ANSWERING PROCESS QUESTIONS

Greg W. Scragg
Institute for the Study of Semantics and Cognition
6976 Castagnola, Switzerland

Abstract

A program is described which answers questions about processes- It is assumed that such questions should be answerable using as data only procedures for performing (simulating) the actions. Thus, the same procedure that is used to simulate the making of a sandwich can be listed to tell, in English, how to make a sandwich or can be examined to predict results and side effects such as the dirtying of a knife or the consumption of cheese. A wide variety of questions is answered with very few distinct answering processes. Sample output is provided.

Introduction

Many questions ask for information about actions or processes. For example "How is a given action performed?", "Why is a substep performed?", "What would happen if a particular action were performed?", or " is it possible to do such and such?" I call such questions process questions. Although there have been many studies of question answering systems, they have been primarily concerned with questions which can be answered by table look up or simple deduction. Process questions frequently cannot be answered by such means. Although it may be feasible to associate, with each action, a list of consumables used in the execution of that action it would not be feasible to maintain lists of all possible side effects. In general, the problems of answering process questions have been little explored (but see Brown, Burton & Zdybel, 1973).

To explore the problems of answering questions which require a knowledge of processes I have developed a system which knows about kitchens and cooking. The system, LUIGI, which was developed within the MEMOD computer system at the University of California at San Diego, provides an English parser and data base management facilities. It was written in the programming language SOL*. LUIGI has two parts: a data base, called Kitchen-world, which contains information about objects found in a kitchen and a set of procedures for manipulating those objects*, a set of procedures for answering questions using the data base.

LUIGI can simulate simple actions within Kitchen-world such as cooking and preparing foods. More important, he can examine his own procedures for performing actions and calculate or predict the results of those actions**. This capability enables LUIGI to answer questions about those actions. This approach is analagous to the theory that says persons who engage in an activity are likely to be the best authorities on that activity. Although LUIGI is designed to operate within the world of the kitchen, nothing in his question-answering abilities depends upon the environment. Thus, were LUIGI moved to some other domain, say that of American History or to a sport such as baseball, once the data base were constructed, containing information about the facts and actions of the new domain, LUIGI's procedures which are now used to answer questions about the kitchen and about cooking should also apply directly to the new domain.

Preliminaries

If a program is to be able to examine its own procedures meaningfully, it must be aware of the significance of program structure. As a preliminary step, it was necessary to impose restrictions on how programs could be defined. In MEMOD all actions are encoded as part of a semantic network and are accessible to any SOL procedure. However, they are not necessarily in a form that other procedures may readily interpret. To help correct this, I have established three classes of actions: conscious, semi-conscious and unconscious.

Action classes. Conscious actions: conscious actions are those actions that can be broken down into constituent parts, each of which the performer is aware of. For example, the conscious action "MOVE x" may be defined to have these parts:

(1)
```
MOVE x:
        PICK-UP x
        TRANSPORT x
        PUT-DOWN x
```

The definition of each conscious procedure should contain meaningful subprocedures. That is, the simulation should be composed of appropriately named simulated subsets. A listing of a conscious program should be similar to a list of conscious steps done by a human in accomplishing that action. An English-like language such as SOL, coupled with an inverse parser completes the requirements for producing English-like descriptions of actions. Similarly, the instructions given to LUIGI are English-like sets of instructions reflecting the procedure a person would use to accomplish an action. Note that a side effect such as getting a spoon dirty would never be a conscious act.

Semi-conscious actions: Semi-conscious actions correspond to the lowest levels of conscious actions, those which humans are barely aware of performing. "PICK-UP" is probably a semi-conscious act. Humans are conscious of the act of picking up an object. But they need not consciously think

* Everything described herein actually runs

** LUIGI cannot actually make a sandwich but he can simulate the process. To answer certain questions, it will be necessary for LUIGI to "simulate his simulation". For convenience, I refer to LUIGI as if he could actually make a sandwich and I refer to his "simulations of simulations" simply as "simulations".

about the process by which they grasp the object with fingers and thumb. The definitions of semi-conscious actions do not need to be clear descriptions of substeps of the actions (since by definition this is probably impossible), but they must yield the proper effects when executed.

Unconscious actions: Unconscious actions are those that do not play a direct role in the simulation of human performance. These include routines for which there is no obvious human analogue, but which must be done by the program in order to perform the simulation (bookkeeping, printing, etc.)

