USING KNOWLEDGE TO UNDERSTAND

Roger C. Schank
Yale University
New Haven CT

Minsky's frames paper has created quite a stir within AI but it is not entirely clear that any given researcher who would agree that the frames approach is correct would agree with any other researcher's conception of what exactly that meant. What is a frame anyway?

It has been apparent to researchers within the domain of natural language understanding for some time that the eventual limit to our solution of that problem would be our ability to characterize world knowledge. In order to build a real understanding system it will be necessary to organize the knowledge that facilitates understanding. We view the process of understanding as the fitting in of new information into a previously organized view of the world. Thus we would extend our previous view of language analysis (Schank [1973] and Riesbeck [1974]) to the problem of understanding in general. That is, a language processor is bottom up until it gets enough information to enable it to make predictions and become top down. Input sentences (like input words in intra-sentence analysis) set up expectations about what is likely to follow in the text. These expectations arise from the world knowledge that pertains to a given situation, and it is these expectations that we wish to explore here.

We choose to call our version of frames, SCRIPTS. The concept of a script, as we shall use it here, is a structure that is made up of slots and requirements on what can fill those slots. The structure is an interconnected whole, and what is in one slot affects what can be in another. The entire structure is a unit that describes a situation as a whole and makes sense to the user of that script, in this case the language understander.

A script is a predetermined sequence of actions that define a situation. Scripts are responsible for, and can be recognized by, the fact that they allow for references to objects within them just as if that object had been mentioned before. That is, certain objects within a script may be referenced by 'the' because the script itself has implicitly introduced them.

Some examples:

I. John went into the restaurant.
He ordered a hamburger, but he found it tasteless.
He asked the waitress to yell at the chef for him.

II. John got in his car.
When he put the key in, he didn't hear a thing.
He called the garage.

In these paragraphs, what we are calling scripts play a major role. We have discussed previously (Schank [1974]) how paragraphs are represented as causal chains in memory. This work implies that whenever a story is understood, inferences must be made that will connect up each input conceptualizaton to those that relate to it in the story. This connecting up process is difficult and dependent upon the making of inferences to tie together seemingly unrelated pieces of text. However, it is a process that can be facilitated tremendously by the use of scripts.

We define a script as a predetermined causal chain of conceptualizations that describe the normal sequence of things in a familiar situation. Thus there is a restaurant script, a birthday-party script, a football game script, a classroom script, and so on. Each script has in it a minimum number of players and objects that assume certain roles within the script. A script is written from the point of view of a player in one of these roles. Different scripts are defined when different roles are used as the focus of a situation.

The following is a sketch of a script for a restaurant from the point of view of the customer:

```
script: restaurant
roles:  customer; waitress; chef; cashier
reason: to get food so as to go down in
        hunger and up in pleasure

scene 1 entering
     PTRANS - go into restaurant
     MBUILD - find table
     PTRANS - go to table
     MOVE -   sit down


scene 2 ordering
     ATRANS - receive menu
     ATTEND - look at it
     MBUILD - decide on order
     MTRANS - tell order to waitress


scene 3 eating
     ATRANS - receive food
     INGEST - eat food


scene 4 exiting
     MTRANS - ask for check
     ATRANS - give tip to waitress
     PTRANS - go to cashier
     ATRANS - give money to cashier
     PTRANS - go out of restaurant
```

In this script, each primitive action given stands for the most important element in a standard set of actions. The instruments for performing each action might vary with the circumstances, as might the whole act itself. For example, in scene 1, the problem of finding a table might be handled by a maitre d', and if the restaurant is fancy enough this might require an ATRANS of money. These variables aside, the above script expresses the

eneral flow of events.

Within each act sequence, the principle f causal chaining (see Schank [1973]) is sed. That is, each action results in onditions that enable the next to occur. ew information that is received from the nalysis of a text is interpreted in terms f its place within one of the causal chains ithin the script.

