Project Part I: Code Review

Daniel M. Wong

dmw@cs.wisc.edu

March 6, 1995

1 Objective

This is a study of the Tcl/Tk coding techniques, parsing techniques, and data structures used in a graphical query browser developed for an object oriented database by Vivek (vivek@cs.wisc.edu). The goal of this code review is to identify components that are of potential interest to the $QBE - \gg SQL$ translator course project.

2 Overview

Majority coding of the browser, including parsing of user inputs, is done in Tcl/Tk. It mainly consists of a query translator, which parses user input and translates to corresponding SQL statements, and a graphical query browser, which manages the presentation of the query interface to the users and the interactions with the databases. The browser is meant to be built upon a specialized interface layer that provides a standard interface of several system dependent functionalities, thus allowing heterogeneous database systems environment to work under the same browser.

3 Query Translator

The program structure intelligently takes advantage of the built-in list manipulation operations in Tcl to parse the user inputs from lists to tokens. Since only constants and algebraic operators are expected in the input, translation mechanism basically involves maintaining a list of relations (tables) and domains (constraints), and when all inputs are obtained, insert appropriate SQL key words, such as FROM and WHERE, to connect the lists together to form a SQL query.

4 Query Browser

The browser employs advanced Tk techniques such as packer for geometry management and grab for controlling input streams. Interactions with specific databases is supported by a system dependent interface layer which provides a standard interface to the browser. Another unique feature is that history of user input is maintained by a global token id, thus users are allowed to inspect previous queries.

5 Remark

There is some propriety code C++ code that is not available. However, it is quite clear that the original widget data structure is being enhanced to include additional information on the resource identifier, process identifier and other relevant information for each query submitted.

6 Conclusion

There exists little similarity between the structure of current graphical query browser and $QBE-\gg SQL$ translator because of the difference in main goal. While a lot of work has been done on the graphical query browser to visually provide general query support for object oriented database systems, $QBE- \gg SQL$ translator focus only on the implementation technique in parsing the user inputs from pure relational environment, such as cross relation joins, which the current graphical query browser does not support yet. The $QBE- \gg SQL$ translator will have to support at least simple joins. Unfortunately, this is where the complexity of the data structure for $QBE- \gg SQL$ lies. Nevertheless, the layer structure of the graphical browser can very well serve as a model to the QBE browser.

7 Acknowledgement

The code review could not have been done without Vivek's help.

8 Reference

Ousterhout, John ——- `An Introduction to Tcl/Tk` ——–