Interaction of Knowledge Sources in a Portable Natural Language Interface

Carole D. Hafner
Computer Science Department
General Motors Research Laboratories
Warren, MI 48090

Abstract

This paper describes a general approach to the design of natural language interfaces that has evolved during the development of DATALOG, an English database query system based on Cascaded ATN grammar. By providing separate representation schemes for linguistic knowledge, general world knowledge, and application domain knowledge, DATALOG achieves a high degree of portability and extendability.

1. Introduction

An area of continuing interest and challenge in computational linguistics is the development of techniques for building portable natural language (NL) interfaces (See, for example, [9,3,12]). The investigation of this problem has led to several NL systems, including TEAM [7], IRUS [1], and INTELLECT [10], which separate domain-dependent information from other, more general capabilities, and thus have the ability to be transported from one application to another. However, it is important to realize that the domain-independent portions of such systems constrain both the form and the content of the domain-dependent portions. Thus, in order to understand a system's capabilities, one must have a clear picture of the structure of interaction among these modules.

This paper describes a general approach to the design of NL interfaces, focusing on the structure of interaction among the components of a portable NL system. The approach has evolved during the development of DATALOG (for "database dialogue") an experimental system that accepts a wide variety of English queries and commands and retreives the answer from the user's database. If no items satisfy the user's request, DATALOG gives an informative response explaining what part of the query could not be satisfied. (Generation of responses in DATALOG is described in another report [6].) Although DATALOG is primarily a testbed for research, it has been applied to several demonstration databases and one "real" database containing descriptions and rental information for more than 500 computer hardware units.

The portability of DATALOG is based on the independent specification of three kinds of knowledge that such a system must have: a linguistic grammar of English; a general semantic model of database objects and relationships; and a domain model representing the particular concepts of the application domain. After giving a brief overview of the architecture of DATALOG, the remainder of the paper will focus on the interactions among the components of the system, first describing the interaction between syntax and semantics, and then the interaction between general knowledge and domain knowledge.

2. Overview of DATALOG Architecture

The architecture of DATALOG is based on Cascaded ATN grammar, a general approach to the design of language processors which is an extension of Augmented Transition Network grammar [13]. The Cascaded ATN approach to NL processing was first developed in the RUS parser [2] and was formally characterized by Woods [14]. Figure 1 shows the architecture of a Cascaded ATN for NL processing: the syntactic and semantic components are implemented as separate processes which operate in parallel, communicating information back and forth. This communication (represented by the "interface" portions of the diagram) allows a linguistic ATN grammar to interact with a semantic processor, creating a conceptual representation of the input in a step-by-step manner and rejecting semantically incorrect analyses at an early stage.

DATALOG extends the architecture shown in Figure 1 in the direction of increased portability, by dividing semantics into two parts (see Figure 2). A "general" semantic processor based on the relational model of data [5] interprets a wide variety of information requests applied to

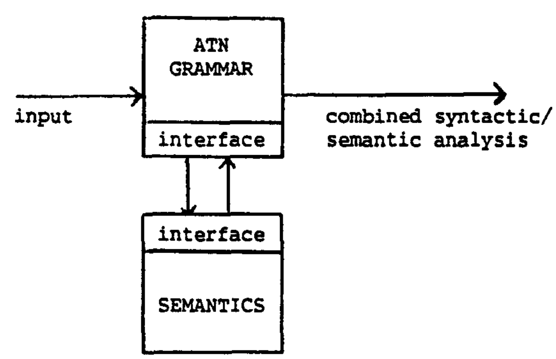

Figure 1. Cascaded Architecture for Natural Language Processing

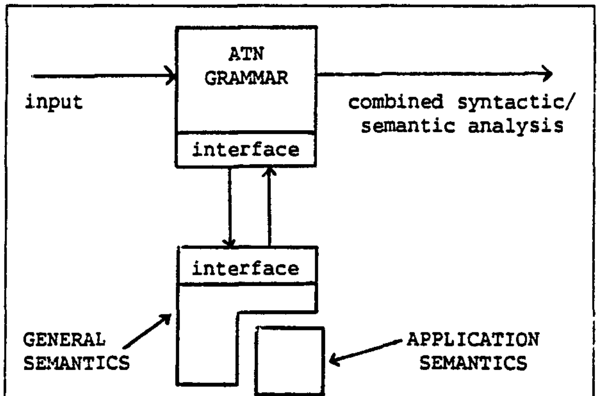

Figure 2. Architecture of DATALOG

abstract database objects. This level of knowledge is equivalent to what Hendrix has labelled "pragmatic grammar" [9]. Domain knowledge is represented in a semantic network, which encodes the conceptual structure of the user's database. These two levels of knowledge representation are linked together, as described in Section 4 below.