Thus in paragraph I, the first sentence xemplifies the first action in scene 1. entence 2 refers to the last line of scene and the last line of scene 3. In addition t provides information about the result of he INGEST in scene 3. The third sentence oes not fit anywhere in the script, but ather is part of a subscript that defines omplaining behavior. (Such a subscript can e called by certain scripts that deal with ervices rovided by an organization.) The inal representation of paragraph I would ontain the entire restaurant script, filled n with what was specifically stated and ith assumptions about what must of been rue also included (that he sat down, for xample). In addition there would be a omplaining script attached to the entire escription at the appropriate point in the inal representation.

The general form for a script then is a et of paths that conjoin at certain crucial oints. These crucial points serve to efine the script. The paths of a script re the possibilities that are extant in a ituation.

A script is made up of a number of istinct parts. In order to know when a cript is appropriate a set of script eaders are necessary. These headers define he circumstances under which a script is alled into play. Certain key words serve his purpose together with a range of ontexts in which those words may or may not it. The headers for the restaurant script re the words restaurant, diner, out to eat, nd so on when mentioned in the context of a lan of action for getting fed. States such s hunger can call up the restaurant script s well. Obviously contexts must be estricted so as to not call the restaurant cript for sentences which use the word estaurant only as a place (i.e., "Fuel oil as delivered to the restaurant.")

Situational scripts have crucial parts hich can be said to define them. For estaurants the crucial parts are the INGEST nd the ATRANS of money. All other parts ave alternatives that allow for certain aths within the script to be followed while thers are ignored. Thus, ordering may be one by MTRANSing to a waiter or by electing and taking what you like (as in a afeteria). Likewise the ATRANSing may be one by going to the cashier or paying the aitress, or saying "put it on my bill". hese variations indicate that a situational cript is not a simple list of events, but ather a linked causal chain that can branch nto multiple possible paths. These paths ome together again at crucial defining arts of the script.

We believe that the nature of human memory is episodic. By that we mean that memory is organized around past sequences of action. When certain sequences happen often enough generalized situational scripts come to be associated with the words or circumstances that set them up, as their definition. People that have not had a familiarity with a given situation cannot be expected to have a script for that situation. Children learn these scripts by repeated associations with them. We learn to make sense of the world, by organizing the knowledge that we have so as to enable us to interpret new data in terms of our expectatons. These expectations have been generated, in part, by scripts. This is really no more than saying tha a person who has never been to a football game will have no script by which he can understand the events that go on there. (There is an important human ability to generalize scripts from others of course. So if he has seen other games it will help.)

Not everything one encounters in life has necessarily been seen before. On occasion we encounter novel situations in which we must create a plan or else understand somebody else's plan. Consider the following:

> John wanted to become chief supervisor at the plant. He decided to go and get some arsenic.

How are to make sense of such a paragraph? This paragraph make no use of situational words or the scripts that they denote. It would be unreasonable to posit a "want to be a supervisor" script that had all the necessary acts laid out as in our restaurant script. But, on the other hand, the situation being described is not entirely novel, either. The problem of understanding this paragraph would not be significantly different if "chief supervisor of the plant" were changed to "president of the men's club" or "king", the similarity is that there is a general goal state that is the same in each case and a generalized plan or group of plans that may potentially lead to that goal state. One possible desired goal state is POWER. The plan in memory associated with POWER is probably fairly complex. For that reason we have chosen to deal in the initial stages with a simpler world than that of general society. We have chosen bears.

Suppose you are a bear in the woods and you can talk to the other animals there and you are hungry. It is necessary to develop a plan of action that will enable you to eat. In the dullest of cases, you have always lived in the same old forest, in that forest is a bee's nest that regularly produces honey which they allow you to take. So you follow the course of action that you have used many times before and you get fed. This is a script. A script is applied whenever a course of action is laid out and need only be blindly followed in order to achieve a goal. Thus it is basically a set of knowledge associated with a given goal.

