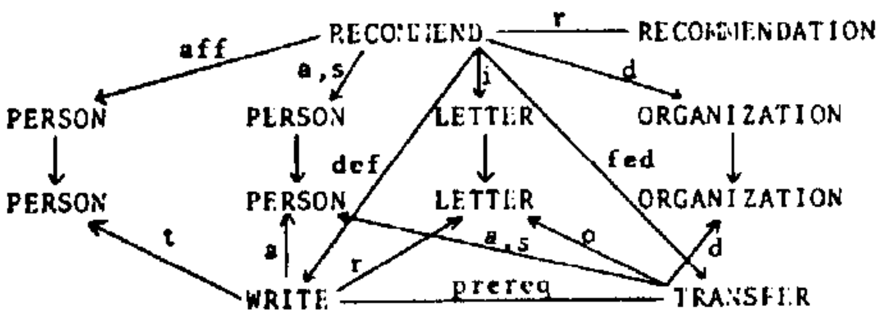
It is interesting to note that with such scenarios being present at the top of the SUB axis, only structures which fail to satisfy the syntactic rules of the semantic net will be rejected (i.e. will match no scenario on the net). Thus the structure

university.of.toronto <---ch--- color --- v---> 3kg

will match at least at the most general characteristic scenario given earlier, provided that "color" can be classified as a characteristic and 3kg as a characteristic value. This feature of the semantic net enables us to introduce scenarios that will account for (semantically) unusual structures. These scenarios may cause inferences such as "I don't understand what he's talking about", etc., and result in corresponding actions on the part of the system (question the user, etc.).

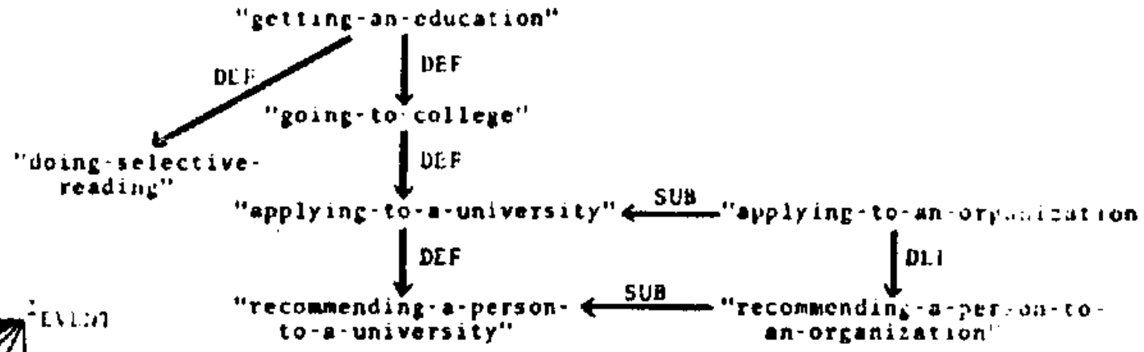
1.3.2 The DEF axis

We have already defined RECOMMEND (through a letter) in terms of a scenario. We will denote the relationship between RECOMMEND and its cases, on one hand, and the RECOMMEND scenario, on the other hand, by using "def" and "fed"-labelled edges:



There need not be just a single "def" or "fed" edge leaving an event being defined. There can be several defining scenarios for the same event and many possible entry points ("def" edges) and exit points ("fed" edges). Moreover, an event or characteristic may be defined recursively.

The RECOMMEND event of the previous diagram may be itself part of an APPLY-to-an-organization scenario, which can have as SUBscenario APPLY-to-a-university. This, in turn, may be part of a "going to college" scenario, which in turn may be part of a "getting an education" scenario. These scenarios are then related to each other along the DEF and SUB axis as follows:



At one end of the DEF axis one would find very general scenarios and plans. In the "education" world, these might involve "getting an education", "choosing a vocation", etc. activities which interact with a huge number of other general scenarios which would lead off into diverse areas. A domain of discourse, for which our representation might be adequate, would attempt to minimize its dependence upon this end of the definitional axis. One of the underlying assumptions of any attempt to model small, closed domains of discourse is that this minimization can be done.

At the other end of the DEF axis, one would find "primitives" — those entities which we choose not to represent and thus take as "given", implying that they are, somehow, understood. Unlike Schank, we believe that the primitives that would be incorporated into a net should be dependent upon the domain of discourse as well as convenience.

