KNOWLEDGE BASED QUESTION ANSWERING

Michael J. Pazzani and Carl Engelman
The MITRE Corporation
Bedford, MA 01730

ABSTRACT

The natural language database query system incorporated in the KNOBS interactive planning system comprises a dictionary driven parser, APE-II, and script interpreter which yield a conceptual dependency conceptualization as a representation of the meaning of user input. A conceptualization pattern matching production system then determines and executes a procedure for extracting the desired information from the database. In contrast to syntax driven Q-A systems, e.g., those based on ATN parsers, APE-II is driven bottom-up by expectations associated with word meanings. The processing of a query is based on the contents of several knowledge sources including the dictionary entries (partial conceptualizations and their expectations), frames representing conceptual dependency primitives, scripts which contain stereotypical knowledge about planning tasks used to infer states enabling or resulting from actions, and two production system rule bases for the inference of implicit case fillers, and for determining the responsive database search. The goals of this approach, all of which are currently at least partially achieved, include utilizing similar representations for questions with similar meanings but widely varying surface structures, developing a powerful mechanism for the disambiguation of words with multiple meanings and the determination of pronoun referents, answering questions which require inferences to be understood, and interpreting ellipses and ungrammatical utterances.

THE SETTING

The KNOBS [Engelman, 1980] demonstration system is an experimental expert system providing consultant services to an Air Force tactical air mission planner. The KNOBS database consists of several nets of frames, implemented within an extension of FRL [Roberts, 1977], representing both individual and generic classes of targets, resources, and planned missions. The KNOBS system supports a planner by checking the consistency of plan components, enumerating or ranking possible choices for plan components, or automatically generating a complete plan. Because these activities are accomplished by means of rules and constraints expressible in English, KNOBS will hopefully be a relatively easy system to learn. For the same reasons, it is also being considered as an aid to train mission planners. The natural language subsystem of KNOBS plays several roles including those of database query, database update, command language, plan definition, and the addition or modification of production system rules representing domain knowledge. The most developed of these is database query, upon which this paper will focus.

The balance of this paper will first outline the use of conceptual dependency and mention some prior related work and then describe the several knowledge sources and the parts they play in the parsing of the input query. Finally, it will describe the method of deriving the appropriate database search and output response as well as a script-based approach to interpretting commands.

USE OF CONCEPTUAL DEPENDENCY

APE-II utilizes Conceptual Dependency theory [Schank, 1972] to represent the meaning of questions. Once the meaning of a question has been found, the question is answered by a rule based system whose tests are CD patterns and whose actions execute database queries.

We feel it is important to represent the meaning in this manner for several reasons. First, the canonical meaning representation enables questions which have different surface expressions, but the same meaning, to be answered by the same mechanism. This is not only of theoretical significance, but is also a practical matter as it requires less effort to produce a robust system.

Because people do not always say precisely what they mean, inferences may be required to explicate missing information. This inference process can also utilize the canonical meaning representation. Finally, finding the referent of a nominal which is modified by a relative clause is, in some cases, similar to question answering although the syntactic constructions used differ. As a result of this similarity, the question answering productions can also be used for determining the referents of a relative clause. The conversation with KNOBS (whose database is fictional) in Fig. 1 illustrates these points.

The first question is represented in the same manner as "Does Ramstein have F-4G's?" and would be answered by the same rule. The second question,

USER: Are there F-4G's at Ramstein?
KNOBS: RAMSTEIN has F-4Gs.

USER: Can its fighters reach the target?
KNOBS: F-15s can reach BE50301 from RAMSTEIN.
F-4Gs and F-4Cs can not reach BE50301 from RAMSTEIN.

USER: Which SCL which are carried by an F-4C contain ECM?
KNOBS: B1, B7 and B8.