But the dullest of cases is of course not the best one to learn things from. So, now suppose that you are a bear in the woods who has not been a bear in the woods before. You have no set script to follow, all you know is what you like to eat. In that case you must develop a PLAN. In order to discuss what such a plan might look like, we must first point out that the setting down of a plan that will work is not the same as the creation of a plan. If you use a prestored plan for getting food in the woods you have cheated. You have used a script. In creating a plan we make use of some general knowledge about goals and subgoals. Such general knowledge is made up of sequences of actions that are used to obtain certain goals. Abstract entities called PLANS are names of possible combinations of action sequences (sort of mini-scripts) that will achieve a given goal.

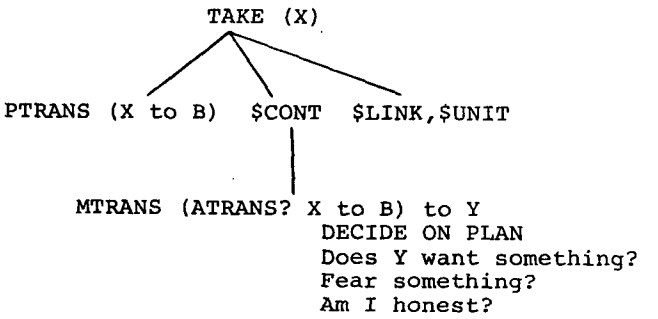
If you want to eat you must GET some food. This information is found by consulting two sources. First, the desired ACT INGEST requires 'food' as its object. Second, in order to do any ACT on any physical object, you must have that physical object in proximity. The plan to do this is called GET(X), where X is the object being sought. The plan GET(X) should tell us how to obtain the needed X in a way that uses knowledge about getting things in general before it uses knowledge about X in particular.

Once it is established that GET(X) is what we want, the problem is to translate the abstract entity GET(X) into a sequence of conceptualizatons that can actually be executed. GET(X) is simply the name of a set of subplans: FIND(X) PROX(X) and TAKE(X). FIND(X) is the name of a set of possible sequences of actions that will result in the state that will enable PROX(X) to be executed. PROX(X) stands for the possible sets of actions that get an actor where he wants to be. In order to do that an actor must know the location of X. So when FIND(X) is done the knowledge about where to go has been detemined. This knowledge enables PROX(X) which tells how to get there. Now TAKE(X) can be executed. The successful completion of TAKE(X) enables the ultimate goal INGEST(X).

The above entities are the names of PLANS. PLANS are made up of desired stages and the actions that will effect them together with the cost and circumstances surrounding the choice of a particular set at a particular time. The possible paths are called PLANBOXES. Planboxes are made up of conceptualizations that will yield desired state changes together with the preconditions that must be satisfied in order to enact the actions in those conceptualizations.

We can now examine one plan in particular. The TAKE plan is intended to enable whatever is done with an object in general, to be done at this particular time. Consequently its eventual result is potentially different if what is to be done is physical or social. On the physical level, the result is always ATRANS which is accomplished by means of a PTRANS. The enabling conditions for the ATRANS are then simply the enabling conditions for the PTRANS. In order to PTRANS something you must be physically proximate to it, so the location of the object and the taker must be identical or a PTRANS to the location of the object must have previously taken place.

The result of the ATRANS above is that a possession change exists. This will enable the final desired ACT to take place. The TAKE plan is concerned with eliminating any preconditions that might get in the way of the enabling PTRANS. The preconditions are that no one else has CONTROL of the object being sought or else that there are no concomitant bad consequences in the attempt to PTRANS to self. The TAKE plan simply calls a PTRANS if all the preconditions are positive. However if someone else CONTROLS the object, a plan for gaining CONTROL must be called. The rough outline of the TAKE plan is then as follows:



BARGAIN INFORM STEAL THREATEN TRADE ASK OVER-POWER

The theoretical constructs used here are as follows:

A DELTACT (a state preceded by a $) is a desired state change that has attached to it a set of questions. The answer to these questions determines which planbox shall be chosen (i.e. the one appropriate to the situation). A Deltact has numerous planboxes attached to it. These planboxes define the Deltact just as Inferences define a primitive ACT.

