A WEB TOOL FOR WEB DOCUMENT AND DATA SOURCE SELECTION WITH SQLFI

Marlene Goncalves and Leonid Tineo
Universidad Simón Bolívar, Departamento de Computación, Apartado 89000, Caracas 1080-A, Venezuela

Keywords: Queries, fuzzy queries, database, Web interface, SQL.

Abstract: WWW is composed of a great volume of documents that are stored by several data sources. Normally, a user is interested in those documents that include certain keywords. However, these documents might be incomplete, ancient or huge, and therefore, the user would have to discard irrelevant ones. An ideal Web search tool might select the best documents in base on user criteria defined over quality parameters such as completeness, recentness, frequency of updates and granularity. Traditional query languages are very restrictive in expressing preference-based queries. Therefore new query languages, such as SQLf, are needed. We present a tool that allows the selection of the best data sources and documents in terms of user preferences. Documents and data sources are described according to quality parameters. User preferences are expressed by means of SQLf queries. Our tool contains a wizard to retrieve the best documents and data sources. Thus, the user is oriented by a set of steps where a preference query that involves quality parameters is built easily.

1 INTRODUCTION

Two problems exist when a user wants to specify preference using a traditional querying language, such as SQL. First, a traditional query might return empty answers if user criteria are very restrictive and none of the rows satisfy the criteria. Second, a lot of answers might be retrieved if criteria are so soft and therefore, the user must discard irrelevant ones. For these two reasons, preference querying languages have been defined mainly.

SQLf is a preference querying language based on fuzzy logic (Bosc and Pivert, 1995). SQLf2 and SQLf3 are two SQLf extensions that consider SQL2 (ANSI X3, 1992) and SQL3 (Melton, 1992) standards, respectively (Goncalves and Tineo, 2001a, 2001b). In this work, we have adopted SQLf extensions due to the best of our knowledge none of the existing preference querying languages include SQL2 and SQL3 features in order to get a higher expressivity in the queries.

We think that it is not sufficient to propose fuzzy query languages, but also make a real implementation of fuzzy querying systems. In this sense, we have developed SQLfi (Eduardo, Goncalves and Tineo, 2004) which is a fuzzy querying system that implements SQLf, SQLf2 and SQLf3 languages, and it is conceived to be used on Internet.

On the other hand, high volume of available data sources and documents on the Web leads us to select what are the best ones that must be used for any search. Both documents and data sources might be incomplete, ancient or huge. Thus, the best ones might be selected in terms of user criteria defined over quality parameters such as completeness, recentness, frequency of updates and granularity.

As ever, if the selection is based on traditional querying languages, there are risks of empty answers and a lot of undiscriminated answers. As we are selecting not the documents but the data sources, the problem is intensified, because a bad selection might leads to a lot of undesired answers and unacceptable performance. Additionally, a user may spend a lot of superfluous time examining a huge undiscriminated answer. In consequence, SQLfi might be used to choose the best data sources and documents.

However, a SQLfi user needs to know SQLf language. So, an amateur user requires a wizard that builds a preference query where the best data sources and documents are retrieved. In this work, we present a Web tool that allow to select the best documents and data sources in base on user criteria defined over quality parameters and using SQLfi.

Gonçalves M. and Tineo L. (2007).
A WEB TOOL FOR WEB DOCUMENT AND DATA SOURCE SELECTION WITH SQLFI.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - DISI*, pages 449-452
DOI: 10.5220/0002393704490452

Our tool has two components. The first, a preference query wizard to give the user guidance on the choice of the best documents building a preference query that involve quality parameters. The second, a catalog manager that describes documents and data sources by means of quality parameters.

The paper comprises 5 sections. In Section 2, we briefly describe related works and background. In Section 3, we present a motivating example for selecting of the best documents and data sources. In Section 4, we show the architecture of our Web selection tool. Finally, in Section 5, the concluding remarks and future work are pointed out.

2 RELATED WORKS AND BACKGROUND

On one hand, several works have been proposed to provide flexible querying database systems; some of them are: (Loo and Lee, 2000; Galindo, Urrutia and Piattini, 2006). However, none of these systems allow to specify fuzzy queries with both SQL2 and SQL3 features, and therefore they have less expressivity capabilities. On the other hand, various researches have focused their interest in Web querying languages: (Konopnicki and Shmueli, 1995; Mendelzon, Mihaila and Milo, 1997; Mihaila, Raschid and Tomasic, 1998). But, none of these previous works have incorporated fuzzy features. Finally, Goncalves and Tineo (2005) proposed solving Web data source selection problem by means of SQLfi use.

A SQLf query has the following syntax:

```
SELECT <attributes> FROM <relations>
WHERE <fuzzy condition>
WITH CALIBRATION [n|λ|n,λ]
```

Its result is Cartesian product of the relations in the FROM clause, selecting those rows that satisfied the fuzzy condition and taking the fuzzy projection of attributes in the SELECT clause. In the WHERE clause, some logical expressions can be used with user-defined terms (atomic predicates, modifiers, connectors, comparators and quantifiers) and predefined fuzzy operators (Bosc and Pivert, 1995). The WITH CALIBRATION clause specify a tolerance which may express a maximum number "n" of best rows (quantitative) or may specify a satisfaction degree "λ" where returned row must have degree greater or equal to "λ" (qualitative).