Figure 1. A Question Answering Interchange within KNOBS.

after resolving the pronominal reference, requires an inference to find the location from which the F-4G's will be leaving. This inference states that if the source of the object of a physical transfer is missing, then the source could be the initial location of the object. The third question can be thought of as two questions: "Which SCL (Standard Configuration Load - a predefined weapons package) are carried by an F-4C?" and "Which of those contain ECM (Electronic Counter Measures - radar jamming equipment)?". The first part requires a script based inference: In order for an SCL to be carried by an aircraft, the aircraft must be capable of having the SCL as a part. After the first part is answered as a question, the second part is answered as a second question to discover which contain ECM.

The system of representation used for nominals (or picture producers) differs from that normally present in a CD system. Typically, an object such as an F-4C would be represented as a picture producer with a TYPE case filled by VEHICLE, a SUBTYPE case filled by aircraft, and, perhaps, a MODEL case filled by F-4C. In KNOBS, the meaning representation produced by the parser is F-4C, the name of a frame. The set membership of this frame is indicated by links to other frames. F-4C is a kind of FIGHTER which is a kind of AIRPLANE which is an AIRCRAFT which is a VEHICLE which is a PICTURE PRODUCER. We feel that representing nominals in this manner allows a finer degree of discrimination than explicitly labeled cases to denote a conceptual hierarchy.

Many of the attributes of objects in the database (which are stored as value facets of slots in FRL) are represented as kinds of RELATIONS in the KNOBS system. For example, the representation of "Hahn's Latitude" is (LATITUDE ARGUMENT (HAHN)). Note, however, that the representation of "Hahn's aircraft" is (AIRCRAFT LOC (AT PLACE (HAHN))).

PREVIOUS WORK

We would like to distinguish the KNOBS natural language facility from such familiar natural language query systems as LADDER [Hendrix, 1978] and LUNAR [Woods, 1972] in both function and method. The functional model of the above systems is that of someone with a problem to solve and a database containing information useful in its solution which he can access via a natural language interface. KNOBS, by contrast, integrates the natural language capability with multi-faceted problem solving support including critiquing and generating tactical plans. Our approach differs in method from these previous systems in its bottom-up, dictionary driven parsing which results in a canonical representation of the meaning of the query, its ability to perform context dependent inferences with this representation during question answering, and the use of a declarative representation of the domain to assist parsing, question answering, plan updating, and inferencing.

A system similar to APE-II in both its dictionary driven approach to parsing and its direct attack on word sense disambiguation is the Word Expert Parser (WEP) [Small, 1980]. This parser associates a discrimination net with each word to guide the meaning selection process. Each word in a sentence is a pointer to a coroutine called a word expert which cooperates with neighboring words to build a meaning representation of the sentences in a bottom-up, i.e., data driven, fashion. At each node in the discrimination net a multiple-choice test is executed which can query the lexical properties or expectations, (selectional restrictions [Katz, 1963]) of neighboring words, or proposed FOCUS, ACTIVITY, and DISCOURSE modules. The sense selection process of WEP requires that each word know all of the contexts in which its senses can occur. For example, to find the meaning of "pit", the pit expert can ask if a MINING-ACTIVITY, EATING-ACTION, CAR-RACING, or MUSIC-CONCERT-ACTION is active.

APE-II evolved from APE (A Parsing Experiment), a parser used by the DSAM (Distributable Script Applying Mechanism) and ACE (Academic Counseling Expert) projects at the University of Connecticut [Cullingford, 1982]. APE is based on the CA parser [Birnbaum, 1981] with the addition of a word sense disambiguation algorithm.

In CA, word definitions are represented as requests, a type of test-action pair. The test part of a request can check lexical and semantic features of neighboring words; the actions create or connect CD structures, and activate or deactivate other requests.

The method available to select the appropriate meaning of a word in CA is to use the test part of separate requests to examine the meanings of other words and to build a meaning representation as function of this local context. For example, if the object of "serve" is a food, the meaning is "bring to"; if the object is a ball, the meaning is "hit toward". This method works well for selecting a sense of a word which has expectations. However, some words have no expectations and the intended sense is the one that is expected. For example, the proper sense of "ball" in "John kicked the ball." and "John attended the ball." is the sense which the central action expects.