A Plan is the name of a desired action whose realization may be a simple action (A conceptualization involving a primitive ACT). However, if it is realized that some state blocks the doing of that action, the plan may be translated into a deltact to change the state that impedes the desired ACT. Thus, a Plan has attached to it a group of deltacts with tests for selecting between them. The attached Deltacts must be taken care of any time that the state they

ıange is found to be true.

A Planbox is a list of primitive ACTs ıat when performed will achieve a goal. ssociated with each primitive ACT are the et of conditions under which that ACT can e performed. Within a planbox those ɔnditions are checked. A set of conditions ıat are positive allow for the completion the desired ACT. Negative conditions ıll up new planboxes or deltacts that have s their goal the resolution of the blocking ate. Completion of the ACTs in these new anboxes remedies the state thus enabling ıe ACT that will remedy a state that will ıable an ACT and so on.

When TAKE calls up $CONT it is ecessary to select a planbox and attempt to the first ACT in the box. In order to elect a box the salient aspects of what is the set of boxes available must be nsidered. Under every Deltact we have: ıe set of questions that are relevant for ıoosing an appropriate planbox; (Choice of anbox is shown here by the number of the elevant box after a question.) Some entry ıriables that fill in the particulars in ıe generally applicable planboxes; A set of anboxes relevant to changing the desired ate. Within the planboxes are: The ACT to done; the controlled preconditions (CP) ıat must be checked to see if the ACT can done. These preconditions are found ıder the relevant ACT (i.e. in the bear rld in order to MTRANS it is necessary to ıve close physical location to the ecipient of the MTRANS). Negative CPs may fixed by remedying the bad state. The T to do is listed under negative (-) if it special; the uncontrolled preconditions P) are those which can block execution of plan but cannot be remedied. If the UPs e negative another planbox must be tried; ıe mediating preconditions (MP) are those ıich can be altered, but probably require a an of their own to change. They refer to ıe willingness of second parties to ırticipate in a plan; The result (RES) ıdicates the actions and states that will true after a plan meets its econditions.

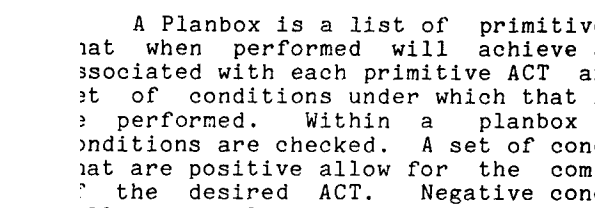
Under $CONT we have:

```
          $CONT
  MBUILD  Is W true and a good reason
                        for ATRANS? 1
          Object not valuable? 6
          Does Y value an object? 3
          Can B get Z? 3
          Does Y want something done? 2
          Can B do it? 2
          Is B honest? 4,5,1T,2T,3T
              (T=trick option)
          Is B more powerful than Y? 5
```

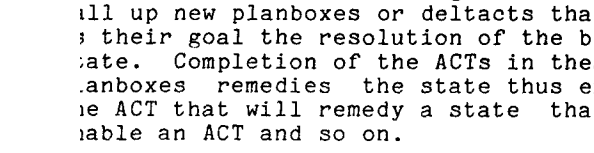
$CONT entry variables:

