Appendix B: FRDB Architecture: The FSQL Server

The FSQL Server is a program that allows users to employ the FSQL language (see Chapter VII). The FRDB and the FSQL Server have been implemented partially by using an already existing DBMS.

Actually, the FSQL Server does not include all the statements and clauses defined in the FSQL language. However, it is useful for the most usual operations and queries. Basically, using an existing DBMS involves three consequences:

1. The system will be slower than if it were programmed at a lower level.
2. The task is made much simpler (we do not have to build the DBMS).
3. We obtain all the advantages of the host DBMS (security, efficiency, etc.) without the server having to take them into account.

The DBMS chosen was Oracle because of its adaptability, its large extension, and its ability to program packages (with functions and procedures) internal to the system in its own language, PL/SQL, which turns out to be quite efficient. Of course, this architecture can be implemented in other DBMSs, and we are now working in an implementation using PostgreSQL.

The tests that we carried out prove that the FSQL Server is very fast due to its ability to function as a real-time server. Obviously, if the query is very complicated and the database is very large, even if the translation is fast, the information recovery may be a little slow; if the condition were long, then the DBMS would carry out many operations.

The usefulness of this server is clear (see Chapter VIII), because it is even useful in traditional databases, which use only fuzzy attributes Type 1. However, we would like to underline the possibilities the server could offer when used in combination with data-mining techniques (such as classification).

Data, FSQL Server, and FSQL Client

Basically, the architecture of the FRDB with the FSQL Server is as follows:
1. **Data**: Traditional database and fuzzy metaknowledge base (see Chapter V).
2. **FSQL Server**.
3. **FSQL Client**.

---

### FSQL Server

The Oracle version of the FSQL Server has been programmed entirely in PL/SQL and includes three kinds of functions:

1. **Translation Function** (**FSQL2SQL**): This function carries out a lexical, syntactic, and semantic analysis of the FSQL query. If errors of any kind are found, it will generate a table with all those errors. If the query has no errors, it is translated into a standard SQL sentence, which includes reference to the following two kinds of functions.
2. **Representation Function**: This function is used to show the fuzzy attributes in a comprehensible way for the user and not in the internal format.
3. **Fuzzy Comparison Function**: This function is utilized to compare the fuzzy values and to calculate the compatibility degrees (**CDEG** function).

Summarizing, the translation function replaces the fuzzy attributes of the **SELECT** by calls to representation functions, the fuzzy conditions by calls to the fuzzy comparison functions, and the **CDEG** functions by calls to the fuzzy comparison functions and other functions if some logic operators exist (default functions are shown in Table 7.4).

The current version of the FSQL Server stores and controls some aspects about it and its processing: version, installation date, last utilization date, number of error for the last utilization and for all of them, number of access without errors, time employed for the last utilization and for all of them, and so forth.

### FSQL Client

The FSQL Client is an independent program that serves as an interface between the user and the FSQL Server. The user introduces the FSQL query, and the client program communicates with the server and with the database in
order to obtain the final results. The translation function of the FSQL Server is the only function that the client directly executes. We have developed a FSQL Client for Windows, called FQ, Fuzzy Queries (Galindo, 1999). On the other hand, there exists another FSQL Client, called Visual FSQL, which allows the user to construct an FSQL query through mouse clicks, without the need to write much (Oliva, 2003; Galindo et al., 2004d).

**Calling to the FSQL Server**

We summarize the process of using the FSQL Server for queries in Figure B.1. In short, an FSQL query involves the following steps:

1. The FSQL Client program sends the FSQL query to the FSQL Server.
2. The FSQL Server analyzes the query and, if it is correct, generates an SQL sentence starting from the original query. In this step, the FSQL Server uses the information of the FMB.
3. After the query has been generated in SQL, the client program reads it.
4. The client program sends the SQL query to any database that is coherent with the FMB. In the execution of this query, functions of the FSQL Server are used (representation and fuzzy comparison functions).
5. Finally, the client receives the resulting data, which shows them.
Steps 3 and 4 could have been eliminated to increase the efficiency, but the method we have presented achieves independence between the translation phase (Steps 1, 2, and 3) and the consultation phase (Steps 4 and 5). As such, if we make use of a local database with FSQL Server and FMB, we will be able to translate our sentences locally and send the translated queries to a remote database, avoiding network overload with error messages, translated queries, and so forth. This way, the remote database would not have to have the translation function installed.

If the statement is not a query, the process is similar. That statement is analyzed, and the FSQL Server generates modifications in the FMB and/or in the storage structures (including the Data Dictionary of the DBMS).

The presented architecture is not ideal for a final product, but it allows us to evaluate the possibilities of an FRDB on a practical level rather than only on a theoretical one. Let us hope that the DBMS soon incorporates new types of internal data (see Chapter IV) that allow the storing of fuzzy values and the fuzzy processing of these values.

**Endnotes**

1. The actual version of the FSQL Server can be downloaded free from http://www.lcc.uma.es/~ppgg/FSQL
2. PostgreSQL is Open Source database software: http://www.postgresql.org