The word definitions of APE are also represented as requests. A special concept called a VEL is used to represent the set of possible meanings of a word. When searching for a concept which has certain semantic features, an expectation can select one or more senses from a VEL and

discard those that are not appropriate. In addition, APE can use expectations from a contextual knowledge source such as a script applier to select a word sense. Each script is augmented with parser executable expectations called named requests. For example, at a certain point in understanding a restaurant story, leaving a tip for the waiter is expected. The parser is then given a named request which could help disambiguate the words "leave" and "tip", should they appear.

APE-II

A word definition in APE-II consists of the set of all of its senses. Each sense contains a concept, i.e., a partial CD structure which expresses the meaning of this sense, and a set of conceptual and lexical expectations.

A conceptual expectation instructs the parser to look for a concept in a certain relative position which meets a selectional restriction. The expectation also contains a selectional preference, a more specific, preferred category for the expected concept (cf. [Wilks, 1972]). If such a concept is found, the expectation contains information on how it can be combined with the concept which initiated the expectation. A lexical expectation instructs the parser to look for a certain word and add a new, favored sense to it. This process is useful for predicting the function of a preposition [Reisbeck, 1976]. The definition of a pronoun utilizes a context and focus mechanism to find the set of possible referents which agree with it in number and gender. THE PRONOUN IS THEN TREATED LIKE A WORD WITH MULTIPLE SENSES. The definitions of the words "fly", "eat" and "A/C" are shown in Fig. 2.

The definition of "A/C" states that it means AIRCRAFT or AIR-CONDITIONER. APE-II uses selectional restrictions to choose the proper sense of "A/C" in the question "What A/C can fly from Hahn?". On the other hand, in the sentence "Send 4 A/C to BE70701.", APE-II utilizes the facts that the OCA script is active, and that sending aircraft to a target is a scene of that script, to determine that "A/C" means AIRCRAFT. In the question "What is an A/C?", APE-II uses a weaker argument to resolve the potential ambiguity. It utilizes the fact that AIRCRAFT is an object that can perform a role in the OCA script, while an AIR-CONDITIONER cannot.

The definition of "fly" states that it means FLY which is a kind of physical transfer. The expectations associated with fly state the actor of the sentence (i.e., a concept which precedes the action in a declarative sentence, follows "by" in a passive sentence, or appears in various places in questions, etc.) is expected to be an AIRCRAFT in which case it is the OBJECT of FLY or is expected to be a BIRD in which case it is both the ACTOR and the OBJECT of the physical transfer. This is the expectation which can select the intended sense of "A/C". If the word "to" appears, it might serve the function of indicating the filler of the TO case of FLY. The word "from" is given a similar definition, which would fill the FROM case with the object of the preposition which should be a PICTURE-PRODUCER but is preferred to be a LOCATION.

The definition of "eat" contains an expectation with a selectional preference which indicates that the object is preferred to be food. This preference serves another purpose also. The object will be converted to a food if possible. For example, if the object were "chicken" then this conversion would assert that it is a dead and cooked chicken.

We will first discuss the parsing process as if sentences could be parsed in isolation and then explain how it is augmented to account for context. The simplified parsing process consists of adding the senses of each word to an active memory, considering the expectations, and removing concepts (senses) which are not connected to other concepts.

Word sense disambiguation and the resolution of pronominal references are achieved by several mechanisms. Selectional restrictions can be helpful to resolve ambiguities. For example, many actions require an animate actor. If there are several choices for the actor, the inanimate ones will be weeded out. Conversely, if there are several choices for the main action, and the actor has been established as animate, then those actions which require an inanimate actor will be discarded. Selectional preferences are used in addition to selectional restrictions. For example, if "eat" has an object which is a pronoun whose possible referents are a food and a coin, the food will be preferred and the coin discarded as a possible referent.

