A Paradigm For Natural Language Explanation Of Database Queries:
A Semantic Data Model Approach

Vesper Owei
Kunihiko Higa
Georgia Institute of Technology

Although the increasing pervasiveness of end-user computing has made concerns with database interfaces increasingly more important, paradoxically, there does not seem to be sufficient research into database interfaces for end-user computing. A much needed but neglected enrichment of database systems is an interface that provides the user with automatic feedback in the form of an explanation of how the database management system interprets user-specified queries. The study presented in this paper proposes an approach that exploits the rich semantics of graphical semantic data models to construct restricted natural language explanations of database queries that are specified in a very high-level declarative form. These interpretations of the specified query represent the system’s “understanding” of the query, and are returned to the user for validation.

“Why aren’t researchers more interested in user interfaces?” is the frequent complaint of Professor Larry Rowe of the University of California, Berkeley (Stonebraker, 1988). The question is in itself a testament to the increasing recognition of the importance of user-interfaces in this era of end-user computing (EUC). And as the database approach makes its relentless match into an increasingly more diverse array of application areas, concerns with user-interfaces would become increasingly more important.

Arguably, application as well as business needs have fueled extensive research into database (DB) query languages, and this has given rise to the numerous DB query languages that currently exist. Such is the extent and intensity of the research in this area that already, the field of database query languages shows signs that it is approaching maturity. For instance, no longer is it the case that database systems can be evaluated primarily on the query languages that they support. Instead, the focus has begun to shift towards user interfaces. Stonebraker (1988) recognizes the reason for this shift, and lists two reasons why more research should be done in the area of user-interfaces:

1. The functions provided by database systems are determined by user interface programs.
2. The “value added” by database systems will increasingly migrate to their front-end, since query languages will increasingly become commodity products with respect to their level of performance. There is, therefore, need for increased research in user-interfaces.

A much needed but neglected enrichment of database systems is an interface that provides the user with automatic feedback in the form of an explanation of how the database management system interprets user-specified queries. This is intended to assure the user that the system interprets, or understands, the query as it is intended, or that the specified query is the desired one. The study presented in this paper proposes a paradigm that exploits the rich semantics of graphical semantic data models to construct restricted natural language (NL) interpretations of database queries that are specified in a very high-level declarative form. These interpretations of the specified query represent the system’s “understanding” of the query, and are returned to the user for validation.

The rest of the paper is organized as follows. A review of those works that are related to this study is presented in Related Research. Then, the components and architecture of the proposed approach are presented through illustrative ex-
amples in the following section. The section, Pseudo-Natural Language Interpretation System, is devoted to illustrating the proposed paradigm through examples. We conclude this paper with a discussion on its contributions, limitations, and related future research.

**Motivation**

In his continuing argument in support of a universal-relation interface, Ullman (1989) states that the real motivation is that queries can be posed, due to the simplicity of the interface, by users “who understand nothing of the structure of relations.” This holds true for the system we propose, and our motivational example here illustrates Ullman’s argument. This section is solely devoted to illustrating the motivation for this study.

Consider Figure 1, which is a Structured Object Model (SOM), described in Higa and Owei (1991) and Higa and Sheng (1989), of a university department. The boxes on this schema represent objects, or entities, while the double-headed arrows indicate that the objects at both heads have some direct semantic relationship, e.g., Is-a or Has relationship. The number k on an arrow has the following meaning: k represents the type of association, or link, semantics between the entities linked by the arrow. As we discuss later, a link semantic dictionary is used as a repository for these link semantic types, i.e., for the set of k values and their semantic meaning. However, for our motivational example in this section, the link semantic types can be ignored.

The equivalent relations for the objects in the schema of Figure 1 are:

- STUDENT(S-id, S-name, S-age)
- COURSE-ENROLLMENT(C-id, Sec-no., S-id)
- COURSE(C-id, C-name)
- SECTION(Sec-no., C-id, T-name)
- TEACHER(T-name, T-title, T-age)
- SECRETARY(Sek-name, Sek-title)

Suppose we pose the following query:

**Query 1:** What course(s) is the student with student-id ‘10’ taking from Professor ‘Jones’?

A Structured Query Language (SQL)-type command can be written to specify this query as follows:

```
SELECT C-name
FROM C
WHERE C-id IN (SELECT C-id
FROM Sec
WHERE T-name IN (SELECT T-name
FROM T
WHERE T-name = 'Jones')
AND C-id, Sec-no. IN (SELECT C-id,
Sec-no.
FROM CE)
WHERE S-id IN (SELECT S-id
FROM Std
WHERE S-id = '10')))
```

An alternative implementation of this query is:

```
SELECT C-name
FROM C, Std, Sec, CE
WHERE S-id = '10'
```

**Legend:**

- Std: STUDENT
- Sec: SECTION
- CE: COURSE-ENROLLMENT
- C: COURSE
- T: TEACHER
- Sek: SECRETARY

---

**Figure 1: University Department Schema**
Related Content

Improving Business Intelligence Traceability and Accountability: An Integrated Framework of BI Product and Metacontent Map
www.igi-global.com/article/improving-business-intelligence-traceability-and-accountability/118087?camid=4v1a

A Two-Stage Zone Regression Method for Global Characterization of a Project Database
www.igi-global.com/chapter/two-stage-zone-regression-method/8016?camid=4v1a

Conceptual Data Modeling Patterns: Representation and Validation
www.igi-global.com/article/conceptual-data-modeling-patterns/3333?camid=4v1a

Improving Constraints Checking in Distributed Databases with Complete, Sufficient, and Support Tests
www.igi-global.com/chapter/improving-constraints-checking-distributed-databases/20718?camid=4v1a