SQLf also allows nesting subqueries and partitioned queries. Additionally, SQL2 features have incorporated into SQLf2 (Goncalves and Tineo, 2001a). Such features includes: relational algebra operators, integrity constraints, views, date and time data types, subqueries in the FROM clause, data manipulation operations. On the other hand, SQLf3 (Goncalves and Tineo, 2001b) includes features of deductive, active and object oriented databases. For simplicity and space restrictions, we do not present here the features of SQLf2 and SQLf3 in detail.

3 MOTIVATING EXAMPLE

Consider a collection of WWW data sources covering different domains. Data quality is described in terms of parameters, such as: *completeness* (a subset of data domain may be contained by a data source), *recentness* (the last date of data updating), *frequency of updates* (updating rates of data) and *granularity* (different levels of aggregations are possible in different data sources).

A query does not have to be answered by a single source. For example, the querying system must combine multiples sources data because they might contain incomplete documents. In this case, the source selection must be done in base on completeness parameter.

We assume the existence of a catalog containing information about Web documents stored in diverse registered data sources and their quality parameters. We can suppose that the Web users score each necessary quality parameter, and thus, the catalog is loaded. Finally, the catalog schema is described by:

- *SOURCE* that contains information about registered data sources. Its attributes are: *sid* (primary key), *name, country, city, completeness, recentness, frequency.*
- *DOCUMENT* that is characterized by *did* (primary key), *sid* (foreign key to SOURCE relation), *keywords, text, url and date attributes.* It contains all the data about Web documents belonging to registered data sources.

Our problem is to select the best Web data sources and their documents needed to solve any Web search. This selection is specified by a SQLf statement and involves criteria that are based on quality properties of the data sources.

Now, suppose that someone is interested for getting documents with the keyword 'fuzzy queries' due to his research. A query that system may perform in order to select the best data sources is:

Query 1 What are the 10 best data sources that contains very recent Web documents with a high

frequency of updates and whose keywords include the phrase 'fuzzy queries'?

Query 1 may be specified in SQLf as:

```
SELECT s.* FROM SOURCE s, DOCUMENT d
WHERE (d.idSource = s.id AND
      d.keywords LIKE '%fuzzy queries%'
      AND s.recentness = very current
      AND s.frequency = highFrequency)
WITH CALIBRATION 10;
```

Where the *highFrequency* and *current* predicates in Figure 1 and Figure 2, and *very* modifier must be previously defined with the following three sentences:

```
CREATE FUZZY PREDICATE current ON DATE
   AS ('01/01/2007','03/01/2007',∞,∞);
CREATE FUZZY PREDICATE highFrequency
AS (1/daily,0.8/weekly,0.6/bimonthly,
    0.5/monthly,0.3/quarterly,
    0.1/semestral,0/yearly);
CREATE FUZZY MODIFIER very AS POWER 2;
```

Moreover, we want to facilitate user's task by means of a generator of fuzzy queries as Query 1.

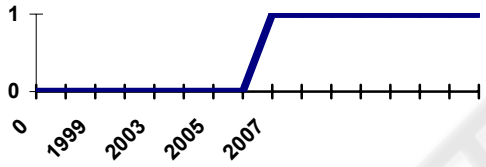

Figure 1: Current predicate.

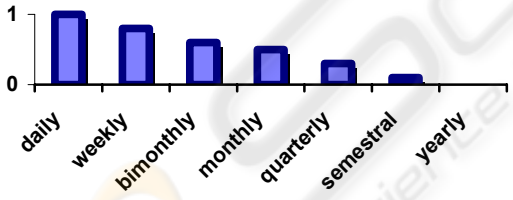

Figure 2: High Frequency predicate.

4 WEB TOOL ARCHITECTURE

For building the Web tool for data source and document selection, we have used a three layers' architecture. The lowest, data layer, is the RDBMS SQLfi. The middle is the logical layer. The upper layer is the client's interface.

The system has three types of users. First, the unregistered user may search documents in terms of criteria and terms by default. Second, the registered user may create him own terms and quality parameters, and make searches based on them. Third, the administrator registers data sources, documents and links.