A conflict resolution mechanism is invoked if more than one concept satisfies the restrictions and preferences. This consists of using "conceptual constraints" to determine if the CD structure which would be built is plausible. These constraints are predicates associated with CD primitives. For example, the locational specifier INSIDE has a constraint which states that the contents must be smaller than the container.

The disambiguation process can make use of the knowledge structures which represent stereotypical domain information. The conflict resolution algorithm also determines if the CD structure which would be built refers to a scene in an active script and prefers to build this type of conceptualization. At the end of the parse, if there is an ambiguous nominal, the possibilities are matched against the roles of the active scripts. Nominals which can be a script role are preferred.

A planned extension to the parsing algorithm consists of augmenting the definition of a word sense with information about whether it is an uncommonly used sense, and the contexts in which it could be used (see [Charniak, 1981]). Only some senses will be added to the active memory and if

```
(DEF-WORD A/C (SENSE (AIRCRAFT))
              (SENSE (AIR-CONDITIONER)))

(DEF-WORD EAT (SENSE [EAT ACTOR (NIL)
                          OBJECT (NIL)
                          TO (*INSIDE* PLACE (*STOMACH* PART (NIL]
                  EXPECTATIONS ([IF (IN-ACT-SPOT #ANIMATE)
                                    THEN ((SLOTS (TO PLACE PART)
                                                 (ACTOR]
                                [IF (IN-OBJ-SPOT *PP*)
                                    PREFER (#FOOD)
                                    THEN ((SLOTS (OBJECT]))

[DEF-WORD FLY (SENSE (FLY OBJECT (NIL)
                          ACTOR (NIL)
                          INSTRUMENT ($FLY)
                          TO (*PROX* PLACE (NIL))
                          FROM (*PROX* PLACE (NIL)))
                  EXPECTATIONS ([IF (IN-ACT-SPOT AIRCRAFT)
                                    THEN ((SLOTS (OBJECT)))
                                    ELSE (IF (IN-ACT-SPOT BIRD)
                                             THEN ((SLOTS (ACTOR) (OBJECT)]
                  LEXICAL-EXPECTATIONS ((TO (MAKE-DEF (OB-PREP *PP*)
                                                      (TO PLACE)
                                                      (*LOC*)))
                                        (FROM (MAKE-DEF (OB-PREP *PP*)
                                                        (FROM PLACE)
                                                        (*LOC*))))))
```

Figure 2. APE-II Dictionary Definitions.

none of those concepts can be connected, other senses will be added. A similar mechanism can be used for potential pronoun referents, organizing concepts according to implicit or explicit focus in addition to their location in active or open focus spaces (see [Grosz, 1977]).

Another extension to APE-II will be the incorporation of a mechanism similar to the named requests of APE. However, because the expectations of APE-II are in a declarative format, it is hoped that these requests can be generated from the causally linked scenes of the script.

QUESTION ANSWERING

After the meaning of a question has been represented, the question is answered by means of pattern-invoked rules. Typically, the pattern matching process binds variables to the major nominals in a question conceptualization. The referents of these nominals are used in executing a database query which finds the answer to the user's question. Although the question conceptualization and the answer could be used to generate a natural language response [Goldman, 1975], the current response facility merely substitutes the answer and referents in a canned response procedure associated with each question answering rule.

The question answering rules are organized according to the context in which they are appropriate, i.e., the conversational script [Lehnert, 1978], and according to the primitive of the conceptualization and the "path to the focus" of the question. The path to the focus of a question is considered to be the path of conceptual cases which leads to the subconcept in question.