Semi-conscious actions may not entail conscious actions and unconscious actions may entail neither conscious nor semi-conscious actions. In the definitions of procedures, conscious actions should contain as few explicit unconscious actions as possible. These should be placed in the semi-conscious actions if possible. The intent is that a semi-conscious action will correspond to a sort of primitive action for people and that unconscious actions will be invisible to some of the systems processing routines. Following these criteria, the definition of a conscious action becomes the ordered list of conscious and semi-conscious actions, resembling the list of instructions a human might be asked to follow.

Perspective. A procedure is a set of directions for performing a task. But to the question answering routines, it becomes an outline of the effects of performing the action, a plan for performing the act in the future, and a guide for determining what other acts would be performed. Thus, if LUIGI is told to perform an action, he does so by executing the appropriate procedure. If he is told an action has taken place, he notes the fact. If he needs to know about any subactions that might have occurred or of the results of the action, he must derive them from the procedure. This is an implicit statement that infprences are often not made at read time. He answers questions by examinang the procedure in varying levels of detail, either looking only at the surface level actions or recursively examining each subaction. Thus a question like

(2) How do you make cookies?

is answered by examining the program MAKE x substituting "cookies" for the formal parameter x. Questions like

(3) Can you make cookies?

can be answered by examining the preconditions for MAKE x. To answer questions like

(4) What utensils do you need to make cookies?

LUIGI must "trace" the operation of MAKE, and all of MAKE's subprocedures, noting the use of any utensil.

Inferential reasoning and procedural questions

It is not always possible to find an answer by look up or simple deduction. Yet it may still be possible to infer an answer from data in the data base. Questions about processes are often of this sort. For instance, the extraneous effects of making a sandwich are not stored anywhere, but they can be inferred by careful examination of the recipe. I now describe several classes of process questions. Each of these classes is answered by similar methods of "introspective analysis" of the system's own procedures for performing actions.

"How do you" questions. To see how LUIGI answers questions like

(5) How do you make a ham and cheese sandwich?

let us examine the format of the recipe for sandwich. A general form of the recipe, applicable to a wide variety of ingredients is entered into the system thus:

DEFINE SANDWICH AS RECIPE.

(6) The definition frame for sandwich is:

SANDWICH X.
The definition is:

PLACE A SLICE OF BREAD ON THE COUNTER.
SPREAD PREFERRED-SPREAD OF X ON THE BREAD.
PLACE EACH INGREDIENT OF X ON THE BREAD.
PLACE A SECOND PIECE OF BREAD ON THE BREAD.

Now, when LUIGI is asked question (5) , he must first determine the form of the sandwich ("ham and cheese" in this case), and then trace through the steps of (G), substituting the appropriate form of the sandwich for the variable x, and reconstructing each conscious or semi-conscious action. In this case the output would be:

IF I WERE TO MAKE A SANDWICH

(7) THEN I WOULD DO THE FOLLOWING THINGS:
I PLACE SLICE OF BREAD ON THE COUNTER.
I SPREAD MUSTARD ON THE BREAD.
I *PLACE* HAM ON THE BREAD.
I PLACE CHEESE ON THE BREAD.
I PLACE LETTUCE ON THE BREAD.
I PLACE SECOND PIECE OF BREAD ON THE BREAD.

To produce this output, a procedure called TRACE finds each conscious or semi-conscious subact in the proper sequence. TRACE is capable of recursively examining the procedure and each of its subparts. The procedure HOW runs interlaced with trace, evaluating each act node that TRACE finds. HOW must find a suitable English name for each argument to each substep and translate the node into English. To obtain the results that ham, cheese and lettuce are needed the argument X of INGREDIENT must first be converted to the node representing the concept "ham and cheese sandwich". The operator INGREDIENT is then executed, finding HAM, CHEESE and LETTUCE. This requires examining the node "sandwich", the specific node "ham and cheese sandwich" and selecting the condiment lettuce as suitable for ham and cheese. Each of these names must be substituted, one at a time, into the token of PLACE and the node described in English. Note that the procedures ingredient and preferred spread are the same procedures executed by the system as part of its efforts in actually making the sandwich.

There is some evidence that people actually go through a process something like this when they answer questions. Many persons have told me that

when asked "What ingredients do you need to make a cake?" they can say perhaps flour, milk and sugar very quickly, but to come up with baking powder, butter, eggs, etc. they must actually visualize themselves (or another) making a cake. That they actually recurse through substeps (as LUIGI does in other question answering techniques) is doubtful, largely because they probably do not have such explicit structural representations.