The output of the Cascaded ATN grammar is a combined linguistic and conceptual representation of the query (see Figure 3), which includes a "SEMANTICS" component along with the usual linguistic constituents in the interpretation of each phrase.

3. Interaction of Syntax and Semantics

The DATALOG interface between syntax and semantics is a simplification of the RUS approach, which has been described in detail elsewhere [11]. The linguistic portion of the interface is implemented by adding a new arc action called "ASSIGN" to the ATN model of grammar. ASSIGN communicates partial linguistic analyses to a semantic interpreter, which incrementally creates a conceptual representation of the input. If an assignment is nonsensical or incompatible with previous assignments, the semantic interpreter can reject the assignment, causing the parser to back up and try another path through the grammar.

In DATALOG, ASSIGN is a function of three arguments: the HEAD of the current clause or phrase, the CONSTITUENT which is being added to the interpretation of the phrase, and the SYNTACTIC SLOT which the constituent occupies. As a simplified example, an ATN grammar might process noun phrases by "collecting" determiners, numbers, superlatives and other pre-modifiers in registers until the head noun is found. Then the head is assigned to the NPHEAD slot; the pre-modifiers are assigned (in reverse order) to the NPPREMOD slot; superlatives are assigned to the SUPER slot; and numbers are assigned to the NUMBER slot. Finally, the determiners are assigned to the DETERMINER slot. If all of these assignments are acceptable to the semantic interpreter, an interpretation is constructed for the "base noun phrase", and the parser can then begin to process the noun phrase post-modifiers. Figure 3 illustrates the interpretation of "the tallest female employee", according to this scheme. A more detailed description of how DATALOG constructs interpretations is contained in another report [8].

During parsing, semantic information is collected in "semantic" registers, which are inaccessible (by convention) to the grammar. This convention ensures the generality of the grammar; although the linguistic component (through the assignment mechanism) controls the information that is passed to the semantic interpreter, the only information that flows back to the grammar is

```
Pushing for Noun Phrase.

    ASSIGN Actions:
       HEAD              CONSTITUENT          SYNTACTIC SLOT
       employee            employee                 NPHEAD
       employee            (AMOD female)            NPPREMOD
       employee            (ADJP                    SUPER
                             (ADV most)
                             (ADJ tall))
       employee            (the)                    DET

Popping Noun Phrase:
       (NP
         (DET (the))
         (PREMODS ((ADJP (ADV most) (ADJ tall)) (AMOD female))
         (HEAD employee)
         (SEMANTICS
           (ENTITY
             (Q nil)
             (KIND employee)
             (RESTRICTIONS (
               ((ATT sex) (RELOP ISA) (VALUE female))
               ((ATT height) (RANKOP MOST) (CUTOFF 1)) )))))
```

Figure 3. Interpretation of "the tallest female employee".

the acceptance or rejection of each assignment. When the grammar builds a constituent structure for a phrase or clause, it includes an extra constituent called "SEMANTICS", which it takes from a semantic register. However, generality of the grammar is maintained by forbidding the grammar to examine the contents of the SEMANTICS constituent.

4. Interaction of General and Application Semantics

The semantic interpreter is divided into two levels: a "lower-level" semantic network representing the objects and relationships in the application domain; and a "higher-level" network representing general knowledge about database structures, data analysis, and information requests. Each node of the domain network, in addition to its links with other domain concepts, has a "hook" attaching it to the higher-level concept of which it is an instance. Semantic procedures are also attached to the higher-level concepts; in this way, domain concepts are indirectly linked to the semantic procedures that are used to interpret them.