A question answering production is displayed in Fig. 3. It is a default pattern designed to answer questions about which objects are at a location. This pattern is used to answer the question "What fighters do the airbases in West Germany have?". In this example, the pattern variables &LOC is bound to the meaning representation of "the airbases in West Germany" and &OBJECT is bound to the meaning representation of "fighters". The action is then executed and the referent of &OBJECT is found to be (FIGHTER) and the referent of &LOC is found to be (HAHN SEMBACH BITBURG). The fighters at each of these locations is found and the variable ANSWER is bound to the value of MAPPAIR:

((HAHN . (F-4C F-15)) (SEMBACH . NIL)
(BITBURG . (F-4C F-15))).

The response facet of the question answering production reformats the results of the action to merge locations with the same set of objects. The answer "There are none at Sembach. Hahn and Bitburg have F-4Cs and F-15s." is printed on successive iterations of PMAPC.

The production in Fig. 3 is used to answer most questions about objects at a location. It invokes a general function which finds the subset of the parts of a location which belong to a certain class. The OCA (offensive counter air) script used by the KNOBS system contains a more specific pattern for answering question about the defenses of a location. This production is used to answer the question "What SAMs are at BE70701?". The action of this production executes a procedure which finds the subset of the surface to air missiles whose range is greater than the distance to the location.

```
(DEF-Q-PAT PAT (*EXISTS* OBJECT &OBJECT
                         LOC (*PROX* PLACE &LOC))

           ACTION [MAPPAIR (FIND-REFERENTS &LOC)
                    (FUNCTION (LAMBDA (LOC)
                      (MAPCONC (FIND-REFERENTS &OBJECT)
                        (FUNCTION (LAMBDA (TYPE)
                            (FIND-OBJECTS-AT LOC TYPE]

           RESPONSE [PMAPC (MERGEPAIRS ANSWER)
                      (FUNCTION (LAMBDA (LOC ITEMS)
                        (COND ((NULL ITEMS)
                            (MSG "There are none at "
                                 (NAME LOC)
                                 "."))
                          (T (MSG (NAME LOC)
                                  (THIRD-PERSON? "have" LOC)
                                  (NAME ITEMS)
                                  "."]
           Q-FOCUS (OBJECT IS-A)]
```

Figure 3. A Question Answering Production.

In addition to executing a database query, the action of a rule can recursively invoke other question answering rules. For example, to answer the question "How many airbases have F-4C's?", a general rule converts the conceptualization of the question to that of "Which airbases have F-4C's?" and counts the result of answering the latter. The question answering rules can also be used to find the referent of complex nominals such as "the airbases which have F-4C's". The path to the focus of the "question" is indicated by the conceptual case of the relative pronoun.

INFERENCE

When important roles are not filled in a concept, "conceptual completion" inferences are required to infer the fillers of conceptual cases. Our conceptual completion inferences are expressed as rules represented and organized in a manner analogous to question answering rules. The path to the focus of a conceptual completion inference is the conceptual case which it is intended to explicate. Conceptual completion inferences are run only when necessary, i.e., when required by the pattern matcher to enable a question answering pattern (or even another inference pattern) to match successfully.

An example conceptual completion inference is illustrated in Fig. 4. It is designed to infer the missing source of a physical transfer. The pattern binds the variable &OBJECT to the filler of the OBJECT role and the action executes a function which looks at the LOCATION case of &OBJECT or checks the database for the known location of the referent of &OBJECT. This inference would not be used in processing the question "Which aircraft at Ramstein could reach the target from Hahn?" because the source has been explicitly stated. It would be used, on the other hand, in processing the question, "Which aircraft at Ramstein can reach the target?". Its effect would be to fill the FROM slot of the question conceptualization with RAMSTEIN.

```
(DEF-INFERENCE PAT (*PTRANS* OBJECT &OBJECT)
               ACTION (FIND-LOCATION &OBJECT)
               INFERENCE (FROM))
```

Figure 4: A Concept Completion Inference.