"Did an action occur" questions. Sometimes it is desirable to calculate the results of a past process. To answer the question

(8) Did Bill beat eggs?

LUIGI can examine each known event involving Bill to determine if it was an act of beating eggs. If there are no such events, LUIGI must recursively TRACE each recent event to see if any of its substeps might involve beating eggs. To search for a semi-conscious event all conscious and semi-conscious steps must be checked. But to find a conscious event, only the conscious substeps need to be checked.

Other classes of questions. "What", "Where", "Who" and "When" questions: Questions such as

(9) What did John do? Who made omelets?

Where did John make omelets? When did John make omelets?

can all be answered with a simple modification to the "Did" question answering routines. To answer each of the above forms LUIGI (in effect) asks the true-false ("Did") question obtained from the given question by replacing the WH-question word with the corresponding indefinite pronoun or adverb (anything, anywhere, anyone, ever). If a yes answer is returned, the description of the event is examined for the appropriate argument (action, location, subject or time) and the content of that argument is returned as the answer.

After (or before) questions: This form of question asks for the next (or previous) step of a process or for an explanation for a particular step.

(10) (a) What do you do after you put bread in the toaster?

(b) What happened before Bill made the sandwich?

The method used to answer such questions is simply an extension of the "How do you" question answering method. To answer (10a), LUIGI traces the procedure "toast" (this assumes that toasting was somehow under discussion) exactly as was done for "how" questions, except that no output is made until the desired step is reached (an act of putting bread in the toaster in this case). The next step is evaluated and printed out as in a "how" question. If the desired step is not found in the definition, TRACE must follow through each of the substeps of the action, recursively (unfortunately, in the implementation this is depth first rather than breadth first). In (10b) (which asks about a real rather than a hypothetical event) LUIGI must check each event known to have occurred rather than the procedure for the event under consideration, as in a "Did?" question.

"What if" questions: To answer questions of the form

(11) What utensils would I get dirty if I made a cake?

it is necessary to trace through the entire process of making a cake and to determine at each step if a utensil were being dirtied. The mechanism for doing this is to ask "Does making a cake get any utensil dirty?" However, the entire process must be examined: it is not possible to stop when one utensil is found as in the true-false questions.

If a process change's the state of the world (as many of them do), the prediction results might be difficult. If the output of one step of a process depends on a previous step and the question inquires about the state of the world at some particular time, it may be necessary to simulate the simulation in a hypothetical world. In such a space, LUIGI can manipulate hypothetical objects while remembering the actual locations of their real world counterparts. Thus, to answer

(12) What utensils would be left on the counter after I made cookies?

it would be necessary to simulate the entire process and then examine the hypothetical counter for utensils. The step by step method used to answer (11) would not work because a utensil might be placed on the counter and later removed, but the question asks about the end state.

"Why" questions: A "why" question could have any of several meanings. The method used to answer a "why" question should depend on the intent of the question. An expected answer to the question

(13) Why did John take a frying pan out of the cupboard?

might be the use to which he put the frying pan. To provide such an answer, LUIGI can examine the sequence of events in which the frying pan was removed just as if it were a "next step" question and return the step immediately following the removal (at the same level of recursion). Such a reply might be

(14) He fried some eggs.

In the question

(15) Why did Oliver put peanut butter on bread?

an answer which gave the event in which Oliver used the peanut butter would be sufficient. To obtain such an answer TRACE can be run just as in a "Did" question. When the event of putting the peanut butter on the bread is discovered, the name of the procedure which is being TRACEd can be returned as the answer:

(16) Oliver made a peanut butter and jelly sandwich.

To answer the question

(17) Why does water boil?

a cook might answer that water boils because (when) it is put into a pan and heated. To derive such an answer, LUIGI behaves as if he were asked a "how" question.

The determination of which form of "why" question to answer is harder than the actual answering of the question. Unfortunately, LUIGI does not do a very good job of this interpretation of input and much more work is needed.

Generated Questions. LUIGI has the ability to generate certain simple questions on his own. If he is asked a question concerning a changeable fact and the results he finds conflict with his beliefs, he can generate a question asking if any event may have changed the facts of the world. For example, if LUIGI believes the bread is in the pantry but it is actually on the counter, he cannot find it by simple means. But he can generate the question

(IB) Did anyone move the bread?