Finally, we should mention that another edge which defines an axis is the "part" edge (a GRADE is a "part" of a TRANSCRIPT, which is a "part" of an APPLICATION, a DEPARTMENT is a "part" of a UNIVERSITY, etc.).

1.4 Procedural Knowledge

Representing knowledge on the semantic net has the advantage that this information can be examined and reasoned about by the system. On the other hand, the representation is expensive.

and for any universe of discourse, there will always be 'peripheral' knowledge for which general reasoning may not be necessary- We represent such knowledge in terms of functions which we associate to corresponding nodes of the semantic net.

Some of these functions we will call <u>recognition functions</u> because their job is to use syntactic and semantic information in order to recognize instances of a class (e.g. address.value). <u>Mapping functions</u> will be useful for mapping structures from one level of the representation to another. For example, mapping functions may be used to replace portions of the definitional axis if it is decided that other levels, more surface or deeper ones, are not of interest. Another type of function is a <u>definitional function</u>/ which is used to define procedures for performing particular actions (EVALUATE an application, MOVE a block, etc.).

It is important to stress that knowledge can be represented, in general, in either procedural or declarative form and which form is selected is strictly an issue of trading cost for 'understanding power'.

The same issue is partly controlling another question: what 'primitives' do we use in our representation? Some primitives are important because of the inferences they generate or require. For example, in modelling the educational or any institutional world, the event of ASSIGNing values to certain characteristics is treated as a primitive because it implies that the agent of the action is socially responsible for the act and has the authority to carry it out (e.g. only a professor is allowed to assign grades to a student). Other 'primitives' are nothing more than arbitrary undefined terms (e.g. WRITE), which are not important to the particular domain of discourse for which this representation is geared. Similarly, some events will be described by scenarios in terms of other events on the semantic net (e.g. APPLY in section 2.3.1) while others are assumed to have an external defining scenario which is known to exist but is not explicitly stored (e.g. "completing an application form").

2. <u>Context and Its Uses</u>

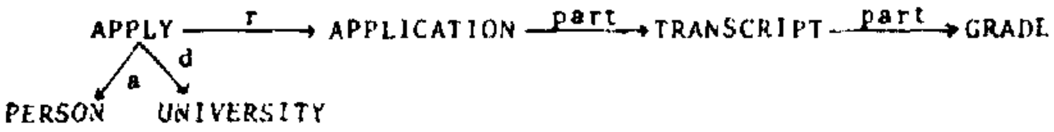
In this section, we describe how context can be organized so that immediate inferences are contextually appropriate and vice versa. The purpose of the current context facility is to maintain information relevant to the topic at hand. This information is derived from expectations and inferences and uses scenarios. The section below describes the process of generating a context, using it for integrating input and determining sentential referents. We also discuss finding the context relevant to an input by a process called "graph fitting". Finally, a detailed example is presented, serving to cover many of the salient features in the preceding discussion.

2.1 <u>Generation of Context</u>

Our study of context here is centered around inferentially generating the "foreground" (Chafe *(11)* for a given sentence. When regarding this problem from a network point of view, one sees that it is necessary to discuss the various inferences that can be made from edges in the net (cases, *'sub", "part") and scenarios (including causal and temporal connectives). Many of the inferences resemble those discussed in Rieger [12]. Since we are using a different representation than his, we have found it necessary to restate those inferences in our terminology. In addition, the use of scenarios within the net will, hopefully, guide the inferential generation of context to potentially relevant information, as opposed to brute "expansion" as done by Rieger.

The presence of a network entity in the context represents the system's expectation that this item is or will be relevant to the current dialogue. When new information, which has been predicted, enters the dialogue, its relevance can be explained by the "generation path" taken to create the expectation. We give an example of part of the context generation process below.

Consider the (partial) scenario



and the sentences

"John applied yesterday. The grades were good."

We can generate part of the context of APPLY by interpreting the "result" case and the "part" edge as described below. When APPLY occurs, it produces an APPLICATION which enters the context. When we bring APPLICATION in, we bring with it, its constituent "part"s. Thus GRADE and TRANSCRIPT are available for discussion without a previous reference, as is indicated by the definite specification in the above example. Inferences are heuristically assigned strengths which are used to limit the extent of their propagation (cf. Rieger [12 1) .