The main components inside the Web tool are:

- *User Management*: Any user may register and connect him to the system. Additionally, a registered user and administrator may update his user data.
- *Catalog Management*: The administrator may add, delete or update data sources which will be accessible by all users. Each data source is characterized by its name, country, city, url and quality parameters, documents and links. A data source is added via interface or loading a XML document specified by the user. Subsequently, Web user may score each data source and the quality parameters are updated. Also, the data sources are queried, eliminated and modified.
- *Criteria Management*: A registered user may add, delete or update his fuzzy criteria or default ones; and the administrator may add, delete or update fuzzy criteria by default.
- *Terms Management*: A registered user may update his fuzzy terms or default ones; and the administrator may update default terms.
- *Fuzzy Searching*: Two types of searches are provided. The first is the basic search where one (or more) keyword is specified by the user in order to verify if it is contained in a document. The second is the advanced search that builds a fuzzy query orienting the user during the process.
- *Favorites Management*: A registered user may add his favorite documents to favorite list and then, query them.
- *Key Management*: A user may add keywords to a document for a better search. Also, he may delete, update and query them.
- *Historical Management*: A registered user may review and delete his historical queries made previously.

An important component of this architecture is the catalog. It contains metadata describing data sources, such as: data source content description, data source capacity, data source completeness, data source performance and use statistics, equivalence between data sources, and physical properties of network and data sources. In the catalog, both documents and data sources have quality parameters. The catalog contains the following entities:

- *Category* is the category name in which several

documents have common features.

- *Criterion* is the parameter name that measures the quality of documents or data sources.
- *Document* is the file that contains information of interest for the user. It is characterized by an identifier, a title, a size, a size unit, a city, keywords, a URL, a country, a language, a date, a last updating date, a format, authors and an abstract.
- *Data Source* is a document container described by identifier, name, country, city and URL.
- *Historical* is the user-search historical whose attributes are date, keyword phrase, search type and query.
- *Data Source instance* is the data source category.
- *User* described by *login, password, user type and e-mail.*

5 CONCLUSIONS

We have addressed here a solution for the problem of selecting web data sources by means of SQLfi use. This system has fuzzy querying capabilities that are useful for expressing quality criteria and selecting data sources with discrimination between them. Expression of such criteria would be very difficult with traditional querying languages.

We have presented a web data source selection tool based on SQLf. It is a complete and friendly system that facilitates the execution of fuzzy queries on a web data source and documents catalog. This tool allows storing the preferences of each user enlarging the degree of satisfaction with the results obtained. This system could be improved or expanded; nevertheless we are sure that its current state already represents a great aid for the end user.

We have not deal here with the performance issue of fuzzy querying and information retrieval systems. Nevertheless the querying system uses an evaluation strategy based on relationship between fuzzy sets and crisp sets. This mechanism is known as the derivation principle and has shown to posses the best performance between proposed evaluation mechanisms. It is matter of further work the whole implementation of the data source selection system and its performance study.

At present time, we only give support to finding data sources that are registered in existing relational catalogs. Therefore, it is also necessary to extend the system in order to support discovering data sources that publish Web documents. A step more would be to integrate this system with a Web tool for discovering information. Thereafter it would be possible to define and implement an intelligent Web querying tool that automatically optimize user request with available data sources on the Web.

ACKNOWLEDGEMENTS

We would like to acknowledge the contribution of our development team conformed by the Computer Engineering students Fabio Canache, Irwing Herrera, Felix González, Giuseppe Pellegrino, Jesús Graterol and Denise Videtta. This work has been made with the subsidy of FONACIT via the project G-2005000278. Finally we acknowledge Jesus Christ, our all time helper, who gives us the inspiration for working, creating and living.

REFERENCES

ANSI X3, 1992. Database Language SQL, 135-1992, American National Standards Institute, New York.

Bosc, P. and Pivert, O., 1995. SQLf: A Relational Database Language for Fuzzy Querying. IEEE Transactions on Fuzzy Systems, 3.

Eduardo, J., Goncalves, M and Tineo, L., 2004. A Fuzzy Querying System based on SQLf2 and SQLf3. Proceedings of CLEI.

Galindo J., Urrutia A., Piattini M., (2006). Fuzzy Databases: Modeling, Design and Implementation. Idea Group Publishing Hershey.

Goncalves, M and Tineo, L., 2001. SQLf Flexible Querying Language Extension by means of the norm SQL2. In Proceedings of FUZZ-IEEE, 1.

Goncalves, M and Tineo, L., 2001. SQLf3: An extension of SQLf with SQL3 features. In Proceedings of FUZZ-IEEE, 3.

Goncalves, M and Tineo, L., 2005. WWW Data Source Selection with SQLfi. In Proceedings of FUZZ-IEEE.

Konopnicki, D. and Shmueli, O., 1995. W3QS: A query system for the World Wide Web". In Proceedings of VLDB, 54-65.

Loo, G. and Lee, K., 2000. An Interface to Databases for Flexible Query Answering: A Fuzzy-Set Approach". In Proceedings of DEXA, 654-663.

Melton, J., 1992. ISO/ANSI Working Draft: Database Language SQL (SQL3), X3H2-93-091/ISO DBL YOK-003.

Mendelzon, A., Mihaila, G. and Milo, T., 1997. Querying the World Wide Web. Journal of Digital Libraries, 68-88.

Mihaila, G., Raschid, L. and Tomasic, A., 1998. Equal time for data on the Internet with WebSemantics. In Proceedings of EDBT, 87-101.