Figure 4 illustrates the relationship between the general concepts of DATALOG and the domain semantic network of a personnel application. Domain concepts such as "female" and "dollar" are attached to general concepts such as /SUBCLASS/ and /UNIT/. (The higher-level concepts are delimited by slash "/" characters.) When a phrase such as "40000 dollars" is analyzed, the semantic procedures for the general concept /UNIT/ are invoked to interpret it.

The general concepts also organized into a network, which supports inheritance of semantic procedures. For example, two of the general concepts in DATALOG are /ATTR/, which can represent any attribute in the database, and /NUMATTR/, which represents numeric attributes such as "salary" and "age". Since /ATTR/ is the parent of /NUMATTR/ in the general concept network, its semantic procedures are automatically invoked when required during interpretation of a phrase whose head is a numeric attribute. This occurs whenever no /NUMATTR/ procedure exists for a given syntactic slot; thus, sub-concepts can be defined by specifying only those cases where their interpretations differ from the parent.

Figure 5 shows the same diagram as Figure 4, with concepts from the computer hardware database substituted for personnel concepts. This illustrates how the semantic procedures that interpreted personnel queries can be easily transported to a different domain.

5. Conclusions

The general approach we have taken to defining the inter-component interactions in DATALOG has led to a high degree of extendability. We have been able to add new sub-networks to the grammar without making any changes in the semantic interpreter, producing correct interpretations (and correct answers from the database) on the first try. We have also been able to implement new general semantic processes without modifying the grammar, taking advantage of the "conceptual factoring" [14] which is one of the benefits of the Cascaded ATN approach.

The use of a two-level semantic model is an experimental approach that further adds to the portability of a Cascaded ATN grammar. By representing application concepts in an "epistemological" semantic network with a restricted set of primitive links (see Brachman [4]), the task of building a new application of DATALOG is reduced to defining the nodes and connectivity of this network and the synonyms for the concepts repre-

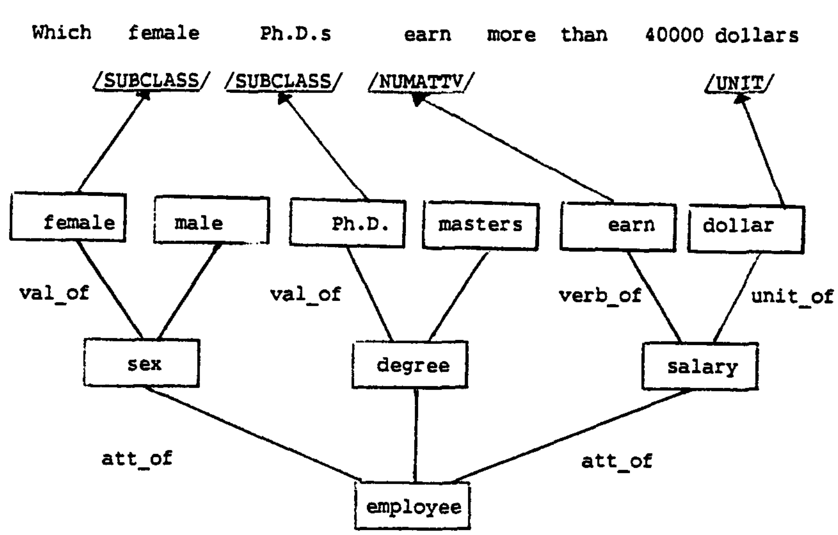

Figure 4. Interaction of Domain and General Knowledge

sented by the nodes. Martin et. al. [12] define a transportable NL interface as one that can acquire a new domain model by interacting with a human database expert. Although DATALOG does not yet have such a capability, the two-level semantic model provides a foundation for it.

DATALOG is still under active development, and current research activities are focused on two problem areas: extending the two-level semantic model to handle more complex databases, and integrating a pragmatic component for handling anaphora and other dialogue-level phenomena into the Cascaded ATN grammar.

