SEMANTIC REPRESENTATIONS FOR NATURAL LANGUAGE QUERY PROCESSING

STEPHEN D. BURD, SHUH-SHEN PAN, AND ANDREW B. WHINSTON

1. Introduction

One of the recommendations of the CODASYL committee\(^{(7)}\) was the development of a nonprocedural language for accessing a database system. The goal of this recommendation was to allow a larger number of users to access a database. Ideally, a query language should allow a user to access the database without procedurally specifying exactly how this access must be performed. In addition, the language should be simple to use and should require a minimal amount of training and prerequisite knowledge on the part of the user.

Since this recommendation was made, many such languages have been developed which have met the ideal standards with varying degrees of success. Examples of such languages include SQUARE,\(^{(2)}\) SEQUEL,\(^{(3)}\) SQL, and Query by Example.\(^{(23)}\) Most of these languages are based on context-free languages with fairly rigid syntactic and semantic requirements. Query-by-Example and SQUARE differ significantly from the others in that queries are expressed via data entries into tables rather than via a language per se.

Although many of these query languages are quite powerful, they are not entirely free of procedural specification and require that users be trained in their use. In a study evaluating the learnability of SQUARE and SEQUEL, Reisner, Boyce, and Chamberlin\(^{(18)}\) showed that reasonable profi-
ciency in either language required 12–14 hours of formal instruction. In addition, considerable difficulty was experienced by some subjects in learning and retaining some of the more complex language features. Thomas and Gould (20) studied Query-by-Example and showed that the system could be learned more quickly than SQUARE and SEQUEL, but the number of query formulation errors encountered was still quite high.

Other researchers sought to bypass these and other problems associated with artificial languages by designing natural language query processing systems. Winograd (21) developed a natural language processing system to interface to "blocksworld." The system was capable of recognizing both commands and queries within this limited domain. Harris (11) developed ROBOT, which employed a parser based on augmented transition networks to recognize database queries. This system was the basis for a commercial product. Kaplan (12) developed the CO-OP system, which provided sophisticated error-handling capabilities for natural language queries. McCord (15) developed a Prolog formulation partially based on his slot grammar (16) and used it to process natural language database queries.

This chapter will examine the role of semantic data modeling techniques in the development of a natural language query system. We will critically examine previous natural language query systems in terms of the tools used to build them and the semantic knowledge and representations employed. We will largely ignore issues of syntax in order to concentrate on semantic problems. This is not to imply that an adequate syntactic model is unimportant for natural language understanding. We simply wish to insulate ourselves from the present competition between alternative syntactic models.

We will consider a query to be a single interrogative statement which can be answered by retrieving data fields from the database. By restricting our attention to single-sentence queries, we can utilize a simpler recognition model than would be required if we were concerned with an entire discourse. We will require that the query be syntactically correct and that the data necessary to answer the query exist within the database.

2. Semantic Issues in Parsing

Most of the more successful attempts at natural language understanding have integrated syntactic and semantic analyses to parse the input string and generate an internal semantic representation thereof. McCord (15) built a system employing this approach using the definite clause grammars of Pereira and Warren. (17) Definite clause grammars and McCord's approach will be discussed in greater depth later in this chapter.