SuperSQL: An Extended SQL for Database Publishing and Presentation

Motomichi Toyama
Department of Information and Computer Science
Keio University, Yokohama, Japan
toyama@ics.keio.ac.jp

Abstract

SuperSQL is an extension of SQL that allows query results presented in various media for publishing and presentations with simple but sophisticated formatting capabilities. SuperSQL query can generate various kinds of materials, for example, a LaTeX source file to publish query results in a nested table, HTML or Java source files to present the result on WWW browsers, and other media including MS-Excel worksheet, Tcl/Tk, O2C, etc. O2C is a data manipulation language of O2 and thus useful to migrate data in a relational database to an object oriented database.

SuperSQL is meant to provide a theoretical and practical foundation for 4GL-type applications such as report writers and DB/WWW coordinators.

In this demonstration, we show how TFE reorganize the query results into various media in a universal way, first by grouping tuples according to an arbitrary tree structured schema, and by translating them with the constructors available in the target media.

1 Introduction

A database system is a central repository of information, which are shared among number of applications. It is shared not only by multiple applications but also by multiple types of business applications. For example, the same data is used to produce printed reports, spread sheets, and Web pages which is very important in intra-net environments[2, 3].

A conventional query language for a relational database system yields another relation as a result in a flat table form. So called 4GL systems such as report writers have been used to translate the information obtained from a DBMS to a specific application. Unfortunately, there is no standard language that covers the specification of such translations into various types of application data.

The SuperSQL system is a working prototype which provides the capability of generating various kinds of applicational data directly as a result of a query. The SuperSQL is a simple yet powerful extension of SQL yielding a tree structured data which is then translated into a target medium.

SIGMOD '98 Seattle, WA, USA

We, therefore, call our system a general purpose database publishing and presentation system.

Current implementation is capable of generating printed matters in LaTeX sources[5], Web pages in HTML[4, 6] or Java, spread sheet data in Excel (Visual Basic), and O2C for data migration to an OODB system. The system's architecture is modularized to make the adaptation to an additional medium easy.

There are many softwares that translate database output to some specific medium in the both of research and commercial fields. For example, PENELOPE in ARANEUS project[1] generates Web pages from the contents of a relational database. Lang and Chow[3] introduced numerous database/WWW linkage commercial products including Cold Fusion by Allair. SuperSQL stands out among them, however, because it is universal for many different target madia and application areas. SuperSQL is also unique in its degree of integration to SQL. The integration introduces new possibilities in query optimization techniques[6].

2 An Overview of TFE (Target Form Expression)

TFE is the language feature that makes SuperSQL different from ordinary SQL in publishing/presentation purposes.

TFE (Target Form Expression) is a generalization of target list of SQL. Unlike an ordinary target list, which is a comma-separated list of attributes, TFE uses new operators (connectors and repeaters) to specify the structure of the document generated as the result of the query. Each connector and repeater is associated with a dimension: when generating a Web document in HTML, for example, the first two dimensions are associated with the columns and rows of the `<TABLE>` structure of HTML and the third dimension is associated with hyper-links.

We have introduced the `GENERATE <medium> <TFE>` clause into SQL syntax to clarify the distinction with the `SELECT <target list>` clause. Target medium designations beside `HTML` include `LATEX`, `JAVA`, `EXCEL`, `TCLTK`, `O2C` and `SQL`, in the current implementation.

2.1 Connectors and Repeaters

Binary operators represented by a comma (,), an exclamation point (!) and a percent (%) are used as the connectors of the first three dimensions. Conceptually, they connect the objects generated as their operands horizontally, vertically and in the depth direction, respectively.

A pair of square brackets ([ ]) followed by any of the above connectors is a repeater for that dimension. It will

connect multiple instances in its associated dimension repeatedly. For example,

```
[dept.name, emp.name, emp.salary]!
```

will connect a department name, an employee name and its salary values into horizontal direction and connect them vertically as long as there are such tuples in the query result; it yields the same result as an ordinary target list.

A nesting of a repeater introduces a grouping if the inner repeater is connected to one or more simple attributes. For example,

```
[dept.name!{[emp.name]!,[item.name]!}]!
```

will group employee names and sales item names by a department name, independently. Curly braces ({ }) specify the precedence of connections.