Below we list a few edge labels and discuss how context growth is affected by each.

"result": A---result---> B (read A produces B)

A is an event (scenario) and B is a concept. This edge corresponds to the output of some process (scenario). Thus, when A is instantiated, an instance of B should be created and added to the context. The propagation value of this edge, in both directions, is high, as can be seen from the APPLY example above. Inferences from "result" can occur when B is a concept which is either an "instrument" or "object" used by some other event C (more generally, an input to C). This produces a prerequisite ordering of events. Namely, A must be executed before C since A's output is input to C. This information may be redundantly discovered from the "prerequisite" edge, but the "result-input" inference is stronger since it explains why the producer is prerequisite to the consumer.

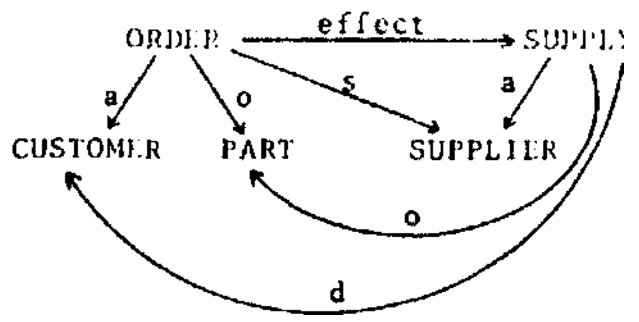
"effect": A----effect----> B (read A causes B, or B is an effect of A)

When A is instantiated, there is a strong inference that B will be relevant and thus it should be instantiated and placed in the context.

This expectation enables the system to predict part of a scenario from a piece of it. The one step prediction corresponds Sieger's resultative class.

Similarly, when B is instantiated, especially when it is within a well specified scenario, the presumption that A exists (and caused B) is strong. This class of inferences is similar to Rieger's causative class.

A particularly important inference to be made from the "effect" edge is that of social obligation. For example,



If the "effect" edge connects two events with different agents, then the agent (supplier) of the caused event (supply) is under social obligation to perform the event or scenario. We feel this usage of "effect" to be very important to institutional scenarios and it may be a distinguishing feature of these scenarios from say, physical or mental scenarios.

"prerequisite": A ---prereq---> B (read as B is a prerequisite of A)

B could be a characteristic or an event. If A is instantiated, we can confidently assert B into the context since A could not have happened unless B were satisfied (Rieger's "enabling" class). If B exists, we can predict that A should be part of the context.

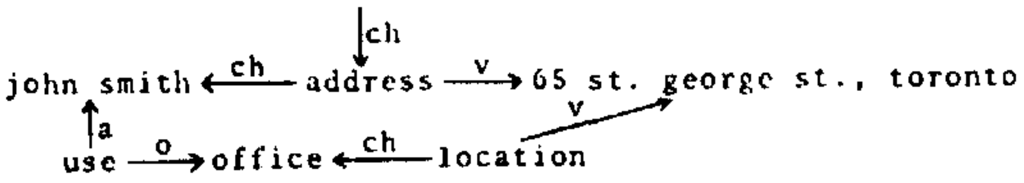
A similar analysis can be carried out for other edge labels such as "part", "agent", "characteristic", "object", etc. With all edges, propagation in the reverse direction from the target node is dependent upon the number of identically labelled edges entering that node.

We have not yet stated where these predictions should be placed with respect to the rest of the net. We view the context as a shadow data base, one which can be seen if looked for but ignored when necessary.

2.2 Graph Fitting

A part of the procedure for integrating new input to the semantic network is done by the graph fitting algorithm. This section discusses both the algorithm and its relation to context.

Consider the sentence "John Smith's office address is 65 St. George St., Toronto", which is mapped in a straightforward manner into



In attempting to integrate this to the semantic net, we could fit "address" below the generic ADDRESS node, "time" below the generic TIME node, etc. However, since we wish to isolate the most specific information in the net relevant to a given input, it is advantageous to fit the above example below the "current office address" scenario. Thus, even though "address" may originally match the generic ADDRESS node, the graph fitting algorithm will push it down to more specific scenarios until this is no longer possible. This is the essence of graph fitting.