That is, he can look for changes in the desired property (location). Solution to (18) will give LUIGI a possible answer. (He will still have to verify that the bread is at the new location, however, because it may have been moved twice.) This could also apply to availability, etc. It would also be appropriate if he were confronted with strange new facts (the knife changed color) or if asked a "why" question concerning a fact previously unknown to LUIGI (why is the knife dirty?)

Control

There are two major control programs in LUIGI: the adjective compiler (described in Scragg '74) which controls the analysis of noun phrases and the comprehender.

The comprehender. Once the parser believes it has successfully parsed the sentence, the comprehender becomes responsible for the remaining processing of the sentence. It is the job of the comprehender to actually call the question answering routines. Sometimes the comprehender can modify the structure created by the parser. For example, when processing the question

(19) How do you make a tuna sandwich?

the comprehender notes that MAKE belongs to a special class of actions which refer to other actions. MAKE X is defined as

(20) FOLLOW THE RECIPE FOR X.

The comprehender must find the recipe for sandwich and replace the object of HOW (HAKE) by the recipe. Now execution of HOW will provide a list of the recipe rather than a list of the MAKE routine. The comprehender can also refuse to run the program generated by the parser. It sometimes does this when the input is a declarative sentence. There are sometimes more than one procedure for finding the answer to a given question (such as "why" questions). It is the comprehender's duty to select the answering procedure and attempt a second choice if the first fails.

Here is an example of how the comprehender might proceed in attempting to answer the question

(21) Where is the cheese?

"The cheese" is recognized as a particular entity in the world. The comprehender decides that since cheese has a preferred location in the refrigerator, the actual search should begin there. The heuristic for physical searching is called to look in the refrigerator for the cheese. Suppose it fails to find any. The comprehender then generates the question: "Did anyone move the cheese?" But suppose the event checking routine finds no references to cheese; the event analyzing routine is then called. Suppose it discovers that Bill made Welsh Rarebit. In analyzing the procedure for Rarebit, it discovers that cheese is used. The routine then concludes that there is no cheese, since it cannot be found (and Bill probably used it up). This may be the wrong answer but if so, it is a typically human error. Notice that the discovery itself provides the reason for answering. Thus in the above example, not only can it be determined that there is no cheese but also that it was used by Bill. So the answer can be returned.

(22) Bill made Welsh Rarebit.

Example

To explicate the process LUIGI must go through to answer a question and show interaction between LUIGI and various parts of the MEMOD system, I will follow, in more detail, the process by which he would answer the earlier question

(5) How do you make a ham and cheese sandwich?

The parser, working top down and left to right will eventually recognize "how" as a word which *can* begin a question. The definition of "HOW" requires that it have a propositional argument, so the parser attempts to find a proposition to fill that argument.

"Do" is interpreted as an auxiliary verb. Then "you" is recognized as a pronoun (actually a noun in SOL). The next word, "make", is an action. This signals the end of a noun phrase. MAKE's definition frame is "X MAKE Y". The arguments X and Y must be filled by the subsequent processing. The preceding noun phrase (you) is used to fill the proposed argument slot X (as SUBJ). A search then starts for the object Y. There is then a long series of steps to parse "a ham and cheese sandwich". Ultimately the structure shown in Figure (1) is passed to the comprehender. The comprehender verifies that the input was a question and records that fact. It also notices that HOW is in fact a question answering routine which can be run to produce its own answer. The comprehender knows of no alternative procedure for answering that sort of question so it invokes HOW. HOW in turn examines the procedure MAKE, from which it will produce an English transcription of the steps necessary to make the sandwich, MAKE is a high level procedure so HOW evaluates y (ham and cheese sandwich) to find the procedure which is the general recipe for sandwich (shown earlier in (6)). There is no explicit recipe for ham and cheese sandwich. The first sentence of the recipe is

(23) Put a slice of bread on the counter

which was coded into the representation shown in Figure (2a). The first step is a token of the

operator SLICE. LUIGI builds a copy of this token and executes it. The returned value (a particular slice of bread) is saved for future use. The first verb to appear in the code is *110, a token of PUT. LUIGI derives a new token of PUT from *110.

