Kaleidoscope: A Cooperative Menu-Guided Query Interface

Sang K Cha and Gio Wiederhold
Stanford University

Querying databases to obtain information requires the user's knowledge of query language and underlying data However, because the knowledge in human long-term memory is imprecise, incomplete, and often incorrect, user queries are subject to various types of failure These may include spelling mistakes, the violation of the syntax and semantics of a query language, and the misconception of the entities and relationships in a database

Kaleidoscope is a cooperative query interface whose knowledge guides users to avoid most failure during query creation We call this type of cooperative behavior *intraquery guidance* To enable this early, active engagement in the user's process of query creation, Kaleidoscope reduces the granularity of user-system interaction via a context-sensitive menu The system generates valid query constituents as menu choices step-by-step by interpreting a language grammar, and the user creates a query following this menu guidance[2] For instance, it takes four steps to create the following query

```
[Q1] Who authored 'AI' journal papers in 'Postquery COOP'
     1   2        3   (3+)           4
```

At each of such steps, as the user selects one of menu choices, the system updates its partial query status window If a choice is unique as in *(3+)*, it is taken automatically To guide the user's entry of values, the system provides a pop-up menu for each value domain

With Kaleidoscope's process of choice generation tightly controlled by the system's knowledge of query language and underlying data, users need not remember the query language and the underlying database structure but merely recognize or identify the constituents coming one after another that match their intended query The system provides additional guidance for users to avoid creating semantically inconsistent queries It informs the user of any derived predicates on the completion of a user-selected predicate To illustrate this, consider a partially constructed SQL query

```
[Q2] SELECT *
     FROM   professor p#1
     WHERE  p#1 dept = 'CS' AND p#1 salary < 40000
```

Suppose that the system has an integrity constraint

```
[IC] FROM professor p
     IF     p dept = 'CS' AND p salary < 45000
     THEN   p rank = 'Assistant'
```

This rules states that a CS professor whose salary is less than 45000 is an assistant professor With the replacement of rule variable p in IC by Q2's range variable p#1, IC's leading two predicates subsume Q2's query condition, producing p#1 rank = 'Assistant ' Because this derived predicate is not subsumed by Q2's query condition, the system suspects that the user may not know of it and presents it to the user

Derived predicates, together with user-selected ones, constrain the user's further conjunctive extension of the partial query condition For example, the system prunes the field rank (as well as the field dept) in the conjunctive extension of Q2, because the derived condition restricts the value of this field to a constant

As shown in examples, we apply Kaleidoscope's approach to two linear-syntax languages in different levels of abstraction SQL[1], and a query language whose syntax and semantics cover a subset of *wh*-queries To implement the intraquery guidance, we extend context-free grammar by associating context variables with each grammar symbol and attaching several types of procedural decorations to grammar rules This extension enables the system to capture the semantic constraints and its user-guiding actions in a domain-independent grammar As the grammar is interpreted, the database-specific information is fed from the system's lexicon and knowledge base The current implementation of Kaleidoscope runs on a XEROX-1186 LISP machine with a SUN server configured with a relational DBMS

The approach of Kaleidoscope is based on the normative system assumption The system presents its capability transparently to the user in a context-dependent manner during the user's query creation This makes the system usable even with a small amount of stored knowledge

References

[1] Sang K Cha Kaleidoscope A cooperative menu-guided query interface (SQL version) In *Proc IEEE Artificial Intelligence Applications*, Santa Barbara, California, pages 304–310, March 1990

[2] C W Thompson, K M Roth, H R Tennant, and R M Saenz Building usable menu-based natural language interface to databases In *9th Conf on VLDB*, pages 43–55, 1983