Suppose now that the input sentence had been "John Smith's address is 65 St. George St., Toronto", where it is no longer specified whether we are talking about a home or office address. There are three choices available: (i) fit the information below all scenarios we can, thus treating "65 St. George St., Toronto" both as an office and home address, (ii) fit the new information below the first scenario found or (iii) realize the ambiguity, but not fitting the input graph to either scenario, and asking for clarification. We will adopt strategy (iii), although there is no overwhelming evidence either for it or the others.

In general, graph fitting is carried out by considering a candidate scenario, call it S, and a structure s, and by trying to match every event or characteristic in S to a corresponding entity in s. Once it has been decided that s matches S, S is added to the 'matched-scenario' list. The SUBscenarios of S are considered next and each is matched against s. All of these, which successfully match against s, then replace S on the matched-scenario list. When this step no longer produces any changes, a search is carried out "downstairs", below every scenario on the matched-scenario list, to see if (partial) instantiations of s can be found. Depending on the result of this search, either (a portion of) s is identified with already existing nodes or new ones are created and placed below all scenarios on the list.

The first scenarios to be considered for matching against s depend on the context. Thus, if SEND-a-recommendation-letter is on the shadow data base and a "send" event appears as part of s, graph fitting will first try the SEND which is part of the context rather than the generic node SEND. If this match is successful, everything proceeds as described before. If not, the system has been 'surprised' but, rather than giving up, it tries to find scenarios higher up on the net, which will match s. As explained in section 1.3.1, this search will always be successful because of the way the top of the SUB axis is organized. Of course, once a position for s has been found, a (possibly new) context is generated. Thus graph fitting and context are intimately related, although they have different functions.

Graph-fitting can be helpful in identifying the head nouns modified by restrictive relative clauses. Thus, for "The woman who supervises John suports Bill", the algorithm would first fit the relative clause for the given information, locate the appropriate woman, and proceed to fit the main clause. In the process, the algorithm would ensure that the woman in question is a professor.

2.3 An Example

We now give an example to illustrate the mechanisms described earlier. The example consists of four sentences that are consecutively presented to the system and must be "understood". The sentences are:

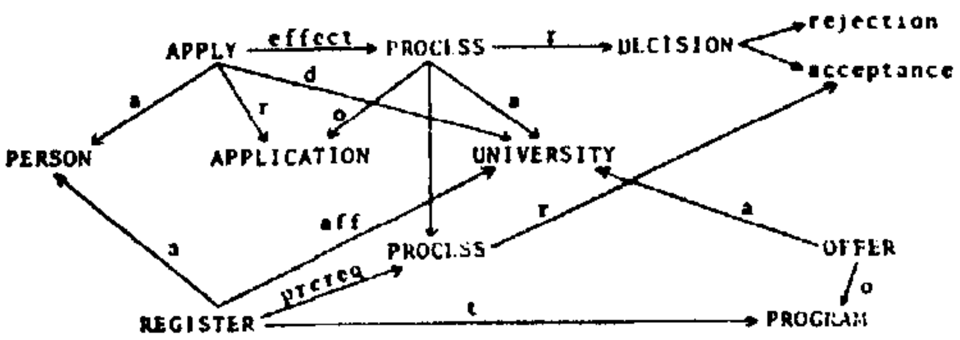
(1) "John applied to the University of Toronto"
(2) "He sent his application last Friday"
(3) "Professor Jones received it on Monday"
(4) "John received his acceptance on Thursday"

2.3.1 Scenarios of the educational world

The scenarios to be used for the analysis of the five sentences will now be described.

Scenario A - "ENTER-a-university"

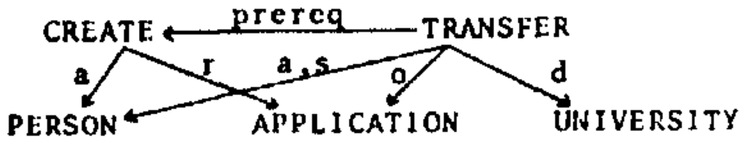
In order to enter a university one must apply for admission into one of the programs offered by the university. Once this is done, the university processes the application and if it finds it acceptable, the applicant is eligible to register for the program that he applied for.



This scenario includes a more generic PROCESS event where the "result" may be "acceptance" or "rejection", and a SUBevent where the "result" case is only "acceptance". This representation accounts for the conditional in the scenario (if the university accepts him, he can register).