Figure 1. Input to the comprehender

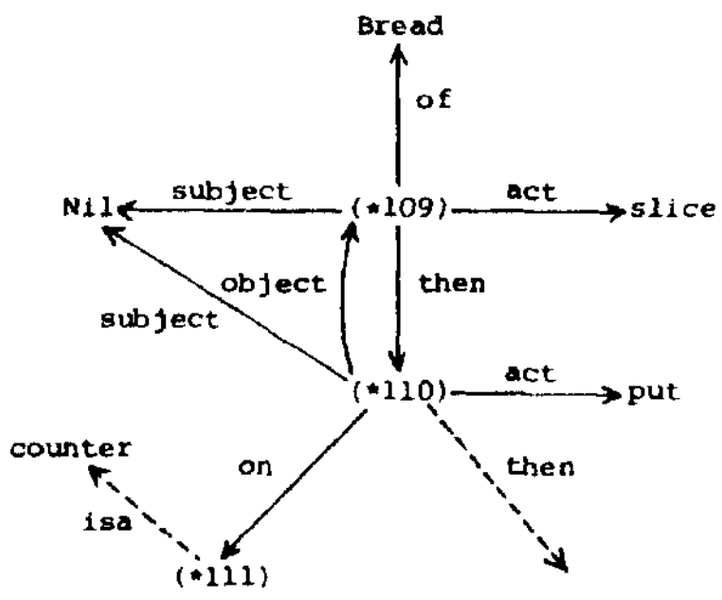

Figure 2 (a). Representation of part of a recipe. As object code.

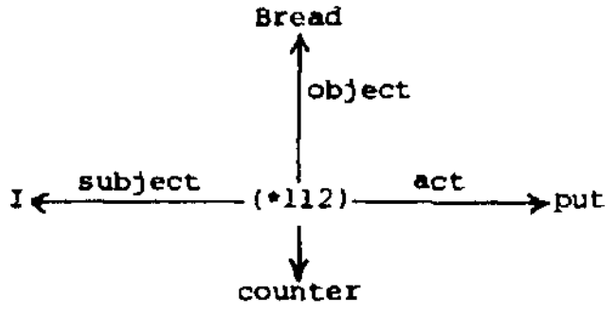

Figure 2 (b). Representation of part of a recipe. As derived code for answering a question.

The subject of this derived token is "I", obtained from the subject (you) of the question by a conversational pronoun transformation. The object of *110 is *109, the operator that was previously executed. The PRINTNAME, BREAD, of the value returned by the operator is used as the object of the new token. The name COUNTER replaces the particular counter (*111) as the location of the action. The final derived event, shown in Figure 2(b), is printed out by the primitive PRINT-TOKEN, giving

(24) I put bread on counter.

Summary

A method was presented for answering questions about processes. The method, implemented in the LUIGI system, consists of examination of the system's own simulation procedures. By simply describing his own procedures, LUIGI can answer "How" questions. "Did, who, what, where, when" and "why" questions are answered by (perhaps recursively) examining these descriptions of actions. The system can generate its own questions of some of these forms and can also simulate those actions about which it can answer questions.

Appendix

This appendix contains actual dialogue with the LUIGI system and two recipes an they were input to LUIGI. The recipes appear in upper case. The first line of each defines it as a recipe. The second is the definition frame indicating an agent, SUB, and an object X which is the item to be made. The body of the recipe follows. The dialogue was actually performed during several sessions, but was edited to appear here as one conversation. Human input is in upper and lower case, preceded by " > ". Computer output is in all upper case. Items *in* this type face are comments added later.

```
%
%                    OMELET
%
DEFINE "OMELET" AS RECIPE AT CONSCIO
OMELET (SUBJ SUB) X.
%
BREAK SOME EGGS IN A BOWL.
ADD SOME MILK TO THE EGGS.
MIX THE EGGS.
PLACE THE SKILLET ON THE STOVE.
POUR THE EGGS IN THE PAN.
FRY THE EGGS UNTIL SOLIDIFIED.
ADD SOME OF EACH INGREDIENT OF X.
ADD SOME OF EACH GARNISH OF X.
FOLD THE EGGS.
RETURN TRUE WITH THE EGGS.
# #
                    SPAGHETTI
%
DEFINE "SPAGHETTI" AS RECIPE AT CONSC
SPAGHETTI (SUBJ SUB) X.
%
BROWN SOME GROUND MEAT.
ADD SOME ONIONS TO THE GROUND MEAT.
ADD SOME MUSHROOMS.
ADD SOME SPICES.
ADD SOME TOMATO SAUCE.
SIMMER THE SAUCE.
BOIL SOME WATER.
PLACE SPAGHETTI IN THE WATER.
BOIL THE SPAGHETTI UNTIL TENDER.
DRAIN THE SPAGHETTI.
BLANCH THE SPAGHETTI.
PLACE THE SPAGHETTI IN A BOWL.
POUR THE SAUCE ON THE SPAGHETTI.
RETURN TRUE WITH THE SPAGHETTI.  #
```

```
> Don made toast,
>
>
> Julie toasted the bread.
>
> How do you make toast?
IF IT WERE THE CASE THAT I MAKE TOAST THEN I MUST
DO THE FOLLOWING:
I PUT TOAST IN TOASTER
I WAIT UNTIL DONE
I PICK-UP TOAST
I PUT TOAST ON PLATE
I SPREAD BUTTER ON TOAST
>
> How do you toast rolls?
IF IT WERE THE CASE THAT I TOAST ROLLS THEN I
MUST DO THE FOLLOWING:
I PUT ROLL IN TOASTER
I WAIT UNTIL DONE
I PICK-UP TOAST
I PUT TOAST CN PLATE
I SPREAD BUTTER ON TOAST
>
> Which pans do you use if you make spaghetti?
I USE PAN AS FOLLOWS:
FRYING-PAN
POT
```