6. References

1. Bates, M. and Bobrow, R. J., "Information Retrieval Using a Transportable Natural Language Interface." In Research and Development in Information Retrieval: Proc. Sixth Annual International ACM SIGIR Conf., Bethesda MD, pp. 81-86 (1983).
2. Bobrow, R. "The RUS System." In "Research in Natural Language Understanding," BBN Report No. 3878. Cambridge, MA: Bolt Beranek and Newman Inc. (1978).
3. Bobrow , R. and Webber, B. L., "Knowledge Representation for Syntactic/Semantic Processing." In Proc. of the First Annual National Conf. on Artificial Intelligence, Palo Alto CA, pp. 316-323 (1980).
4. Brachman, R. J., "On the Epistemological Status of Semantic Networks." In Associative Networks: Representation and Use of Knowledge by Computers, pp. 3-50. Edited by N. V. Findler, New York NY (1979).
5. Codd, E. F. "A Relational Model of Data for Large Shared Data Banks." Communications of the ACM, Vol. 13, No. 6, pp.377-387 (1970).
6. Godden, K. S., "Categorizing Natural Language Queries for Intelligent Responses." Research Publication 4639, General Motors Research Laboratories, Warren MI (1984).
7. Grosz, B. J., "TEAM: A Transportable Natural Language Interface System." In Proc. Conf. on Applied Natural Language Processing, Santa Monica CA, pp. 39-45 (1983).
8. Hafner, C. D. and Godden, K. S., "Design of Natural Language Interfaces: A Case Study." Research Publication 4567, General Motors Research Laboratories, Warren MI (1984).
9. Hendrix, G. G. and Lewis, W. H., "Transportable Natural Language Interfaces to Data." Proc. 19th Annual Meeting of the Assoc. for Computational Linguistics, Stanford CA, pp. 159-165 (1981).
10. INTELLECT Query System User's Guide, 2nd. Edition. Newton Centre, MA: Artificial Intelligence Corp. (1980).
11. Mark, W. S. and Barton, G. E., "The RUSGRAMMAR Parsing System." Research Publication GMR-3243. Warren, MI: General Motors Research Laboratories (1980).
12. Martin, P., Appelt, D., and Pereira, F., "Transportability and Generality in a Natural-Language Interface System." In Proc. Eight International Joint Conf. on Artificial Intelligence, Karlsruhe, West Germany (1983).
13. Woods, W. "Transition Network Grammars for Natural Language Analysis." Communications of the ACM, Vol. 13, No. 10, pp. 591-606 (1970).
14. Woods, W., "Cascaded ATN Grammars." American Journal of Computational Linguistics," Vol. 6, No. 1, pp. 1-12 (1980).

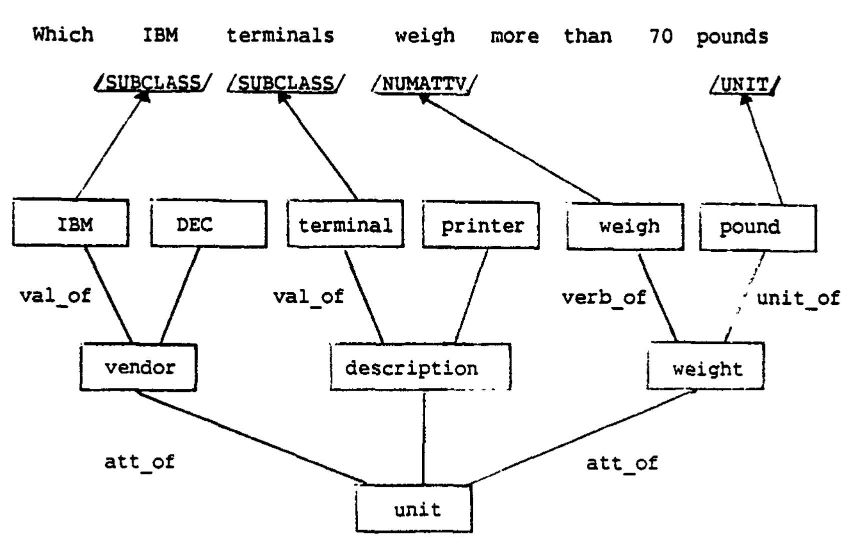

Figure 5. Figure 4 Transported to a New Domain