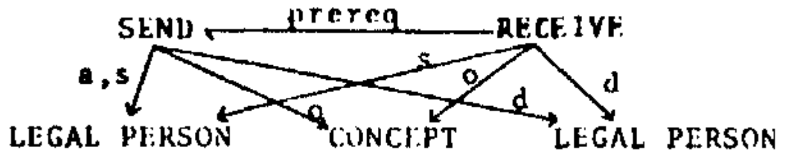
Scenario B - "APPLY~to-a-university"

In order for a PERSON to APPLY to a UNIVERSITY, with the result being an APPLICATION, he must prepare the application and have it transferred to the UNIVERSITY.



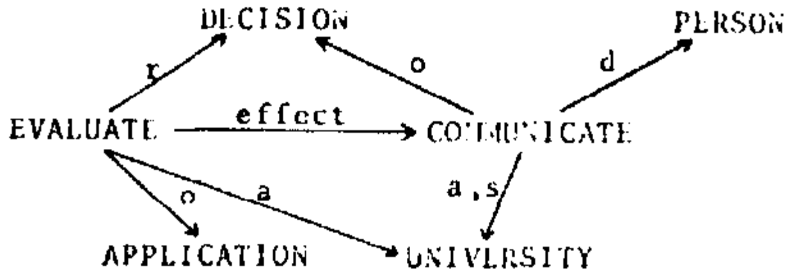
Scenario C - "TRANSFER"

A LEGAL.PERSON TRANSFERs an object to another LEGAL.PERSON by SENDing the object and the other LEGAL.PERSON RECEIVing the object.



Scenario D - "PROCESS-an-application"

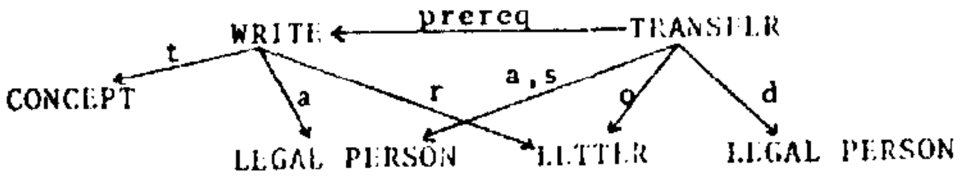
To process an APPLICATION, a UNIVERSITY must EVALUATE it and then COMMUNICATE the DECISION



Note that the "destination" case of COMMUNICATE is a global variable here, to be determined by the surrounding context.

Scenario E - "COMMUNICATE"

A LEGAL.PERSON COMMUNICATEs a CONCEPT to a LEGAL.PERSON by WRITING a LETTER about the CONCEPT and TRANSFERRING to the other LEGAL.PERSON.

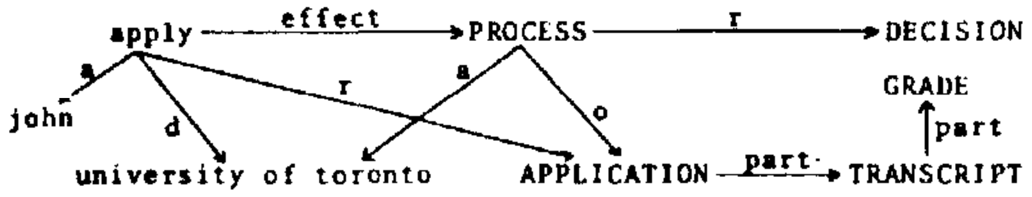


2.3.2 Analyzing the sentences

It will be assumed that the system does expect the topic of discourse to be about students, applications to universities etc., so the scenario A will become immediately relevant when the first sentence is presented for analysis.

Sentence 1: "John applied to the University of Toronto"

The sentence is fitted below the APPLY event of scenario A. The relevance of scenario A causes a number of nodes to be added to the shadow data base (see section 2.1).



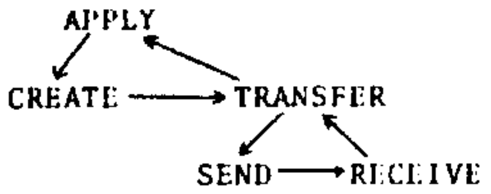
Nodes in small letters indicate here instantiated objects of the context, whereas those in capital letters indicate objects that appear to be relevant but have not been instantiated yet.