1: $CONT
2: ATRANS
3: Y CONT X

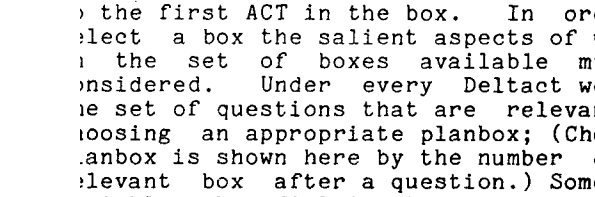
1. INFORM REASON



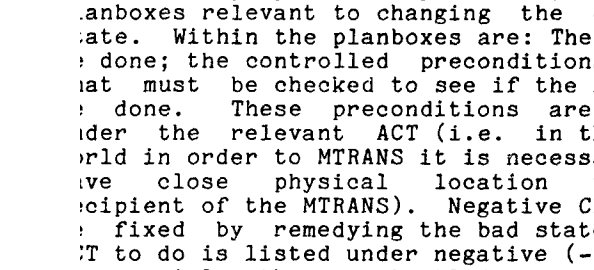
2. BARGAIN FAVOR



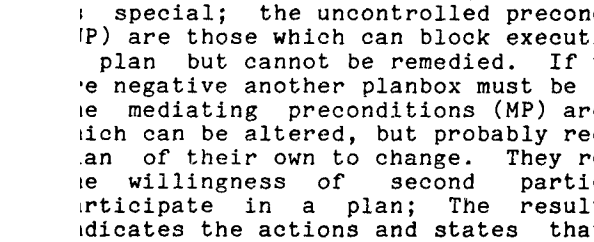
3. BARGAIN OBJECT



4. STEAL



5. THREATEN



6. ASK

7. OVERPOWER

A desired state change has connected to it a set of questions that determine the choice of planboxes. A plan box is not specific to a given state, but a predetermined collocation of them serve to define a deltact. Planboxes have three variables attached to them which are filled in by the particular deltact under which they have been selected. These are: 1: the desired state change, 2: the ACT that changes that state, 3: the previous state that now holds. The preconditions that must be satisfied are those that are true for a given primitive ACT regardless of its occurrance in a particular planbox.

A given planbox could be used in many different situations. The BARGAINOBJECT box will work for CONT, but also will work as a possible strategy under LIKE, SEXSATIATION, POWER (buying votes ?) and any other situation where someone can be convinced to do something that will help you by means of giving them something.

In fact, 1,2,3,5,6, above can be seen as part of a persuade package that will get invoked whenever one person's plan depends on the actions of another. Under FIND(X) above the deltact associated with KNOW gets called. If it is assumed that others know then the persuade package may be used as a means of getting them to tell you. Some words refer to planboxes that have been used under certain goals. Thus, "rob" is THREATEN under $CONT, and "rape" is THREATEN or OVERPOWER under SEXSATIATION.

A very small number of goals and planboxes should be necessary to define the plans that are used in the world. They should constitute a new set of primitive entities that work on top of those that underlie language directly.

Lest the problem we have been attacking get too fuzzy, it is probably time to stop and make a few points.

1. In order to understand it is necessary to have knowledge.

2. One type of knowledge is that which deals with mundane events.

3. This kind of knowledge is used for understanding what was said as well as for guiding the inference process to fill in the details in a mundane event.

4. A second type of knowledge is that which deals with behavior based on an assessment of the goals people have and knowledge that deals with paths to the attainment of those goals.

5. The inference process that is the core of understanding is not random but rather is guided by knowledge of the situation one is trying to understand.

Thus, our answer to "What is a frame?" is that a frame is a general name for a class of knowledge organizing techniques that guide and enable understanding. Two types of frames that are necessary are SCRIPTS and PLANS. Scripts and plans are used to understand and generate stories and actions, and there can be little understanding without them.

REFERENCES

Riesbeck, C. (1974) Computer Analysis of Natural Language in Context. Ph.D. Thesis, Computer Science Dept. Stanford Univ. Stanford CA.

Schank, R. (1973a) Identification of Conceptualizations Underlying Natural Language. In Schank and Colby (eds.) Computer Models of Thought and Language. San Francisco: W.H. Freeman and Company.

Schank, R. (1973b) Causality and Reasoning. Institute for Semantic and Cognitive Studies, Technical Report 1.

Schank, R. (1974) Understanding Paragraphs. Institute for Semantic and Cognitive Studies, Technical report 6.