LUIGI accepts these as statements of fact and adds the stated information to his data base.

Here LUIGI examines his recipe for toast, translating the recipe into English with the appropriate changes in variables.

LUIGI searches the spaghetti recipe. He discovered frying-pan in the definition of fry (which was necessary to brown the meat). Pot was in the definition of boil.

```
> How many utensils would I use if I made a ham
  omelet?
I KNOW OF 2
>
> How do you make a ham and cheese omelet?
IF IT WERE THE CASE THAT I MAKE OMELET THEN I
MUST DO THE FOLLOWING:
I BREAK EGG IN BOWL
I ADD MILK TO EGG
I MIX EGG
I PLACE FRYING-PAN ON STOVE
I POUR EGG IN PAN
I FRY EGG UNTIL SOLIDIFIED
I ADD HAM
I ADD CHEESE
I ADD ONION
I FOLD EGG
>
> How do you make smores?
IF IT WERE THE CASE THAT I MAKE SMORE THEN 1 MUST
DO THE FOLLOWING:
I PLACE FOIL ON COUNTER
I PLACE GRAHAM CRACKER CN FOIL
I PUT MARSHMALLOW ON GRAHAM-CRACKER
I PUT MILK-CHOCOLATE ON MARSHMALLOW
I BROIL SMORE UNTIL BROWN
>
> Make a smore.
I WILL NOW FOLLOW THE DIRECTIONS.
LET ME SEE WHERE THE FOIL IS
PERHAPS IT IS AT THE DRAWER
YES, IT IS THERE.
LET ME SEE WHERE THE GRAHAM-CRACKER IS
PERHAPS IT IS AT THE SHELVES
YES, IT IS THERE
```

These would be a fork for mixing and a spatula for folding.

```
LET ME SEE WHERE THE MARSHMALLOW IS
PERHAPS IT IS AT THE PANTRY
YES, IT IS THERE.
LET ME SEE WHERE THE MILK-CHOCOLATE IS
PERHAPS IT IS AT THE REFRIGERATOR
YES, IT IS THERE.
ALL DONE
>
> Don made a jelly omelet.
>
> Did Don put the frying-pan on the stove?
HMM, LET'S SEE.
DON MAKE OMELET
WOULD THAT DO IT?
YES
>
> Julie toasted bread.
>
> Who used the knife?
LET ME THINK. WHO'S DONE WHAT?
HMM. LET'S SEE.
JULIE TOAST BREAD
WOULD THAT DO IT?
YES
IT WAS JULIE
>
> Why did Julie use the knife?
WHEN DID SHE DO THAT?
LET ME CHECK.
JULIE TOAST BREAD
WAS THAT WHEN?
YES
JULIE SPREAD BUTTER ON TOAST
>
> Why did Julie spread butter on toast?
WHEN DID SHE DO THAT?
LET ME CHECK
JULIE TOAST BREAD
WAS THAT WHEN?
YES
JULIE TOAST BREAD
>
>
> Why did Julie toast bread?
WHEN DID SHE DO THAT?
LET ME CHECK.
JULIE TOAST BREAD
WAS THAT WHEN?
YES
BECAUSE SHE FELT LIKE IT.
>
> Did Julie toast rolls?
HMM. LET'S SEE.
JULIE TOAST BREAD
WOULD THAT DO?
NO BUT
JULIE TOAST BREAD
NO.
I DO NOT KNOW.
>
> What utensils does Don use when he makes smores?
DON USE UTENSIL AS FOLLOWS:
I DON'T KNOW OF ANY.
>
> What vegetables do you add to the
  sauce when you make spaghetti?
I ADD VEGETABLE TO SAUCE AS FOLLOWS:
```

The particular act was searched for in the recipe for omelet.