Sentence 2: "He sent the application last Friday"

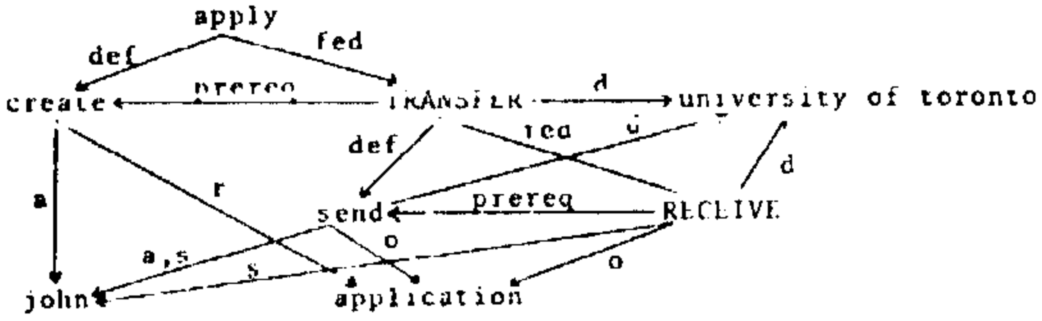
We will ignore the temporal component of the meaning of this sentence.

First of all, there are two problems of reference that must be resolved here. "He" can be associated with "John" easily enough since the latter is the only candidate which meets the selectional restrictions for "He". "The application" is resolved by using the shadow data base generated so far.

It must now be determined where "send an application" fits and how it is related to the current context. Since no SEND appears on scenario A, a search is carried out above and below APPLY on the DEF axis, to locate an occurence of SEND which is relevant to the current context. This search succeeds wo levels below APPLY



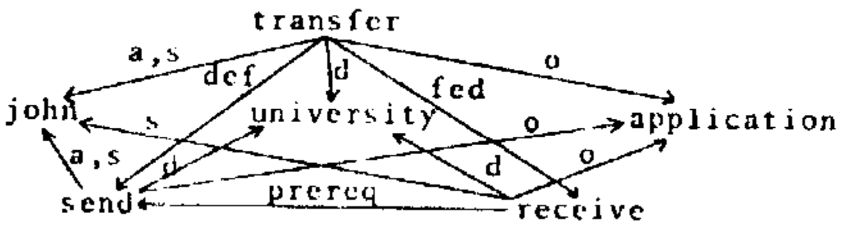
and we can therefore conclude that scenarios B and C are also relevant at this point. The expanded shadow data base includes the nodes shown earlier plus :



Note that "application" has now been instantiated along with "create", which is a prerequisite for TRANSFER, and therefore "send".

Sentence 3: "Professor Jones received it on Monday"

There is a RECEIVE node that is part of the context and it binds "it" to "applicant", as it should. Unfortunately, it has as "source" the "university of toronto", whereas the source of the incoming "receive" is "professor jones". To resolve this problem, we must acertain that "professor jones" is employed by the University, has the authority to review applications and is in the right department. We assume that these checks can be carried out and they succeed, so that the RECEIVE node of the shadow data base can be instantiated.



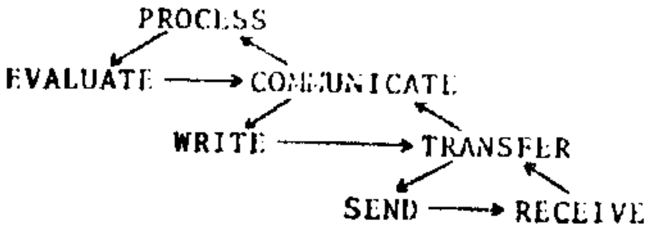
Note that "transfer" has also been instantiated since its scenario has been instantiated completely. The event "assign.authority" is used as a simple-minded representation of the fact that Professor Jones has the authority to handle (and process) applications.

Sentence 4: "John received his acceptance on Thursday"

Again, ignoring the temporal modifier, the structure that we start with and which must be integrated to the semantic net is

john <---d--- receive ---o---> acceptance

"Acceptance" is an instantiation of a node already on the shadow data base and this is used to guide the search that will explain "receive" above and below PROCESS.