If a question answering production cannot be found to respond to a question, and the question refers to a scene in an active script, causal inferences are used to find an answerable question which can be constructed as a state or action implied by the original question. These inferences are represented by causal links [Cullingford, 1978] which connect the states and actions of a stereotypical situation. The causal links used for this type of inference are RESULT (actions can result in state changes), ENABLE (states can enable action), and RESULT-ENABLE (an action results in a state which enables an action). This last inference is so common that it is given a special link. In some cases, the intermediate state is unimportant or unknown. In addition to causal links, temporal links are also represented to reason about the sequencing of actions.

The causal inference process consists of locating a script pattern of an active script which represents the scene of the script referred to by a question. The pattern matching algorithm assures that the constants in the pattern are a super-class of the constants in the conceptual hierarchy of FRL frames. The variables in script patterns are the script roles which represent the common objects and actors of the script. The binding of script roles to subconcepts of a question conceptualization is subject to the recursive matching of patterns which indicate the common features of the roles. (This will be explained in more detail in the section on interactive script instantiation.) After the scene referenced by the user question is identified, a new question concept is constructed by substituting role bindings into patterns representing states or actions linked to the identified scene.

Two script patterns from the OCA script are illustrated in Fig. 5. The script pattern named

```
(DEF-SCRIPT-PAT NAME AC-FLY-TO-TARGET
               PAT (*PTRANS* OBJECT  &OCA:AIRCRAFT
                             TO (*PROX* PLACE &OCA:TARGET)
                             FROM (*PROX* PLACE  &OCA:AIRBASE))
               SCRIPT OCA
               AFTER AC-HIT-TARGET
               RESULT-ENABLE AC-HIT-TARGET
               RESULT AC-OVER-TARGET)

(DEF-SCRIPT-PAT NAME AC-HIT-TARGET
               PAT (*PROPEL* ACTOR &OCA:AIRCRAFT
                             TO (*LOCSPEC* PLACE &OCA:TARGET)
                             OBJECT &OCA:SCL)
               SCRIPT OCA
               RESULT TARGET-IS-DESTROYED
               AFTER AC-FLY-BACK)
```

Figure 5. Definitions of Script Patterns.

AC-FLY-TO-TARGET matches the meaning of sentences which refer to the aircraft flying to the target from an airbase. It results in the aircraft being over the target which enables the aircraft to attack the target. The script pattern AC-HIT-TARGET represents the propelling of a weapon toward the target. It results in the destruction of the target, and is followed by the aircraft flying back to the airbase.

The knowledge represented by these script patterns is needed to answer the question "What aircraft at Hahn can strike BE70701?". The answer produced by KNOBS, "F-16s can reach BE70701 from Hahn.", requires a causal inference and a concept completion inference. The first step in producing this answer is to represent the meaning of the sentence. The conceptualization produced by APE-II is shown in Fig. 6a. A search for a question answering pattern to answer this fails, so causal inferences are tried. The question concept is identified to be the AC-HIT-TARGET scene of the OCA script, and the scene which RESULT-ENABLEs it, AC-FLY-TO-TARGET is instantiated. This new question conceptualization is displayed in Fig 6b. A question answering pattern whose focus is (OBJECT IS-A) is found which could match the inferred question (Fig. 6c). To enable this pattern to match the inferred question, the FROM case must be inferred. This is accomplished by a concept completion inference which produces the complete conceptualization shown in Fig. 6d. Finally, the action and response of the question answering are executed to calculate and print an answer.

INTERACTIVE SCRIPT INSTANTIATION

The script patterns which describe the relationships among the scenes of a situation are also used by the KNOBS system to guide a conversation about that domain. The conversation with KNOBS in Fig. 7 illustrates the entering of plan components by interactively instantiating script patterns.