2.2 Sample Presentation with SuperSQL

Figure 1 shows a sample Web document generated by the following SuperSQL query against a movie information database; the database schema is depicted in Figure 2. The type attribute of `film` specifies a category such as `action` or `comedy` and the `pict` attribute provides a still picture. The `face` of the `actor` is also an image attribute.

```
GENERATE HTML verb(Select a Category) !
    [f.type %
       [f.year, [f.title %
         {f.title ! imagefile(f.pict) !
          [imagefile(a.face), a.name, a.birth]!}
       ]!]!
    ],
FROM film f, cast c, actor a
WHERE f.id=c.film and c.actor=a.id
```

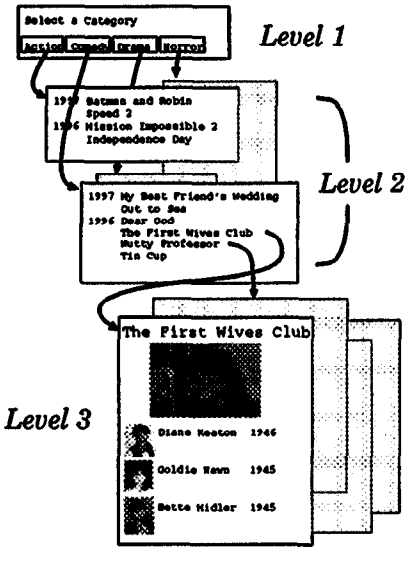

Figure 1: Web Pages Generated with a SuperSQL Query

The query generates a three-level hierarchical Web document whose only root page (first level) contains categories of movies. A third-dimensional connector (%) is used to create

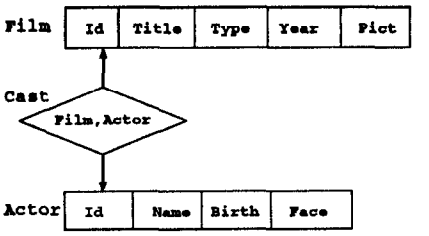

Figure 2: Movie Database Schema

hyper-links from an anchor point (a movie type) to the page associated with it. The second-level pages contain movie titles associated with a category, and each third-level page contains a movie title, a still picture from the movie, and a list of actors and actresses appearing in the movie.

3 The Architecture of SuperSQL Processor

Our current implementation of SuperSQL processor is an independent front-end for various relational database management systems (RDBMS). The 'back-ends' includes free RDBMSs such as PostgreSQL and mSQL.

When a SuperSQL query is submitted to the processor, the parser separates it into a plain SQL query and layout information. The plain SQL query is sent to the back-end RDBMS and a query result is obtained in a flat table form. The tree constructor receives both the layout information and the query result, arranges the result into a tree structure. Finally, a code generator translates it into the designated target medium.

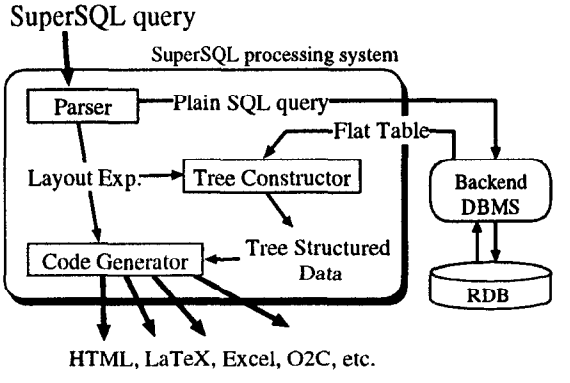

Figure 3: Architecture

A SuperSQL query which is described in the previous section could be easily changed to plain SQL query by changing the TFE to a target list that consists of attributes in the TFE in the original order.

The dependence on the type of back-end DBMS is minimal. The interface, to send a SQL query and to receive the result, must be prepared. We are planning to provide an interface using ODBC to connect to various commercial DBMSs.

The tree constructor receives a query result and rearrange it into tree structure. The code generator is a medium

dependent translator that translates the resulting tree structured data into the language features specific to the target medium.

4 The Trinity Model: a Framework for General Purpose Database Publication and Presentation

The idea of the *trinity model* is depicted in the following figure. Through the experiences on SuperSQL system, we come to recognize the general framework for modelling publication or presentation materials generated from a database.