However, the "object" case of RECEIVE is predicted to be LETTER, not "acceptance". In attempting to explain this contradiction, we note that both entities are used in cases of WRITE, which is a candidate for the shadow data base, LETTER being the "result" and "acceptance" being the "topic" of WRITE. Thus "acceptance" is the message while LETTER is the medium, and these terms are often used interchangeably ("I received (the letter with) the message.", "I understood {the contents of) the book.". We can formalize this rule by saying that entities which serve as "topic" and "result" of an event involving transmission of a message may appear interchangeably in a surface structure. This enables the system to accept the RECEIVE it has found and to instantiate PROCESS, EVALUATE, COMMUNICATE, WRITE, TRANSFER, SEND and RECEIVE in the shadow data base because of the "fed" and "prereq" edges that connect them to the instantiated receive.

Although not shown, the resulting shadow data base is fairly large by now and includes several items which were inferred and/or predicted because of the scenarios that are available to the system.

3. Concluding Remarks

We have extended standard semantic network methodology to include scenarios in order to deal with problems of inference and context. Along with the declarative knowledge stored on a semantic net, we have sketched classes of procedural knowledge that can be used to represent peripheral knowledge, to map from one level of the representation to another, and to execute imperatives where tracing through scenarios would be too expensive. Like other workers proposing similar structures, we consider scenarios as organized chunks of knowledge which model commonplace occurences in our educational world.

It is apparent that more institutional worlds will have to be modelled before we can detect commonalities and can make definitive statements on the kinds of primitives needed to model such worlds. Another very important problem we have not discussed at all in this paper is that of switching topics. Any reasonable language understanding system must have a model (scenario?) of how, when and why the topic under discussion switches. Moreover, additional work is needed to determine the amount of scenario information which must be brought into a context when an event in the net is instantiated. In addition to these problems, we have not dealt with exceptions to scenarios, i.e., what happens when things don't occur as they are supposed to.

Among all these and other open problems we did not mention above, we consider the problem of modelling several institutional worlds and that of modelling the conversation process, in addition to knowledge about which to converse, of paramount importance. We intend to work on these problems in the future.

Acknowledgements

We would like to thank Nick Roussopoulos, Harry Wong, John Tsotsos and Corot Reason for several useful discussions that have helped us formulate the ideas expressed in this paper.

The research described in this paper was partially supported by the Department of Communications of Canada and by the National Research Council of Canada.

References

1. Minsky, M. "A Framework for the Representation of Knowledge", in Psychology of Vision, Winston, P. (Ed.), McGraw Hill, 1975.
2. Schank, R. and Abelson, R. "Scripts, Plans and Knowledge", Proceedings 4IJCAI.
3. Abelson, R. "The Structure of Belief Systems", in Computer Models of Thought and Language, Schank and Colby (Eds.), W.H. Freeman and Co., 1973, pp. 287-339.
4. Norman, D. and Rumelhart, D. Explorations in Cognition, W.H. Freeman, 1975.
5. Martin, W. Automatic Programming Group Memos 6, 8, 11, 12, 13; also OWL Memos 1, 2, 3, 4, MIT.
6. Malhotra, A. "Design Criteria for Knowledge-Based English Language Systems for Management: An Experimental Analysis", Ph.D. thesis, Project MAC, MIT, 1975.
7. Mylopoulos, J., Borgida, A. Cohen, P., Roussopoulos, N., Tsotsos, J., Wong, H. "TORUS—A Natural Language Understanding System for Data Management", Proceedings 4IJCAI.
8. Quillian, R. "The Teachable Language Comprehender", CACM, 12 (8), 1969, pp. 459-476.
9. Simmons, R. "Semantic Networks: Their Computation and Use for Understanding English Sentences", in Computer Models of Thought and Language.
10. Riesbeck, C. "Computational Understanding Analysis of Sentences and Context", Ph.D. Thesis, Stanford University, 1974.
11. Chafe, W. "Discourse Structures and Human Knowledge", in Language Comprehension and the Acquisition of Knowledge, Freedle and Caroll (Eds.), Winston, 1972.
12. Rieger, c "Conceptual Memory: A Theory and Computer Program for Processing the Meaning Content of Natural Language Utterances", Ph.D. thesis, Stanford AI Memo 233, Stanford University, 1974.