The first user sentence instantiates two script patterns (the flying of aircraft, and the striking of a target) and binds the script roles: TARGET to BE70501, WING to 109TFW, AIRCRAFT-NUMBER to 4, and TIME-OVER-TARGET to 0900. KNOBS asks the user to select the AIRCRAFT. Because the user replied with a question whose answer is an aircraft, KNOBS asks if the user would like would like to use that aircraft as a component of the developing plan. This is accomplished by a rule that is activated when KNOBS asks the user to specify a plan component. The interpretation of the user's negative answer is handled by a rule activated when KNOBS asks a yes-no question. KNOBS checks the consistency of the user's answer and explains a constraint which has failed. Then, the user corrects this problem, and KNOBS processes the extra information supplied by matching the meaning of the user's input to a script pattern.

```
(*PROPEL*  TO (*PROX* PLACE (BE70701))
           ACTOR (AIRCRAFT IS-A (*?*)
                           LOC (AT PLACE (HAHN)))
           OBJECT (NIL)
           MODE (*POTENTIAL*))
```

Figure 6a. The Meaning Representation of "What aircraft at Hahn can strike BE70701?"

```
(*PTRANS* OBJECT (AIRCRAFT IS-A (*?*)
                           LOC (AT PLACE (HAHN)))
          TO (*PROX* PLACE (BE70701))
          FROM (*PROX* PLACE (NIL))
          MODE (*POTENTIAL*))
```

Figure 6b. The Concept Inferred from 6a. (What aircraft at Hahn can go to BE70701?)

```
(*PTRANS* TO (*PROX* PLACE &TARGET)
          FROM (*PROX* PLACE &AIRBASE)
          OBJECT &AIRCRAFT)
```

Figure 6c. A Question Answering Pattern which could Match 6b.

```
(*PTRANS* OBJECT (AIRCRAFT IS-A (*?*)
                           LOC (AT PLACE (HAHN)))
          TO (*PROX* PLACE (BE70701))
          FROM (*PROX* PLACE (HAHN))
          MODE (*POTENTIAL*))
```

Figure 6d. The Complete Question Conceptualization after Inferring the Source.

```
USER:   Send 4 aircraft from the 109tfw to strike BE70501 at 0900.
KNOBS:  What aircraft do you want to use?
USER:   What aircraft are in the 109TFW?
KNOBS:  The 109TFW has F-4Cs.
        Would you like to use F-4Cs for the aircraft?
USER:   No, F-4Gs.
KNOBS:  The 109TFW does not contain F-4Gs.
USER:   Fly the F-4Gs out of the 126TFW at Ramstein.
```

Figure 7. A Conversation with KNOBS.

```
(DEF-ROLE-PAT PAT (AIRCRAFT LOC (*PROX* PLACE &OCA:AIRBASE)
                                PART (PART ARGUMENT &OCA:WING))
              SCRIPT-ROLE &OCA:AIRCRAFT
              SCRIPT OCA)
```

Figure 8. A Script Role Pattern.

A script role can be bound by matching against patterns associated with other script roles in addition to matching against script patterns. Fig. 8 shows a role pattern associated with the script role AIRCRAFT. This pattern serves two purposes: to prevent bindings to the script role which would not make sense (i.e., the object which plays the AIRCRAFT role must be an aircraft) and to recursively bind other script roles to attached concepts. In this example, the AIRBASE or the WING could be attached to the AIRCRAFT concept, e.g., "F-4Cs from Hahn" or "F-4Cs in the 126TFW".

The interactive script interpreter is an alternative to the menu system provided by KNOBS for the entering of important components of a plan to be checked for consistency. KNOBS also provides a means of automatically finishing the creation of a consistent plan. This can allow an experienced mission planner to enter a plan by typing one or two sentences and hitting a key which tells KNOBS to choose the unspecified components.

TRANSFERRING DOMAINS

To demonstrate their domain independence, the KNOBS System and APE-II have been provided with knowledge bases to plan and answer questions about naval "show of flag" missions. This version of KNOBS also uses FRL as a database language.