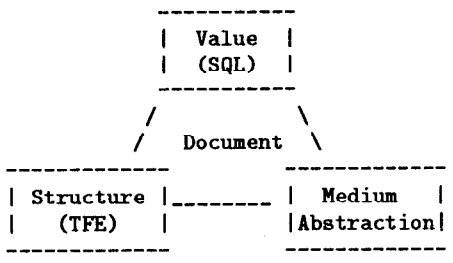

Figure 4: The Trinity Model

The figure shows the idea that a certain documents (application data) could be expressed as a combination of three orthogonal components: the *data value*, the *structure*, and the *medium abstraction*. The values are contained in a relational database and retrieved with a plain SQL query. TFE part of SuperSQL essentially specifies the structure of a resulting document as a tree structure embedded in a three or more dimensional space. The last component, medium abstraction, is a matrix of the data type constructors of a specific medium. For example, the constructor for a vertical repeater is a one column **tabular** form in LaTeX and also a one column **<TABLE>** in HTML.

The meduim abstraction matrix has rows for each dimension and three columns, each of which represents preamble, separator and trailer, respectively. The preamble and the trailer of the dimension zero represent the structures constantly appear at the beginning and at the last part of a document in a specific medium. Followings are extremely simplified examples of medium abstraction matrices for LaTeX and HTML.

LaTeX

Dim	Preamble	Separator	Trailer
0	\documentstyle{article} \begin{document}		\end{document}
1	\begin{tabular}{cc}	&	\end{tabular}
2	\begin{tabular}{c}	\\	\end{tabular}
3		\vfill\eject	

HTML

Dim	Preamble	Separator	Trailer
0	<html><body>		</body></html>
1	<table><tr><td>	</td><td>	</td></tr></table>
2	<table><tr><td>	</td></tr><tr><td>	</td></tr></table>
3	<a href="	">	

Figure 5: Sample Medium Abstraction Matrices

The three components of the trinity model are orthogonal because 1) we can change the contents of a document without changing the structure and medium by changing the values in the database, 2) we can change the layout of a document by replacing the TFE with another leaving the values and the medium designation intact, and 3) we can specify different media for a document presented in the same value and the same layout structures just by changing the medium designation.

From this point of view, we foresee the idea of normalization of documents, which is comparable to normal form design of relational database, to make them designed for better manageability.

5 Conclusion

In this demonstration, we present SuperSQL prototype system to show the possibility of providing a universal language for database publishing and presentation. The TFE introduced in SuperSQL is capable of generating sophisticated hierarchical layout structures that are required in broad range of real world applications.

Although not mentioned in depth in this paper, the demonstration should cover not only the ordinary publishing / presentation such as in LaTeX or in HTML, but also the data migration to OODB or the data downloading to MS-Excel, all treated in a single principle of structuring with TFE.

We hope this work helps extending the role of DBMS as a central repository of information supplying various applicational needs, while avoiding the 4GL chaos.

Acknowledgment

The author is grateful to Calton Pu, David Maier, Lois Delcambre and Lyman Do for the valuable discussions while the author was visiting OGI. The author thanks deeply to his (ex-)students who worked very hard on implementing SuperSQL in various stages. Some of them, including Toshiyuki Seto, Kei Sato, Hideo Sashida, Yuji Nakagawa, Takayasu Kato, Takuhiro Nagafuji, Noriyuki Tanaka, Tatsuya Arisawa, Shiro Udoguchi, Akira Ohara and Yoshiyuki Kawamura are actually worth the co-authorship of this paper.

References

[1] P. Atzeni, G. Mecca, P. Merialdo, Semistructured and Structured Data in the Web: Going Back and Forth, in *SIGMOD Record*, 26(4), ACM (1997), 16-23.

[2] P-C. Kim, **http://grigg.chungnam.ac.kr/projects/UniWeb/documents/www_dbms.html**

[3] C. Lang, J. Chow, *Database Publishing on the Web and Intranets*, Coriolis Group Books (1996).

[4] T. Seto, T. Nagafuji, M. Toyama, Generating HTML Sources with TFE Enhanced SQL, in *Proc. ACM Symp. on Applied Computing (SAC '97)*, ACM (1997), 96-105.

[5] M. Toyama, Three Dimensional Generalization of Target List for Simple Database Publishing and Browsing, in *Research and Practical Issues in Database (Proc. 3rd Australian Database Conference)*, World Scientific Pub. Co. (1992), 139-153.

[6] M. Toyama, Dynamic and Structured Presentation of Database Contents on the Web, to appear in *Proc. EDBT '98.*