The recent history is searched for an act of knife using. Each event in the recent history is checked to see if the use of a knife constitutes a subpart

The event involving the use is returned as the answer.

There is no event which had toasting as a subpart.

Toasting rolls is a near miss The similar event is noted.

```
ONION
MUSHROOM
>
> Where are the marshmallows?
PERHAPS IT IS AT THE PANTRY
YES, IT IS THERE.
THE MARSHMALLOW IS IN THE PANTRY
> Greg made a smore.
> Greg worked on the car.
>
> Where is the foil?
PERHAPS IT IS AT THE DRAWER
NOPE, NOT THERE.
MAYBE SOMEONE MOVED IT.
HMM, LET'S SEE.
GREG WORK ON CAR
WOULD THAT DO IT?
NO.
GREG SMORE SMORE
WOULD THAT DO IT?
YES
PERHAPS IT IS AT THE COUNTER
YES, IT IS THERE.
IT IS AT THE COUNTER
>
> Where is the cheese?
PERHAPS IT IS AT THE REFRIGERATOR
NOPE, NOT THERE.
MAYBE SOMEONE MOVED IT.
HMM, LET'S SEE.
I DO NOT KNOW
I'M NOT SURE.
>
> I ate the cheese
>
> Where is the cheese?
PERHAPS IT IS AT THE REFRIGERATOR
NOPE, NOT THERE.
MAYBE SOMEONE MOVED IT.
HMM, LET'S SEE.
YOU EAT CHEESE
WOULD THAT DO IT?
YES
IT NO LONGER EXISTS.
```

The mental world location of the marshmallows is the pantry. In this case it was correct. Note LUIGI's lack of a sense of plurality.

This time the belief is incorrect. LUIGI attempts to find a reason why his belief is incorrect. Each known event is checked to see if some substep would move the foil.

REFERENCES

Brown, J.S. Old fashioned question answering re-visited. (Technical Report No. 10) Irvine, Ca. University of California, 1972.

Brown, J.S., Bell, A.G., & Zdybel, F. Steps toward an artificially intelligent CAI system: An interim report on research in progress. (BBN Report No. 2790). Cambridge: Bolt, Beranek and Newman, 1973.

Brown, J.S., Burton, R.R., & Zdybel, F. A model driven question answering system for a CAI environment. (Technical Report No. 13) Irvine, Ca. University of California, 1972.

Carbonell, J. Mixed initiative man-computer dialogues. (Technical Report No. 1970) Cambridge: Bolt, Beranek and Newman, 1971.

Charniak, E. Toward a model of children's story comprehension. (Artificial Intelligence Report AI-TR 266) Cambridge: Massachusetts Institute of Technology, 1972.

Hendrix, G.G. Question answering via canonical verbs and semantic models: A model of textual meaning. (Technical Report NL 12) Austin: University of Texas, 1973.

Kellogg, C. A natural language compiler for on line data management. American Federation of Information Processing Societies fail conference proceedings, 1968, 31, 473-492.

Norman, D.A. & Rumelhart, D.E. (Eds.) Explorations in Cognition. San Francisco: Freeman, 1975.

Schank, R.C., Goldman, N., Rieger, C.J., Riesbeck, C.K. Primitive concepts underlying verbs of thought. (Artificial Intelligence Memo AIM-162) Palo Alto: Stanford University, 1972.

Sussman, G. Teaching of procedures - Progress report. (Artificial Intelligence Memo AI-270) Cambridge: Massachusetts Institute of Technology 1912.

Scragg, G.W, Answering Questions about processes. San Diego*. University of California (Thesis), 1974. (Also appears in part in Explorations in Cognition).

Winograd, T. A program for understanding natural language. (Artificial Intelligence Report AI TR-17) Cambridge: Massachusetts Institute of Technology, 1971.