A large portion of the question answering capability was directly applicable for a number of reasons. First of all, dictionary entries for frames are constructed automatically when they appear in a user query. The definitions of the attributes (slots) of a frame which are represented as RELATIONs are also constructed when needed. The definitions of many common words such as "be", "have", "a", "of", etc., would be useful in understanding questions in any domain. The question answering productions and concept completion inferences are separated into default and domain specific categories. Many of the simple but common queries are handled by default patterns. For example, "Which airbases have fighters?" and "What ports have cruisers?" are answered by the same default pattern. Currently, the Navy version of KNOBS has 3 domain specific question answering patterns, compared to 22 in the Air Force version. (There are 46 default patterns.) The most important knowledge structure missing in the Navy domain is the scripts which are needed to perform causal inferences and dialog directed planning. Therefore, the system can answer the question "What weapons does the Nimitz have?", but can't answer "What weapons does the Nimitz carry?".

CONCLUSION

We have argued that the processing of natural language database queries should be driven by the meaning of the input, as determined primarily by the meanings of the constituent words. The mechanisms provided for word sense selection and for the inference of missing meaning elements utilize a variety of knowledge sources. It is believed that this approach will prove more general and extensible than those based chiefly on the surface structure of the natural language query.

ACKNOWLEDGEMENTS

We would like to thank Tom Fawcett, Bud Frawley, Frank Jernigan, and Ethan Scarl for their comments.

This work was supported by USAF Electronics System Division under Air Force contract F19628-82-C-0001.

REFERENCES

Birnbaum, L., and Selfridge, M., "Conceptual Analysis," in Inside Artificial Intelligence: Five Programs Plus Miniatures, Schank, R., Riesbeck, C. K. (eds), Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.

Charniak, E., "Six Topics in Search of a Parser: An Overview of AI Language Research," in Proceeds of the 7th International Joint Conference on Artificial Intelligence, Vancouver, 1981.

Cullingford, R., "Script Application: Computer Understanding of Newspaper Stories," Research Report 116, Department of Computer Science, Yale University, 1978.

Cullingford, R. and Pazzani, M., "Word Meaning Selection in Multimodule Language-Processing Systems," TR-82-13, EE&CS Dept., University of Connecticut, 1982.

Engelman, C., Scarl, E., and Berg, C., "Interactive Frame Instantiation," in Proc. of The First Annual Conference on Artificial Intelligence, Stanford, 1980.

Goldman, N., "Conceptual Generation," in Conceptual Information Processing, Schank, R. (ed), Ninth-Holland Publishing Company, 1975.

Grosz, B., "The Representation and Use of Focus in Dialog Understanding," SRI Technical Note 151, 1977.

Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., and Slocum, J., "Developing a Natural Language Interface to Complex Data." Association for Computing Machinery Transactions on Database Systems, Volume 3, Number 2, June 1978.

Katz, J. S. and Fodor, J. A., "The Structure of Semantic Theory," Language, 39, 1963.

Lehnert, W., The Process of Question Answering, A Computer Simulation of Cognition, Lawrence Erlbaum Associates, Inc., 1978.

Reisbeck, C., and Schank, R. C., "Comprehension by Computer: Expectation Based Analysis of Sentences in Context," Research Report #78, Department of Computer Science Yale University, 1976.

Roberts, R. Bruce, and Goldstein, Ira P., "The FRL Manual," MIT AI Lab. Memo 409, September 1977.

Schank, R., "Conceptual Dependency: A Theory of Natural Language Understanding," Cognitive Psychology, Vol. 3, No. 4, 1972.

Small, S., "Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding," TR-954, University of Maryland 1980.

Wilks, Y., Grammar, Meaning & The Machine Analysis of Language, London, 1972.

Woods, W. A., Kaplan, R. M., and Nash-Webber, B., "The Lunar Sciences Natural Language Information System." BBN Report 2378, Bolt, Beranek, and Newman Inc., Cambridge, MA, 1972.